



HAL
open science

IoT networks 3D deployment using hybrid many-objective optimization algorithms

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

► **To cite this version:**

Sami Mnasri, Nejah Nasri, Malek Alrashidi, Adrien van den Bossche, Thierry Val. IoT networks 3D deployment using hybrid many-objective optimization algorithms. *Journal of Heuristics*, 2020, 26, pp.663-709. 10.1007/s10732-020-09445-x . hal-02613283

HAL Id: hal-02613283

<https://hal.science/hal-02613283>

Submitted on 15 Sep 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 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/26398>

Official URL: [DOI:10.1007/s10732-020-09445-x](https://doi.org/10.1007/s10732-020-09445-x)

To cite this version: Mnasri, Sami and Nasri, Nejah and Alrashidi, Malek and Van den Bossche, Adrien and Val, Thierry *IoT networks 3D deployment using hybrid many-objective optimization algorithms*. (2020) *Journal of Heuristics*, 26. 663-709. ISSN 1381-1231

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

IoT networks 3D deployment using hybrid many-objective optimization algorithms

Sami Mnasri¹  · Nejah Nasri^{2,3} · Malek Alrashidi² · Adrien van den Bossche¹ · Thierry Val¹

Abstract

When resolving many-objective problems, multi-objective optimization algorithms encounter several difficulties degrading their performances. These difficulties may concern the exponential execution time, the effectiveness of the mutation and recombination operators or finding the tradeoff between diversity and convergence. In this paper, the issue of 3D redeploying in indoor the connected objects (or nodes) in the Internet of Things collection networks (formerly known as wireless sensor nodes) is investigated. The aim is to determine the ideal locations of the objects to be added to enhance an initial deployment while satisfying antagonist objectives and constraints. In this regard, a first proposed contribution aim to introduce an hybrid model that includes many-objective optimization algorithms relying on decomposition (MOEA/D, MOEA/DD) and reference points (Two_Arch2, NSGA-III) while using two strategies for introducing the preferences (PI-EMO-PC) and the dimensionality reduction (MVU-PCA). This hybridization aims to combine the algorithms advantages for resolving the many-objective issues. The second contribution concerns prototyping and deploying real connected objects which allows assessing the performance of the proposed hybrid scheme on a real world environment. The obtained experimental and numerical results show the efficiency of the suggested hybridization scheme against the original algorithms.

Keywords IoT collection networks · 3D indoor redeployment · Experimental validation · Many-objective optimization · Preference incorporation · Dimensionality reduction

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s10732-020-09445-x>) contains supplementary material, which is available to authorized users.

Extended author information available on the last page of the article

1 Introduction

To implement a wireless sensor network (WSN), the location of the nodes should be first chosen according to specific criteria in order to optimize several targeted objectives like coverage, localization, connectivity or consumption rate of energy. Thus, node deployment greatly influences the network performance and its operation. It aims essentially at proposing a network topology with well-defined number and positions of nodes. This deployment is said to be 3D if the variation of the heights between nodes are important with respect to the width and length of the “Region of Interest” (RoI).

In this study, we investigate the 3D deployment which is more complicated and represents the RoI topography better than the 2D deployment. The migration of the WSNs to the Internet of Things (IoT) gave birth to the Internet of Things collection networks which consist of a set of connected objects that collect information from the RoI. Therefore, our main issue consists in deploying an indoor 3D DL-IoT, a scenario where autonomous objects (devices, robots, etc.) having unique identifiers can communicate with each other via a set of protocols such as Bluetooth or 802.15.4 to transmit the measures sensed by the sensors of the WSN. In fact, IoT is responsible for processing the collected values and making the decisions. We are specifically interested in the redeployment problem where a number of nodes are added to an initial configuration of nodes. Indeed, in our experiments, we focus on the indoor 3D deployment of nodes in a site composed of several buildings.

Most mathematical formulations consider the 3D deployment as a NP-hard problem (Cheng et al. 2008) which cannot be solved by deterministic approaches especially for the large size of the problem. This requires using heuristic approaches. In this paper, we suggest a many-objective modeling of the deployment problem based on real hypotheses and constraints. Hence, many-objective evolutionary algorithms (MaOAs), such as Two_Arch2 (Wang et al. 2015) and MOEA/DD (Li et al. 2015) are used with a hybridization scheme incorporating dimension reduction and user preferences. Besides, several issues encounter the MaOAs when resolving the real-world problems with a high number of objectives exceeding three as it is the case in our approach.

Recently, with the continuous rise of the number of objectives, the complexity and the realism of optimization problems, the interest of the evolutionary multi-objective optimization (EMO) community is focused on the evolutionary many-objective optimization (EMaO). This focus is explained by the fact that the efficiency of most evolutionary optimization algorithms deteriorates if the number of objectives exceeds three (Ishibuchi et al. 2012).

The most important challenges faced by EMO and EMaO when tackling many-objective optimization problems (MaOPs) are as follows: The exponential complexity in space and time, the inaccuracy of Pareto-based EMOs, the problem of representing the trade-off surface, the ineffective recombination and mutation operators and the inaccuracy of density estimation. Hence the need of using new approaches resolving these issues. Moreover, the many-objective optimization theory is based on the fact that optimizing each objective independently from the others cannot give a good candidate solution and even if a good representation of the Pareto Front is done for high-dimensional objective space, maintaining the diversity is not ensured (Yuan et al.

2016). This is due to the complexity of the real-world problems and the nature of the objectives: unless the objectives are all inter-dependent and they can be reduced to one or two objectives, these latter are often antagonist, which leads to the reality that improving an objective value separately will deteriorate one or more other objective values (Rostami et al. 2014).

As a solution to overcome the mentioned challenges, our approach relies on combining different paradigms (reduction, preference and evolutionary ones) in a well justified scheme.

The main proposed contributions of this study are presented below:

- To evaluate the behavior and the performance of the EMO algorithms, the majority of the EMO studies and their applications rely only on theoretical hypothesis or simulations in the case of engineering problems. Unlike these studies, ours is based on real empirical experiments with real nodes and a platform of prototyping. Indeed, the importance of this work relies in the used algorithms based on real hypotheses and practical measurements. It presents a proof of the accuracy of the recent algorithms, such as MOEA/DD and Two_Arch2, which are studied by their authors only on academic problems like ZDT and DTLZ.
- Moreover, the proposed mathematical formulation gives a detailed description of the 3D deployment problem taking into account various real hypotheses and constraints to comply with the assumptions of the experiments and the numerical tests.
- Another analysis contribution consists in the proposed hybridization scheme which includes different classes of many-objective algorithms based on reference points, reduction of dimensionality and incorporation of preferences. According to the obtained results, this hybridization scheme increases the performance of the original used MaOAs.

The remainder of the paper is composed of the following sections: a set of relevant recent related works on the 3D deployment in WSN are discussed and criticized in Sect. 2. Afterwards, an integer linear programming modeling is detailed in Sect. 3. The suggested hybrid scheme is presented in Sect. 4. Then, the numerical results of the EMOs evaluation and interpretations are discussed in Sect. 5. Statistical nonparametric tests are proposed in Sect. 6. Moreover, experimental tests on a platform of prototyping are investigated in Sect. 7. Finally, concluding findings are presented in Sect. 8.

2 Related works

Different studies have suggested optimization evolutionary approaches, such as Genetic Algorithms (GAs), to guarantee an efficient deployment in WSNs. Table 1 illustrates the recent researches in this context.

In relation to our approach, in order to resolve the 3D indoor deployment issues in WSN, authors in Mnasri et al. (2017a) and Mnasri et al. (2015) proposed a genetic algorithm then a hybrid algorithm that stems from the behavior of ant foraging. Despite the efficiency of the latter algorithm compared with the standard ACO and NSGA-III, the scalability of this algorithm is not tested in dense networks. In the same regard, authors in Mnasri et al. (2018) aim to resolve the deployment problem using the

Table 1 Recent studies resolving the deployment problem using optimization approaches

| References | Approach | Application | Objectives | Highlights | Drawbacks |
|-------------------------|--|---------------------------|----------------------------------|---|---|
| Xu et al. (2014) | A parsing crossover strategy for GAs | Sensor deployment problem | Reduce the redundancy of sensors | Under different terrain irregularities, the proposed parsing crossover strategy showed better performances than uniform, one-point and two-point crossover strategies | No instances of failure of GAs are recorded. Besides, authors did not indicate that the proposed parsing crossover may be the best solution in such cases |
| Ko and Gagnon (2015) | A tabu search (TS) and a GA based on an integer linear programming model | Sensor deployment problem | Coverage connectivity | This method outperformed some regular sensor deployment patterns | No optimal solution for the big size deployment problem For simulations, a simple case of study was illustrated and no simulator or known benchmarks were used to show the efficiency of the used approach |
| Drechsler et al. (2015) | A new algorithm called Prio- ϵ -Preferred based on a ϵ -preferred relation | [Nurse Rostering Problem] | – | The incorporation of user preferences in many-objective optimization algorithms | The suggested algorithm is compared only with NSGA-II and there is no comparison with other user preferences approaches |

Table 1 continued

| References | Approach | Application | Objectives | Highlights | Drawbacks |
|-----------------------------|--|---|--|---|---|
| Domingo-Perez et al. (2016) | Genetic Evolutionary multi-objective optimization | Indoor sensor placement | Number of sensors, coverage and accuracy | A realistic constraint (the existence of obstacles) is considered | A comparison with other optimization algorithms is needed to position the proposed strategy |
| Zhang et al. (2018) | A balanced evolution genetic algorithm (BEGA) based on a similarity guide matrix (SGM) | - | - | Inject randomness in the population to better control the diversity BEGA is compared with different other genetic variants for 12 benchmarks | No tests on real industrial problems are proposed and no efficiency proof of its application in large-scale problems is given |
| Luo et al. (2018) | Potential Game with Rigid Graph Theory | Rigid Topology management in Wireless Sensor Networks | Node degree Energy consumption | The problem is well formulated | The choice of the parameters for the performance evaluation is not justified Different other indexes could be used in order to appraise the algebra rigid properties |
| Huang et al. (2019b) | Data plane deployment solutions and optimization | Hybrid SDN Networks | Traffic engineering Resource saving Network security | A review of control plane solutions and data plane deployment, and a description of typical cases of SDN hybrid networks | More practical simulation and emulation tools can be used to assess the behavior of the tested strategies using typical performance metrics. |

Table 1 continued

| References | Approach | Application | Objectives | Highlights | Drawbacks |
|---------------------------|---|--|---|--|--|
| Argany et al. (2018) | Evolutionary global optimization algorithms | Two and three dimensional raster models with different resolutions | Coverage | A stochastic sensing models in complex environments Raster and vector environment models | Global optimization algorithms are not compared with other approaches |
| Guo and Jafarkhani (2019) | The deployment and relocation problem is modeled as a constrained source coding problem | Heterogeneous mobile wireless sensor networks | Coverage energy consumption, connectivity | A realistic constraint (the mobility) is considered The performance evaluation takes into consideration different scenarios | A high complexity approach |
| Savkin and Huang (2019) | A distributed optimization model | Surveillance Aerial Drones | Coverage, crowd monitoring, connectivity | A detailed modeling of the problem is given | Different other aspects such as the energy consumption and the communication link quality are not studied |
| Elhabyan et al. (2019) | [A survey] | Coverage in wireless sensor networks | Coverage Energy consumption | A detailed taxonomy for classifying coverage protocols in wireless sensor networks | No simulations or real experimentations are carried out to compare the protocols A non-realistic sensing model: does not reflect the WSN anisotropic properties Connectivity is a crucial objective to be considered |

Table 1 continued

| References | Approach | Application | Objectives | Highlights | Drawbacks |
|-------------------------|--|--|---|---|--|
| Liu et al. (2019) | A path planning strategy (Dual Approximation of Anchor Points) | Disconnected Sensor Networks with Mobile Sinks | Full connectivity Coverage | Low complexity of the proposed strategy | The efficiency of the proposed strategy is not tested in dense networks (large scale problem) |
| Tsang et al. (2019) | A mapping multi-Objective method: a k-means Taguchi-guided clustering embedded in GA | 3D environmental sensor deployment | Connectivity, coverage, heating airflow ventilation and air conditioning, lifetime, fault tolerance | A deployment scheme with three stages is established and a fuzzy window of temperature is suggested to adjust the sensor activation times over different ambient temperatures | The proposed methods must be compared to other deployment approaches using different scenarios |
| Huang et al. (2019a, b) | Three data-centers deployment optimization (DCDO) schemes | Data centers deployment in Vehicular Networks | Code dissemination Data collection | Extensive simulations and experiments relying on two real datasets of the cabs' GPS coordinates | The problem formulation does not rely on realistic assumptions |

incorporation of explicit user preferences procedure (PI-EMO-VF) applied to a many-objective novel variant of the genetic algorithms (NSGA-III).

Compared to the previous studies, we suggest in this paper a new scheme which interactively integrates another more efficient explicit user preferences procedure (PI-EMO-PC) with an implicit one (finding knee regions, ideal and nadir points) applied to four optimization algorithms from different classes (MOEA/DD, MOEA/D, NSGA-III and Two-Arch2). Another difference is that we propose a new scheme of hybridizing the indicated preferences procedures with dimensionality reduction (NLMVU-PCA and L-PCA).

3 Integer linear programming formulation

To resolve our problem, the following model is suggested. We consider the following types of nodes:

- Stationary nodes which are the initially installed fixed nodes. This type of nodes can be randomly disseminated. But, it is better to adopt a strategy to distribute them according to the applicative objectives.
- Nomad nodes which are added to enhance the 3D deployment scheme. Their positions are identified by the proposed algorithms.
- Mobile nodes which are a set of targets to control. Equipped with a sensor transmitting and receiving signals.

The following sets, variables of decision and parameters are used:

- Sets
- **S**: is the set of potential sites where sensor nodes can be installed. $\mathbf{S} = \mathbf{S}_a \cup \mathbf{S}_b$ such that the set “ \mathbf{S}_a ” represents the potential sites where stationary sensors can be installed. “ \mathbf{S}_b ” is the set of potential sites where the nomad sensors can be installed. Note that a site must not be in two different sets. Thus, $\mathbf{S}_a \cap \mathbf{S}_b = \emptyset$.
- **N**: is a set that denotes the different types of the nodes. Let $\mathbf{N} = \mathbf{N}_a \cup \mathbf{N}_b$ such that the set “ \mathbf{N}_a ” represents the types of the stationary nodes. The set “ \mathbf{N}_b ” represents the types of nomad nodes. We can use various types of sensors having different functionalities which can be gathered in the same sensor like detecting the degree of temperature, the degree of luminosity or the opening and closing of doors.
- **T**: is the set of mobile targets to be detected; “ t_k ” is a target.
- **V**: is the set of nodes having several types in **N** and deployed in several sites in **S**.
- **K** is the set of scheduling periods when a sensor $i \in \mathbf{V}$ is activated.
- Decision variables
- $S_{ss'}$ equal to 1 if the sensor positioned at a site $s \in \mathbf{S}$ detects a signal from a sensor positioned at a site $s' \in \mathbf{S}$ with a power of transmission sufficient to detect it; 0 otherwise.
- $T_{ss'}$ equal to 1 if the sensor positioned at a site $s \in \mathbf{S}$ transmits a signal from a sensor positioned at a site $s' \in \mathbf{S}$ with a power of transmission sufficient to detect it; 0 otherwise.

-
- X_{ts} equal to 1 if a sensor positioned at a site $s \in S$ receive a signal from a target at a location $t \in T$ with a power of transmission greater than or equal to the minimum power required to detect it; 0 otherwise.
 - Pos_{ijk} a real variable representing the 3D coordinates providing the potential indoor position of a sensor.
 - $CovP_{ijk}$ equal to 1 if and only if the position Pos_{ijk} is covered by a node with a power of transmission greater than or equal to the minimum required power to detect it; 0 otherwise.
 - $Pf x_s^n$ set to 1 if a stationary node having a type $n \in N$ is positioned at a site $s \in S$; 0 otherwise.
 - Pnd_s^n set to 1 if a nomad node having a type $n \in N$ is positioned at a site $s \in S$; 0 otherwise.
 - Pmb_s^n set to 1 if a mobile node having a type $n \in N$ is positioned at a site $s \in S$; 0 otherwise.

The decision variables are related to each other's in different forms: Indeed, The two variables $Sg_{ss'}$ are $T_{ss'}$ complementary since they represent respectively the ability of reception and transmission of signals between a node and other nodes in the RoI. However, X_{ts} represents the ability of a node to detect the targets in the RoI. Pos_{ijk} representing the 3D position of the node is in a direct relation with $CovP_{ijk}$ indicating if the position Pos_{ijk} is covered or not by a node. The three variables $Pf x_s^n$, Pnd_s^n and Pmb_s^n represents the three types of nodes and its relations with the set of possible sites in the RoI.

As regard the relation between the attenuation ratio between two nodes, the received signal and the distance between these two nodes, this can be modeled by the following two relations:

$$d_{ss'} = \alpha \times Sg_{ss'} \times \delta_{ss'}, \quad \alpha \in R, \quad s \in N, \quad s' \in N \quad (1)$$

Constraint (1) links the distance to the power transmission of the signal between two nodes s and s' . α is a real empirically-determined coefficient.

$$(Sg_{ts} = 1) \Rightarrow (d_{ts} \leq d_{max}) \forall t \in N, s \in N \quad (2)$$

Constraint (2) implies that if there is a signal Sg_{ts} between two nodes t and s , the distance (d_{ts}) between t and s must not exceed the pre-defined maximum distance (d_{max}).

Other relations between the decision variables are modeled in the proposed objective functions and in the constraints.

- Parameters
- M denotes the number of used objectives
- nbT denotes the number of mobile targets. nbF is a parameter representing the number of initially stationary nodes. This number can be set by default to a random number or to $n_{\min} \cdot (nm/2\pi r^2)$. r is the radius of a sensor. nbN is the number of nomad nodes to add. Let N_{\max} be the maximum number of nodes that can be deployed within the wireless network. Thus $(nbT + nbF + nbN) \leq N_{\max}$.

- C_s^n is the hardware cost of a node (including price) having a type $n \in N$ and installed at a site $s \in S$.
- Bt_i is the remaining energy in the battery of the sensor i at an instant t .
- n_{\min} stands for the degree of coverage. It defines the minimum number of sensors to localize a target emitting a signal. When using the proposed hybrid 3D localization model (based on 3D DV-Hop and RSSI), the parameter n_{\min} is generally set to 4.
- $Lf > 0$ is the lifetime of the network (i.e. time in which the desired coverage degree is guaranteed) and Lf_{\max} is an upper bound for Lf . Lf_i is the lifetime of the node $i \in V$.
- TP_i^f representing the power of the signal transmitted (emitted RSSI) of the sending node $i \in V$.
- RP_i^f representing the power of the signal received (emitted RSSI) at a distance r from the sender $i \in V$.
- $\delta_{ss'}$ represents the attenuation ratio between two nodes in two sites $s \in S$ and $s' \in S$.
- d_{ts} is a parameter representing the distance between two nodes 't' and 's'.
- d_{\max} is a constant representing the maximum distance separating a node i and a target j or separating two nodes i and j so that they could detect each other.

3.1 The objective functions

The many-objective fitness function is: Maximize $\mathbf{F}(\vec{x})$ where $\mathbf{F}(\vec{x}) = (f_1, \dots, f_8)$

3.2 The number of the added nomad nodes

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

$$f_1 = \text{Minimize} \sum_{s \in S} Pnd_s^n \quad (3)$$

$$\text{Subject to} \sum_{s \in S} Pfx_s^n \leq nbF \forall s \in S, n \in N \quad (4)$$

$$\sum_{s \in S} Pnd_s^n \leq nbN \forall s \in S, n \in N \quad (5)$$

$$\sum_{s \in S} Pmb_s^n \leq nbT \forall s \in S, n \in N \quad (6)$$

3.3 Energy consumption

A deployed active sensor dissipates energy when transmitting, sensing, receiving, or being idle. Therefore, energy efficiency is considered as a fundamental key in designing a wireless sensor network. Since being idle and sensing energies are negligible compared with transmitting and receiving energies, we proposed a model in which E_i^{elec} represents the energy consumed to reactivate the transmitter/receiver circuit and ϵ_{amp} represents the transmitter amplifier to communicate. The energy dissipated to

transmit m -bit packet within a distance d is E_i^{transm} and the energy of receiving the same packet is E_i^{recv} .

$$f2 = \text{Minimize } \sum E_i^{transm} + \sum E_i^{recv} \text{ where } E_i^{recv} \\ = E_i^{elec} * m \text{ and } E_i^{transm} = E_i^{elec} * m + \epsilon_{amp} * m * d^2 \quad (7)$$

Subject to the following constraint: In order to minimize the consumption of energy, we can minimize the interferences during transmission. The neighboring nodes cause interferences which can be minimized by out bounding the maximum number of neighbors that a sensor may have as indicated in (8).

$$\sum_{s=0, n=0} Pfx_s^n + \sum_{s=0, n=0} Pnd_s^n + \sum_{s=0, n=0} Pmb_s^n \\ \leq nbT + nbF + nbN + |S| - |S| * (Pfx_s^n + Pnd_s^n + Pmb_s^n) \quad (8)$$

3.4 Hardware deployment cost

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

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

3.5 Network Utilization

To optimize the network lifetime, nodes can be placed near to the base station which can cause a poor utilization of the resources in the network and increase the overall cost of deployment. Thus, it is important to extend the network lifetime with the simultaneous deployment of a reasonable number of sensors. The network utilization (NU) is modeled as:

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

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

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

$$\text{Subject to } \overline{lf} \cdot \sum (Pfx_s^n + Pnd_s^n + Pmb_s^n) \leq \overline{lf} \max \text{ where } \overline{lf} \max = 1/lf \max \quad (12)$$

3.6 Localization rate

We suggest a hybrid localization model that enhances the utilized range-free technique (3DDV-Hop) by incorporating a range based localization Received Signal Strength Indication (RSSI). To guarantee better localization, each monitored target $t \in T$ must be surveyed by at least n_{\min} anchor nodes. Then, $\sum_{s \in S} x_{ts} \geq n_{\min} \forall t \in T$. Thus, the following function (11) is suggested to model the localization:

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

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

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

3.7 Coverage rate

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

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

3.8 Lifetime

In the literature, the network lifetime can be modeled as the time in which the first node totally consumes its energy or as the time until the first loss of coverage appears. In fact, different factors, like the node density, the node transmission, the routing strategies

and the initial energy, can influence the network lifetime. To model the lifetime, we suggest the following function:

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

Subject to the following constraints:

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

$$\sum_{s \in S} Pfx_s^n + \sum_{s \in S} Pnd_s^n \leq N_{\max} + |S| - |S| \times (Pfx_s^n + Pnd_s^n) \forall s \in S, n \in N \quad (18)$$

where $N_{\max} = Pfx_s^n + Pnd_s^n + Pmb_s^n, \forall s \in S, n \in N$ and $Lf_i = Bt_i / \max(E_i^{transm} + E_i^{recv}), \forall i \in V$. The network lifetime is equal to the minimum lifetime Lf_i among the lifetimes of all sensors.

3.9 Connectivity rate

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

$$f8 = \text{Maximize } RP_i^r \quad (19)$$

Subject to

$$RP_i^r \leq T_{ss'} * Sg_{ss'} * \alpha * r^{-\omega} * TP_i^r \quad (20)$$

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

$$r = r_c \Leftrightarrow RP_i^r = P_{\min}^n \quad (21)$$

Constraint (21) indicates that the sender can be connected to the receiver and the data can be received only when the power at the receiver is greater or equal to P_{\min}^n . The transmission range r_c is defined by $RP_i^r(r = r_c) = P_{\min}^n$.

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

$$\sqrt{-\ln(1 - (Pr ob^G)^{1/n}) / \lambda \pi} \leq r_c \quad (22)$$

$$\sum_{s \in S} x_{ts} \leq \sum_{n \in Na} Pfx_s^n + \sum_{n \in Nb} Pnd_s^n \quad (23)$$

Constraint (23) denotes the number of nodes able to detect a target. This number should not exceed the number of the installed nodes in the different sites.

4 The proposed hybrid scheme

Indeed, resolving real-world problems become particularly critical when the objectives are conflicting or should be handled simultaneously. For such complex problems, the aim is to select a solution from a set of possible solutions. Often, this selection is difficult and manually done by the decision maker. Hence the need of an efficient approach reducing the complexity and number of objectives, resolving the problem and selecting an acceptable solution. Therefore, the choice of the proposed algorithm is motivated by the issues raised by the many-objective optimization. The main contribution is not only a combination of existing algorithm components: The aim is to find solutions to the limitations of the existing proposed algorithms. To resolve the previously-mentioned difficulties of EMOs in resolving MaOPs, we suggest a justified hybrid scheme incorporating different approaches. This scheme is illustrated in Fig. 1.

In this scheme, four classes of MaOEA are combined: decomposition-based, reference point-based, reduction-based and preference-based.

Firstly, the MaOA is executed (MOEA/DD, NSGA-III or Two-Arch2) with preserving the diversity using an adaptive neighborhood mechanism. Thus, the solutions corresponding to the optimization of the initial set of objectives are obtained. On this set, the dimensionality reduction is performed and the solutions corresponding to a

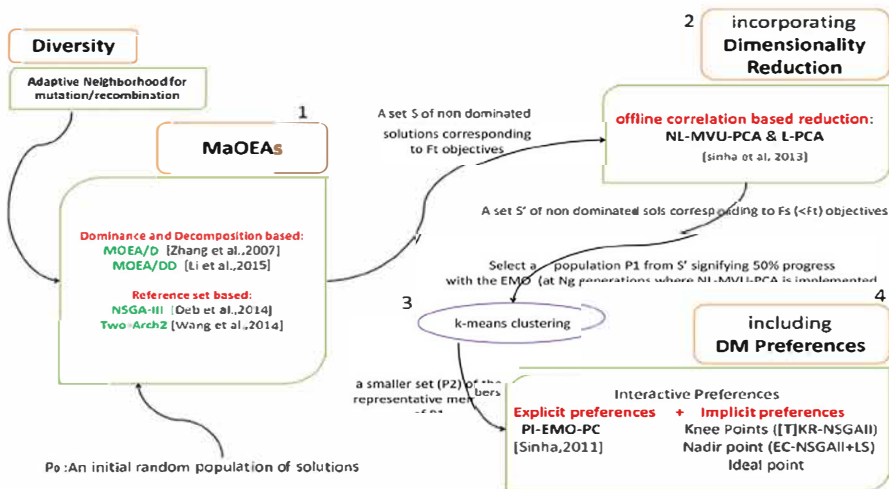


Fig. 1 The four steps of the proposed hybridization scheme

smaller set of objectives is provided. Afterwards, the preference procedure is achieved. Indeed, the preference methods assume that there are no redundant objectives in the given problem (Saxena et al. 2013). Thus, in our approach, dimensionality reduction procedure is always performed before applying the decision maker (DM) preferences.

Since the PI-EMO-PC procedure requires a sufficient search window, its input population must be chosen while guaranteeing that the search converges to a solution in accordance with the interactive DM's preferences. To choose the input population for our preference procedure guaranteeing a balance between the computational performance and the convergence to a candidate solution, we follow the same steps as indicated in Sinha et al. (2013): The input population is an intermediate population taken at Ng generations (50% of progress) after applying the reduction. After identifying this input population for the preference procedure, only a smaller set representing its members is considered using the k-means clustering. The previously indicated members composed the initial population members of the PI-EMO-PC.

4.1 Including diversity

In the MOEA/D relying on decomposition, the multi-objective optimization problem is decomposed into a set of single objective sub-problems (or simple multi-objective ones) then a population based method is used to optimize these sub-problems simultaneously and independently but cooperatively. In MaEAs like MOEA/D, the high dimension of the objective space leads to very diversified population, which make the recombination and mutation operators inefficient and produce dominant offspring. As a solution to these issues, we proposed a strategy that relies on adaptive multi-operators with a niching restricted on the neighborhood. In what follows, we present and explain these concepts and the proposed algorithms.

4.1.1 Chromosomes coding and fitness functions definition

***Chromosomes coding** The first phase of an EMO is the coding of the chromosomes. In the 3D deployment problem, each chromosome is an individual that correspond to a feasible position of a node in the 3D space of the RoI. In our work, the chromosome, whose genes express the value of his position according to the x, y and z axes, is coded as a binary point of the position (x, y, z). Figure 2 illustrates the chromosome representing the node mapped to the location [46, 53, 34]. The choice of the initial number of chromosomes depends on several factors including the initial existing distribution of nodes and the 3D shape of the RoI. In this study, the binary coding is applied thanks to its simple use, low computational complexity and its adaptability to the search in a neighborhood where it is enough to change a gene (in the case of mutation) or some consecutive genes (in the case of recombination) in order to have another individual in the neighborhood. Although the obtained individual is not a possible position in the RoI, it will be discarded by penalizing it with a weighting coefficient.

In many-objective optimization, many studies proved that improving the crossing can be done by using the neighbors. Thus, the distance between the chromosomes to cross $\phi(i, j) \forall i \in V, j \in V$ should be minimized. ϕ is the distance between the

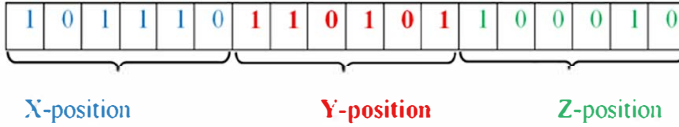


Fig. 2 The chromosome representing the sensor in the 3D position (46, 53, 34)

Table 2 Positions of nomad added nodes proposed by a candidate solution

| | N1 | N2 | N3 | N4 | N5 | N6 |
|-------|-----|-----|-----|-----|-----|-----|
| x_i | 149 | 348 | 59 | 269 | 368 | 541 |
| y_i | 35 | 532 | 682 | 186 | 87 | 409 |
| z_i | 236 | 191 | 373 | 365 | 434 | 122 |

two nodes on the search space and i, j are two candidate nodes representing the two chromosomes to cross.

The decision of using binary representation of chromosomes is privileged due to its ease of use and low complexity which is very recommended when manipulating MaOPs. Moreover, this decision is explained by favoring the neighbors in mutation and recombination in our methodology: In fact, the binary coding permits to better survey the distinctions in qualities between two chromosomes according to the distance between them. However, binary coding may give non-possible solutions. Weighting coefficients will penalize these solutions which will not be chosen later by the algorithm.

***Fitness functions definition** Let's consider a RoI of $700 * 700 * 700$ with 15 initial fixed nodes, the following candidate solution represents an individual from the population chosen as a suggested optimal solution (belongs to the final population that represent the Pareto Front, after the stopping condition is met and the algorithm converges). According to our model, this candidate solution of the deployment problem represents the locations (coordinates in 3D plan) of a set of positions that meets the final proposed positions of the nomad nodes added to correct the initial deployment proposing the set of fixed nodes, in Table 2. Note that this representation of nodes in Table 2 is the transformation from the binary coding to the decimal one. Binary coding is used to apply the mutation and recombination operators.

The corresponding fitness values (average of 30 random-chosen iterations) of objectives are as follow (Table 3).

Let's explain for example how the fitness function f_6 is defined: it symbolizes the coverage degree of nodes in the RoI: the coverage degree can be measured by computing the degree of coverage of targets in the RoI. To achieve the 3D full coverage, every position in the RoI must be covered by at least one sensor (ideally, by n_{\min} sensors). Thus, the sum of all positions occupied by nodes should be more than n_{\min} : the sum of binary variables $Cov P_{ijk}$ should be equal or higher than n_{\min} ($\sum_{s \in S} Cov P_{ijk} \geq n_{\min}$). Hence the objective function is to maximize the sum of the gathered positions occupied and covered by sensors.

Table 3 Average values of fitness functions

| | Formula | Average values | |
|------|--|----------------|---------|
| | | NSGA-III | MOEA/DD |
| $f1$ | Minimize $\sum_{s \in Sb} Pnd_s^n$ (3) | 126.98 | 134.367 |
| $f2$ | Minimize $\sum E_i^{transm} + \sum E_i^{recv}$ (7) | 4.266 | 3.843 |
| $f3$ | Minimize $\sum_{s \in Sb} \sum_{n \in Nb} Pnd_s^n C_s^n$ (9) | 89.701 | 92.124 |
| $f4$ | Minimize $\overline{lf} \cdot \sum (Pfx_s^n + Pnd_s^n + Pmb_s^n), \forall s \in S, n \in N$ (11) | 0.787 | 0.928 |
| $f5$ | Maximize $\sum_{t \in T} \left(\sum_{s \in S} x_{ts} - n_{\min} \right)^+$ (13) | 3.761 | 3.235 |
| $f6$ | Maximize $\sum_{t \in T} \left(\sum_{s \in S} CovP_{ijk} - n_{\min} \right)^+$ (15) | 4.833 | 5.136 |
| $f7$ | Maximize Lf (16) | 3532 | 3982 |
| $f8$ | Maximize RP_i^r (19) | 164.022 | 171.679 |

Since we use a hybrid 3D localization model relying on combining the 3D DV-Hop and RSSI information, the coverage degree for each position is generally set to 4 (4 anchors are needed for localization). Note here that MOEA/DD is better than NSGA-III since its average fitness value representing the coverage ($f6$) is equal to 5.13 which is higher than the average fitness value proposed by NSGA-III (4.83). Note also that the fitness function value may do not reflect the exact value of the concept for which it is defined. This depends of its formula that can consider coefficients, constants and decision variables: for example, the number of added nomad nodes in our example is equal to 6 while the fitness function value is about 130 (126 for NSGA-III and 134 for MOEA/DD).

4.1.2 The neighborhood restriction strategy

In the case of MaOP, the dimension of the objective space is too high. This increases the population diversity. Thus, the mutation and recombination operators become ineffective; hence the possibility that they create individuals who are not selected as parents. Indeed, the used neighboring concept is based on the following steps:

- Computation of the distance in the objective space separating the individuals.
- Identification of the subset of $(|P|.N_s)$ nearest neighbors for each individual, where N_s is the size of the neighborhood and P is the population. According to Qu et al. (2012), N_s between 1/20 and 1/5 is preferred. In our tests, $N_s = 1/10$.

As case of the efficiency of utilizing the neighborhood in including diversity in MaOEA, authors in Ishibuchi et al. (2012) demonstrates that MaOEA are more efficient if the recombination takes place with a neighboring chromosome and if the objectives are correlated.

***The neighbourhood mutation** The neighbourhood mutation restricts producing new individuals on an area which is near from their parents in order to establish a stable niching. Authors in Qu et al. (2012) prove that the neighbourhood mutation improves the detection of local optima. They propose a study to identify the preferred neighbourhood size and its effect on the algorithm behaviour. They affirm that the ideal neighbourhood size should be between 1/5 and 1/20 from the population. Hence, the size of the neighbourhood is a specific parameter to be set proportionally to the size of the population. In our approach, achieving a neighbourhood mutation needs to set only one parameter: the neighbourhood size ns which determine the number of mutation vectors in each subpopulation. To model this, we propose to minimize $\phi(i, j) \forall i \in V, j \in V$ two candidate individuals to mate and ϕ is the search space distance between them. Moreover, the proposed strategy allows each individual to evolve progressively toward its “nearest optimal point”. In addition, the algorithm is independent from the neighbourhood size. The proposed neighbourhood mutation algorithm is shown in Algorithm 1.

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

Algorithm 1 The Neighbourhood mutation algorithm

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

***The neighborhood recombination** In evolutionary optimization algorithms, the recombination permits generating good offspring from parents. In ideal cases, this new offspring should contain a set of uniformly scattered non-dominated solutions. Firstly, according to the selection mating model, any two individuals belonging to the population may be considered as “parents”. However, this type of mating model suffer from the uncertain choice of parents and the possibility of having a large Euclidian-distance between them, which increase the possibility of obtaining dominated offspring. To mitigate this situation, a more determinist selection model relying on mating closer individuals having shorter Euclidian-distance on the objective space, can be considered to perform the recombination. Indeed, crossing individuals near from each other in the variable space lead to an offspring that is generated near from the parents (in terms of the objective values). This increases the possibility of obtaining a non-dominated solutions and a diversified population. For continuous functions, the individuals that

are neighbours in the objective space are generally neighbours in the variable space. Our neighbourhood recombination method is presented in Algorithm 2.

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

Algorithm 2 The Neighbourhood recombination algorithm

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

4.1.3 The adaptive multi-operators strategy

When solving many-objective problems, MaOAs have the problem of finding the appropriate mutation and recombination operators according to the specificities of the problem to be solved. To overcome this weakness, we suggest varying the used operator adaptively. Indeed, the contribution of an operator in the previous iteration was taken into account for the adjustment of its probability of being selected during the current iteration. Thus, a probability of selection of use in the next generation is computed for each operator. This probability depends on its contribution. This “adaptive” mutation is dynamic since it is modifiable during the execution of the algorithm. In our case, a “directed adaptive” mutation is used. It utilizes the feedback information taken from the past generations to select operators in the future generations without affecting the probabilistic aspect of the operators. As a consequence, the new produced solutions are deterministically generated and guided by earlier individuals in the search space toward optimal regions. This proposed adaptive multi-operators neighbourhood strategy enhance improving the search and adapting it to the problem local characteristics. Moreover, it facilitates avoiding local optima and increasing the diversity by adaptively modifying the chromosomes values.

4.1.4 Application of the diversity strategies on the NSGA-III

The suggested adaptive multi-operator NSGA-III algorithm relies on the NSGA-III algorithm (Deb and Jain 2014) with an enhanced neighbourhood mutation and recombination process which adaptively integrates different mutation operators. This the

first time such a modification of the NSGA-III is proposed. Algorithm 3 illustrates the generation t of the proposed adaptive multi-operator NSGA-III algorithm.

| |
|--|
| <p>Input: H structured reference points Z^s or supplied aspiration points Z^a, parent population P_t</p> <p>Output: P_{t+1}</p> <p>01: // Initializations identical to the standard NSGA-III (Deb et al., 2014)</p> <p>02: $P_t' = \text{Niching_and_Neighbor_Based_Selection}(P_t)$</p> <p>03: $\text{ProbaRecombinationOp} \leftarrow \text{choosing_operator}()$</p> <p>04: $\text{ProbaMutationOp} \leftarrow \text{choosing_operator}()$</p> <p>05: $Q_t = \text{Neighborhood_Adaptive_Mutation}(P_t', \text{ProbaMutationOp});$</p> <p>06: $Q_t = \text{Neighborhood_Adaptive_Recombination}(Q_t, \text{ProbaRecombinationOp})$</p> <p>07: // The rest is the same as the standard NSGA-III algorithm</p> |
|--|

Algorithm 3 The generation t of the proposed adaptive multi-operator NSGA-III algorithm

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

| |
|--|
| <p>Input: The N operators</p> <p>Output: ProbaOp</p> <p>01: $\text{TotalContrib} \leftarrow 0$</p> <p>02: for $1 \leq \text{operator} \leq N$ do</p> <p>03: $\text{OpContrib} \leftarrow \text{solutionsInNextPopulation}(\text{operator}, P);$</p> <p>04: if $\text{OpContrib} \leq \text{threshold}$ then</p> <p>05: $\text{OpContrib} \leftarrow \text{threshold};$</p> <p>06: end if</p> <p>07: $\text{TotalContrib} \leftarrow \text{TotalContrib} + \text{OpContrib};$</p> <p>08: end for</p> <p>09: for $1 \leq \text{operator} \leq N$ do</p> <p>10: $\text{ProbaOp} \leftarrow \text{OpContrib} / \text{TotalContrib};$</p> <p>11: end for</p> |
|--|

Algorithm 4 *Choosing_operator()* procedure

The same changes are applied to the original versions of the MOEA/D, MOEA/DD and Two-Arch2 to take advantage of our adaptive multi-operators concept.

4.2 Including reduction methods based on machine learning for the 3D deployment problem

As an example of approaches used to overcome the complexity of many-objective problems, we can mention the dimensionality reduction which supposes the existence of redundant objectives in a given M -objective optimization problem.

In our works, as a reduction technique, we use the machine learning method MVU-PCA (“Principal Component Analysis and Maximum Variance Unfolding”) which is an offline correlation-based reduction method. The Machine Learning-based method (Deb et al. 2006) consists in using machine learning techniques, such as Principal Component Analysis (PCA) and Maximum Variance Unfolding (MVU), to eliminate respectively the dependencies of the second and higher order in the non-dominated solutions. MVU-PCA relies on a high-dimensional data structure which may be transformed to minimize the effect of noise (non-optimal solutions that can differ from the solutions defining the true PF) and dependencies (redundancy) between the different objectives.

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

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

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

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

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

output is the kernel matrix representing the kernel space to which the PCA method is applied.

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

4.3 Our proposed hybrid preference algorithm PI-EMO-PC-INK

Incorporating preferences aim at resolving the problem of low selection pressure of convergence by carrying out an ordering of preference over the non-dominated solutions. Because they interest in the search direction on the RoI, the interactive and a priori algorithms are more likely to pay more attention to preferred solutions and decrease the computational cost of the search process. A posteriori preference approaches might give a high number of solutions which are non-interesting to the DM. In our works, as an interactive preference reduction method, we use PI-EMO-PC (Sinha et al. 2014). Despite their numerous advantages, the major weakness of the interactive methods is that the algorithm used in such techniques need to interact frequently with the DM who can become tired, which leads, in some cases, to a misleading information about the preference information provided by the exhausted user (Gong et al. 2013). Thus, to overcome this limitation in our work, we suggest a hybrid preference process.

The progressive engagement of the preferences (interactive algorithms) is more efficient since it allows the DM to adjust his preferences during the intermediate generations of an algorithm (Sinha et al. 2013). Therefore, a model based on an enhanced PI-EMO-PC is used as a preference method in our work.

According to the results in Sects. 5–7, it is approved that the reduction procedure enhances the comportment of the used algorithms. Indeed, once the set of essential objectives is identified by the used reduction procedure, a questionable issue is about the structure of the initial used population for the preference procedure.

The aim is to choose a sub-population of solutions from the pool of population. This representative set could be either randomly generated; either the same as the population output of the used EMO at the generation G^p where the reduction procedure is executed; or an intermediate population where its individuals are picked progressively from the initial population until the population at the generation G^p (the generation with a progress value near to half as the pool population). The last choice, used in our model,

is the natural choice since it provides a balance between the search window of the DM and the computational efficiency.

Interactive preferences are the most interesting preference methods since they are dynamically injected into the selection process to continually guide the search for appropriate actions.

To design our proposed hybrid preference algorithm, we use a recent interactive preference procedure called PI-EMO-PC (progressively interactive EMO based on polyhedral cone).

Despite the fact that the PI-EMO-PC procedure was developed using the NSGA-II algorithm, it is a generic procedure which may be incorporated in any other multi-objective EMO algorithm. However, the PI-EMO-PC suffers from some drawbacks such as the need to know the number of DM calls in advance and the risk to have wrong directives if the DM becomes tired. To solve these problems, we introduce a hybrid preference procedure which requires fewer parameters and more flexibility when interacting with the DM.

In fact, most of preference studies presented in literature either focus on only a sub-set of the PF, which enhances the convergence but decreases diversity, or has a high computational time which rises exponentially as the number of objectives rises. To resolve this issue, we propose a hybrid method relying on combining implicit and explicit interactive preferences. Figure 3 illustrates the proposed hybrid method. Firstly, Ideal point and Nadir one are found respectively by using the Extremized-Crowded NSGA-II algorithm (EC-NSGA-II) (Deb et al. 2006) and by individually minimizing each objective in the search space. Then, an explicit algorithm based on PI-EMO-PC is executed if the DM has preferences. Otherwise, an implicit preference process aiming at finding knee regions based on Trade-off-based KR-NSGA-II (TKR-NSGAI) (Bechikh et al. 2011) is carried out. When using explicit algorithm based on PI-EMO-PC, if the DM is not satisfied but he becomes tired, the procedure performs the process to find the previously-indicated knee regions. Afterward, if the DM is not tired yet, he modifies the aspiration levels of the PI-EMO-PC in order to incorporate new information about his preferences. These processes (PI-EMO-PC and TKR-NSGAI) will be repeated if the DM is not satisfied or the maximum allowed number of permitted interventions is not reached. In both cases, when the DM becomes satisfied or the maximum allowed number of interventions is reached, the global process will be stopped.

The ideal point is a form of implicit DM preferences. It can be defined as the vector $z^I = (z_1^I, \dots, z_M^I)$ constituted by the best objective values of the search space Ω . The ideal point may be specified by individually minimizing each objective in the search space. Mathematically, the ideal objective vector is given by

$$z_m^I = \text{Min}_{x \in \Omega} f_m(x), \quad m \in \{1, \dots, M\} \quad (24)$$

The nadir point is another form of implicit DM preferences. It can be defined as the vector $z^N = (z_1^N, \dots, z_M^N)$ including the worst objective values on the PF. Mathematically, the nadir objective vector is given by

$$z_m^N = \text{Max}_{x \in P^*} f_m(x), \quad m \in \{1, \dots, M\} \quad (25)$$

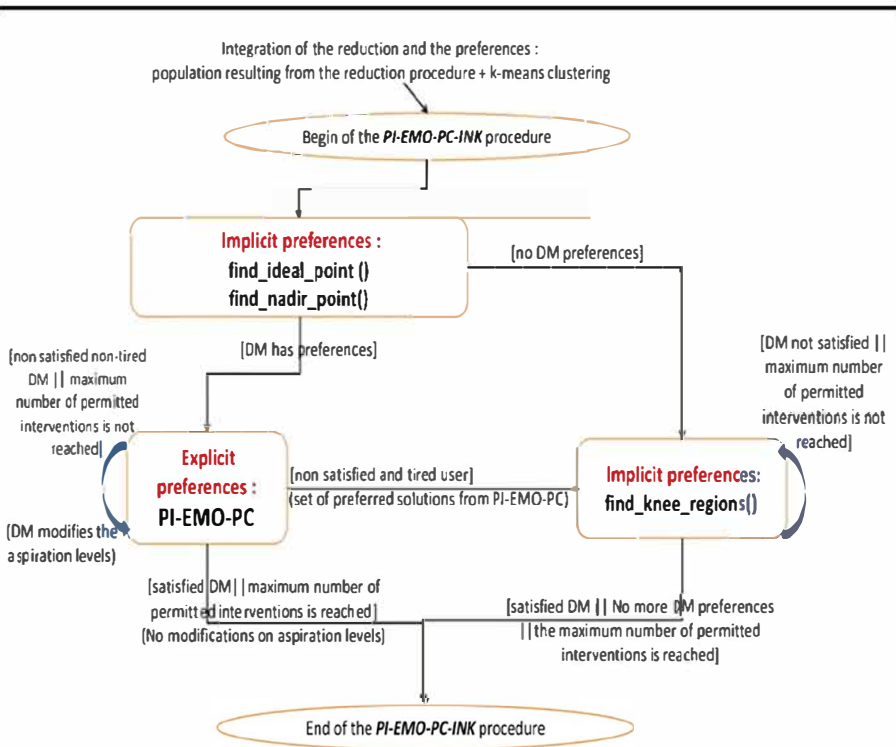


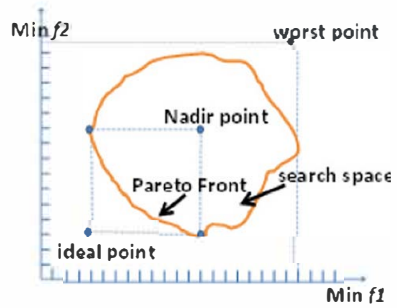
Fig. 3 The proposed hybrid preference algorithm (PI-EMO-PC-INK)

According to Branke et al. (2008), several interactive algorithms used the nadir point as a pre-requisite. However, estimating accurately the nadir point for many objective problems is an open research issue. In order to help the DM in expressing his preferences, Nadir point is applied so that each aspiration level will lie between the nadir value and the ideal one.

Ideal point and Nadir point assists the DM in expressing his preferences by identifying the range of the objective functions at the Pareto optimality stage. Both ideal and nadir points are employed to visualize the optimal Pareto front which facilitate comparing solutions especially for high dimension problems. Because it is used to avoid the worst instead of achieving the best, the Nadir point is considered as a more conservative point of view, compared to the use of the ideal point. Figure 4 illustrates the Nadir and ideal points of a two-objective problem.

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

Fig. 4 Nadir and ideal objective points



5 Numerical results

In this section, we present the assessment indicators and the parameters. Then, we demonstrate the efficiency of the suggested hybrid approach on real testbeds. In our tests, the implementations of the MOEA/DD, NSGA-III and Two_Arch2 in the PlatEvo framework (Tian et al. 2017) are used.

***Performance metrics** A research issues is questionable: What metrics should we choose when evaluating the performance of MaOPs?: The HV is necessary applied to evaluate the MaOPs because it does not require a prior-knowledge of the true Pareto-front which is important when benchmarking on real-world problems (our case). Besides, this metric gives a single measurement to assess both the spread and convergence of solutions. However, the IGD can be used with synthetic problems due to its low computational cost.

***Parameters** The choice of the parameters considerably affects the performance of the algorithms when solving problems. Hence, performing a set of experiments with several population sizes, operators, number of generations and objectives is necessary when testing each many-objective algorithm separately. The objective number (M) is set between 3 and 8, for the real world problem, and between 3 and 15 for the test functions. In all tested problems (the real world problem and the test functions), the best performance of each test is shown with a gray background. The considered EMO algorithms have different parameters. Unless a modification in the value of a parameter to test the impact of varying it, the parameter's common used values can be summarized as follows:

- The operators of reproduction: The crossover probability is $pc=0.9$. The mutation probability is $pm=1/n$. n is the problem dimensionality. Different operators are used. Among them: bit-flip, reversing and bit string for mutation; uniform, single point, n-point, multivariate and elitist for crossover.
- Population size: Several specifications for the number of weight vectors and the size of the population are used.
- The number of runs: Each algorithm is achieved 25 times (25 independent runs) with each configuration. Then, the various configurations are compared based on HV for our real-world problem. The bold values in Tables 4, 5, 6, 7 and 8 contain the best HV values.

-
- The stopping condition is a maximum number of solution evaluations (generations) between 400 and 1500.
 - The MOEA/D and MOEA/DD scalarizing function is PBI (0.5).
 - The size of the neighborhood is 20 and the probability of selection of a neighbor parent is 0.9.

As application, we use a 3D indoor deployment WSN optimization problem with eight objectives. As described in the modeling section (Sect. 3), this problem has 10 decision variables as inputs and fourteen objective values as outputs. In this section, due to the complexity of computing the HV, only eight objective values to be optimized are considered. The difficulty of the tested real-world problem (3D Deployment Problem) increases with the number of objectives. Since we apply a high complexity metric (HV) to evaluate our problem, (due to the unknown PF of our real-world problem), the number of the used objectives ranges from 3 to 8 objectives among the objectives cited in our modeling. Concerning the parameters of the problem, unless indicated, the same parameters detailed in the experimentation section (see Sect. 7) are used: number of nodes, average number of runs and nodes repartition.

5.1 Testing the influence of dependencies between the objectives

This following section aims to test the effect of interdependence between objectives. According to the Eqs. (1), (5), (7), (9), (11), (13), (14) and (16) in the proposed modeling (Sect. 3), there is similarity (then a correlation) between the objectives f_1 , f_3 and f_4 , and a less similarity between the objectives f_5 and f_6 . The population size is 1000 (a large population) which is run for different number of generations. No neighbor mating in the recombination. The objectives are dependent (a minimum of $N/2$ objectives are dependent for an experiment having N objectives). We employ 250 reference points for NSGA-III and MOEA/DD. To reduce the computational cost when calculating the HV, the dimension-sweep algorithm of Fonseca et al. (2006) is used to compute the HV. Tables 4 and 5 illustrate the obtained results.

From the obtained results, it can be concluded that, for different numbers of generations and objectives, NSGA-III is less efficient than MOEA/D and MOEA/DD generally outperforms other algorithms. Besides, when dependencies exist between the objectives, the HV increases. Especially for the MOEA/D having a higher relative degree of improvement compared to other algorithms.

5.2 Testing the influence of the population size

The values of HV are presented in this section for different sizes of population in order to assess the influence of varying the size of population on the behavior of MaOAs. No neighbor mating in the recombination. The objectives are dependent (a minimum of $N/2$ objectives are dependent for an experiment having N objectives). Besides, the number of reference points depends on the objectives number and the population size. Table 6 details the worst, average and best values of HV when varying the population number and the size population.

Table 4 Worst, average and best HV values with non-dependent objectives obtained using 15 independent runs

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

Table 5 Worst, average and best HV values with N ($N \geq \text{Obj Nbr}/2$) correlated objectives using 15 independent runs

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

For most numbers of objectives, better results were found by MOEA/D and MOEA/DD than NSGA-III. The results prove that extending the population size does not affect the ability of search of the MOEA/D. Contrariwise; the MOEA/D performance is degraded by the rise of size of the population until becoming inefficient with large population sizes. Hence, determining the appropriate size of the population according to the number of considered objectives is a relevant topic of research. An important observation is that the efficiency of MOEA/D cannot be influenced by the population size increase due to the multiple neighbors which may be replaced with newly-generated better off-spring.

Table 6 Worst, average and best HV values using various population sizes and various objectives numbers

| Obj Nbr | Population size | MOEA/D (PBI) | MOEA/DD | NSGA-III | Two_Arch2 | Reference points number (MOEA/DD, NSGA-III) |
|---------|-----------------|-----------------|-----------------|-----------------|-----------------|---|
| 4 | 100 | 0.956923 | 0.983461 | 0.973231 | 0.973682 | 90 |
| | | 0.956517 | 0.981027 | 0.972675 | 0.972987 | |
| | | 0.956208 | 0.980429 | 0.972089 | 0.972023 | |
| | 500 | 0.972863 | 0.985237 | 0.977863 | 0.985682 | 130 |
| | | 0.972165 | 0.984162 | 0.977258 | 0.976263 | |
| | | 0.972022 | 0.984103 | 0.977037 | 0.976044 | |
| | 1000 | 0.984426 | 0.983652 | 0.978863 | 0.976498 | 255 |
| | | 0.976412 | 0.981426 | 0.978574 | 0.976422 | |
| | | 0.976328 | 0.976124 | 0.976231 | 0.976346 | |
| | 1200 | 0.976664 | 0.986213 | 0.978683 | 0.977023 | 280 |
| | | 0.976586 | 0.985897 | 0.978764 | 0.976986 | |
| | | 0.972343 | 0.985251 | 0.978037 | 0.976431 | |
| 1400 | 0.976874 | 0.986852 | 0.978985 | 0.987875 | 290 | |
| | 0.976758 | 0.986238 | 0.978583 | 0.987244 | | |
| | 0.974032 | 0.985140 | 0.978362 | 0.977032 | | |
| 8 | 100 | 0.969369 | 0.970145 | 0.959863 | 0.952894 | 90 |
| | | 0.969292 | 0.969233 | 0.959467 | 0.952236 | |
| | | 0.969083 | 0.969002 | 0.959302 | 0.952035 | |
| | 500 | 0.969963 | 0.969786 | 0.960869 | 0.969878 | 230 |
| | | 0.969643 | 0.969645 | 0.960098 | 0.952543 | |
| | | 0.969354 | 0.969423 | 0.959326 | 0.952132 | |
| | 1000 | 0.969886 | 0.971841 | 0.966876 | 0.952886 | 320 |
| | | 0.969815 | 0.969822 | 0.966525 | 0.952835 | |
| | | 0.969702 | 0.969723 | 0.966234 | 0.952803 | |
| | 1200 | 0.969894 | 0.970274 | 0.967964 | 0.970623 | 350 |
| | | 0.969831 | 0.969962 | 0.967663 | 0.953195 | |
| | | 0.969063 | 0.969146 | 0.960022 | 0.952678 | |
| 1400 | 0.969965 | 0.970988 | 0.968326 | 0.971589 | 350 | |
| | 0.969887 | 0.970231 | 0.967989 | 0.953651 | | |
| | 0.969576 | 0.969862 | 0.960374 | 0.953233 | | |

5.3 Testing the influence of using neighborhood mating and adaptive recombination

We examine in this section the effect of the suggested strategy for mating near parent with an adaptive mutation and recombination (see Sect. 4.1). The behavior of each algorithm is assessed using average values of HV over 15 runs. The size of population in MOEA/D is 1000. Neighbor mating is achieved in the recombination process, the objectives are dependent and the number of reference points is 100. Table 7 illustrates the obtained results.

The collected results with different numbers of objectives indicate that the neighborhood mating and the use of adaptive operators considerably enhance the search

Table 7 Worst, average and best values of HV with adaptive operators

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

performance. In fact, better results are recorded for different numbers of objectives on the MOEA/DD. Obviously, when the number of objectives rise, the advantage of the MOEA/DD over the MOEA/D becomes clearer. Nevertheless, MOEA/D improves more considerably the average values of HV (using or without recombination of similar parent), compared to other algorithms. Besides, experimental results show that mating similar parents enhances the diversity and does not deteriorate the convergence.

5.4 Testing the influence of hybridizing the used EMOs with a dimensionality reduction approach

In this section, we investigate the effect of incorporating our proposed approach for dimensionality reduction. In this set of experiments, the HV is calculated and the taken size for the population is 1000. Adaptive mutation and recombination operators are used with neighbor parents mating. The objectives are dependent (a minimum of $N/2$ objectives are dependent for an experiment having N objectives). 8 correlated objectives are employed. 250 reference points are applied for NSGA-III. From the results presented in Table 8, for four and eight objectives, the HV values found when using dimensionality reduction approach are higher than those obtained without using this method due to the reduction in the number of objectives from eight to five in the case of our real-world problem. Moreover, the improvement rate of the MOEA/D clearly exceeds those of other algorithms.

5.4.1 Comparing with other dimensionality reduction methods: the Feature Selection

In our model, we used two dimensionality reduction methods (NLMVU-PCA and L-PCA). To better clarify the efficiency of the suggested model, we compare it with another dimensionality reduction method: the Feature Selection (FS) which is an unsupervised feature selection procedure. The FS is an algorithm that establish an ideal learning model to minimize the dimensionality of the feature space by identifying a

Table 8 Worst, average and best HV values obtained before and after applying the dimensionality reduction

| | Initial Obj Nbr | Obj Nbr after reduction | MOEA/D (PBI) | MOEA/DD | NSGA-III | Two_Arch2 |
|------------------------------------|--------------------|-------------------------------|-----------------|-----------------|----------|-----------------|
| Without reduction | 4/5 | 4/5 | 0.984426 | 0.983652 | 0.978863 | 0.976498 |
| | | | 0.976416 | 0.981426 | 0.978574 | 0.976422 |
| | | | 0.976328 | 0.976124 | 0.976231 | 0.976346 |
| Using L- PCA/NL- MVU- PCA | | 3 | 0.994975 | 0.982897 | 0.982896 | 0.989352 |
| | | | 0.991264 | 0.982542 | 0.982251 | 0.987144 |
| | | | 0.980023 | 0.982231 | 0.982033 | 0.986021 |
| Using the feature selection | | 3 | 0.993284 | 0.982962 | 0.983202 | 0.990249 |
| | | | 0.990737 | 0.982615 | 0.982080 | 0.987808 |
| | | | 0.979601 | 0.980967 | 0.981949 | 0.986645 |
| Without reduction | 8 | 8 | 0.969886 | 0.971841 | 0.966876 | 0.952886 |
| | | | 0.969815 | 0.969822 | 0.966525 | 0.952835 |
| | | | 0.969702 | 0.969723 | 0.966234 | 0.952803 |
| Using L- PCA/NL- MVU- PCA | | 4 | 0.971978 | 0.984986 | 0.969897 | 0.954237 |
| | | | 0.970951 | 0.984158 | 0.961152 | 0.953654 |
| | | | 0.970236 | 0.983943 | 0.960364 | 0.953028 |
| Using the feature selection | | 5 | 0.978234 | 0.974823 | 0.961236 | 0.950893 |
| | | | 0.975231 | 0.974676 | 0.960743 | 0.950239 |
| | | | 0.974328 | 0.973284 | 0.958201 | 0.948327 |

minimum set of inter-dependent necessary ‘features’ from an initial data groups. In our case, a feature is considered as an objective. The choice of the Feature Selection as a dimensionality reduction method to compare with; is motivated by the nature of the 3D indoor deployment which is an objectives-dependent problem. The used FS algorithm relies on the algorithm proposed by Mitra et al. (2002).

To test the effect of the use of the FS, the best, average and worst values of HV of the evolutionary algorithms are measured after the incorporation of the FS. Table 8 shows the obtained results.

5.4.2 Application of the NL-MVU-PCA and L-PCA

NL-MVU-PCA is a nonlinear objective reduction relying on the “Principal Component Analysis” and the “Maximum Variance Unfolding”. We briefly present the nonlinear objective reduction based on the NL-MVU-PCA algorithm. Indeed, using an initial set of objectives $SO_i = \{f_1, \dots, f_M\}$, the NL-MVU-PCA finds the sub-set of essential objectives SO_e . For this purpose, a set of steps are achieved: First, as an input data, the NL-MVU-PCA consider the objective vectors from the non-dominated set of EMO solutions. Then, the process searches for the principal components (significant variance) in the data. Afterwards, it identifies the conflicting subset of objectives along these principal components. Finally, it discards the inter-dependent objectives from the final set. These steps are iteratively achieved until the necessary subset of objective in a couple of successive iterations is reduced to two objectives or stills the same.

Table 9 The correlation Matrix R on the first iteration

| | f_1 | f_2 | f_3 | f_4 | f_5 | f_6 | f_7 | f_8 |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|
| f_1 | 1 | -0.458 | 0.896 | 0.985 | -0.258 | 0.885 | 0.875 | -0.647 |
| f_2 | -0.458 | 1 | -0.521 | -0.678 | -0.735 | -0.613 | 0.365 | -0.647 |
| f_3 | 0.896 | -0.521 | 1 | 0.982 | -0.385 | 0.997 | 0.354 | -0.647 |
| f_4 | 0.985 | -0.678 | 0.982 | 1 | -0.392 | 0.839 | 0.365 | -0.647 |
| f_5 | -0.258 | -0.735 | -0.385 | -0.392 | 1 | -0.264 | 0.238 | -0.647 |
| f_6 | 0.885 | -0.613 | 0.997 | 0.839 | -0.264 | 1 | 0.364 | -0.647 |
| f_7 | 0.875 | 0.365 | 0.354 | 0.365 | 0.238 | 0.364 | 1 | -0.647 |
| f_8 | -0.647 | -0.647 | -0.647 | -0.647 | -0.647 | -0.647 | -0.647 | 1 |

Table 10 The kernel Matrix K

| | f_1 | f_2 | f_3 | f_4 | f_5 | f_6 | f_7 | f_8 |
|-------|--------|--------|--------|--------|---------|---------|---------|---------|
| f_1 | 4.568 | -4.521 | 4.895 | 6.552 | -7.353 | 9.652 | 6.548 | -6.365 |
| f_2 | -4.521 | 6.021 | -4.257 | -6.215 | -7.985 | -9.245 | 9.253 | -6.365 |
| f_3 | 4.895 | -4.257 | 6.892 | 6.812 | -7.154 | 11.246 | 12.568 | -6.365 |
| f_4 | 6.552 | -6.215 | 6.812 | 8.453 | -7.554 | 11.246 | 12.568 | -8.852 |
| f_5 | -7.353 | -7.985 | -7.154 | -7.554 | 12.258 | -11.246 | 18.258 | -8.254 |
| f_6 | 9.652 | -9.245 | 11.246 | 11.246 | -11.246 | 18.254 | 18.891 | -8.852 |
| f_7 | 6.548 | 9.253 | 12.568 | 12.568 | 18.258 | 18.891 | 25.547 | -18.255 |
| f_8 | -6.365 | -6.365 | -6.365 | -8.852 | -8.254 | -8.852 | -18.255 | 21.541 |

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

Step 1 (Computing the correlation and the kernel matrix) Relying on the input data, the correlation matrix R (for linear objective reduction (L-PCA)) is plotted, the kernel matrix K (for nonlinear objective reduction (NL-MVU-PCA)) and its principal component (eigenvectors and eigenvalues) are computed. According to Sinha et al. (2013), $R = (1/M).XX^T$ where M is the number of objectives and X is the input data. K is also calculated according to the formulation in Sinha et al. (2013). Tables 9, 10, 11 and 12 illustrate the values of the matrix R, K as well as their Eigenvectors and eigenvalues.

Step 2 (Eigenvalue Analysis) consists in identifying the set of the important objectives in the initial set of objectives by performing the eigenvalue analysis that

Table 11 Eigenvectors and eigenvalues of the matrix R

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

Table 12 Eigenvectors and eigenvalues of the matrix K

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

Table 13 Eigenvalue Analysis for L-PCA

| PCA (N ^o) | Variance (%) | Cumulative (%) | Selected objectives | | | | | | | | | |
|-----------------------|--------------|----------------|---------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 1 | 64.6 | 64.60 | | | | | <i>f2</i> | | <i>f5</i> | <i>f6</i> | <i>f7</i> | <i>f8</i> |
| 2 | 22.1 | 97.09 | | <i>f1</i> | <i>f2</i> | <i>f3</i> | <i>f4</i> | <i>f5</i> | <i>f6</i> | <i>f7</i> | <i>f8</i> | |
| 3 | 8.4 | 99.62 | | | <i>f2</i> | | <i>f4</i> | | <i>f6</i> | <i>f7</i> | <i>f8</i> | |
| 4 | 0.3 | 99.99 | | | <i>f2</i> | | | <i>f5</i> | | <i>f7</i> | <i>f8</i> | |

identifies the principal components (directions of significant variance) in the data. Table 13 (Table 14, respectively) depicts the eigenvalue analysis for linear objective reduction (non-linear objective reduction, respectively).

Step 3 (Analysis of the Reduced Correlation Matrix) consists in identifying the set of identically-dependent subsets by carrying out the reduced correlation matrix analysis. The important objectives in each subset are retained and other objectives are discarded, which allows further reduction of the objective set obtained after step 2. Table 15 (Table 16, respectively) represents the RCM analysis for linear objective reduction (non-linear objective reduction, respectively). According to Saxena et al. (2013), $T_{cor} = 1.0 - e_1.(1 - M'/M)$ where M' refers to the number of needed principal components to account for 95.4% variance, M is the number of objectives.

Table 14 Eigenvalue Analysis for NL-MVU-PCA

| PCA (N ^o) | Variance (%) | Cumulative (%) | Selected objectives | | | | | | |
|-----------------------|--------------|----------------|---------------------|------------|------------|------------|------------|------------|------------|
| 1 | 54.8 | 54.80 | <i>f</i> 1 | <i>f</i> 2 | <i>f</i> 3 | <i>f</i> 5 | <i>f</i> 6 | | |
| 2 | 27.6 | 94.45 | <i>f</i> 1 | | | <i>f</i> 4 | <i>f</i> 5 | <i>f</i> 7 | <i>f</i> 8 |
| 3 | 4.8 | 99.99 | | <i>f</i> 2 | | | <i>f</i> 5 | <i>f</i> 7 | <i>f</i> 8 |
| 4 | 0.2 | 99.99 | | <i>f</i> 2 | <i>f</i> 3 | <i>f</i> 4 | | <i>f</i> 7 | <i>f</i> 8 |

Table 15 RCM analysis for L-PCA

| | |
|--|---|
| Potential identically correlated set(s) | { <i>f</i> 1, <i>f</i> 3, <i>f</i> 4, <i>f</i> 6} |
| T _{cor} (correlation threshold) | 1.0–0.646(2/8) = 0.8385 |
| Identically correlated set(s) | { <i>f</i> 1, <i>f</i> 3, <i>f</i> 4, <i>f</i> 6} |

Table 16 RCM analysis for NL-MVU-PCA

| | |
|--|---|
| Potential identically correlated set(s) | { <i>f</i> 1, <i>f</i> 3, <i>f</i> 4, <i>f</i> 6} |
| T _{cor} (correlation threshold) | 1.0–0.548(3/8) = 0.7945 |
| Identically correlated set(s) | { <i>f</i> 1, <i>f</i> 3, <i>f</i> 4, <i>f</i> 6} |

Table 17 Selection scheme for L-PCA

| | $e_1 =$ 0.646 V_1 | $e_2 =$ 0.221 V_2 | $e_3 =$ 0.084 V_3 | $e_4 =$ 0.003 v_4 | Objective selection score |
|------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------------|
| <i>f</i> 1 | 0.215 | 0.886 | 0.568 | 0.638 | 0.458 |
| <i>f</i> 1 | – 0.568 | – 0.322 | 0.546 | – 0.457 | 0.462 |
| <i>f</i> 1 | 0.585 | 0.662 | 0.531 | – 0.891 | 0.483 |
| <i>f</i> 1 | 0.689 | – 0.211 | 0.284 | – 0.457 | 0.494 |

Step 4 (Selection scheme) consists in identifying the most important objective in each set by applying the selection scheme. Table 17 (Table 18, respectively) demonstrates the selection scheme for linear objective reduction (non-linear objective reduction, respectively).

Step 5 (Computation of the error) is to measure the error of the proposed framework in one iteration. This measure calculates the unaccounted left variance when discarding the objectives constituting the redundant objective set. According to the

Table 18 Selection scheme for NL-MVU-PCA

| | $e_1 =$ 0.548 V_1 | $e_2 =$ 0.276 V_2 | $e_3 =$ 0.048 V_3 | $e_4 =$ 0.002 v_4 | Objective selection score |
|------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------------|
| <i>f</i> 1 | – 0.234 | 0.056 | 0.448 | – 0.253 | 0.238 |
| <i>f</i> 1 | 0.665 | – 0.094 | 0.125 | 0.151 | 0.295 |
| <i>f</i> 1 | 0.652 | 0.114 | 0.356 | 0.198 | 0.684 |
| <i>f</i> 1 | 0.745 | 0.146 | 0.651 | 0.235 | 0.793 |

equation proposed in Saxena et al. (2013), the error for the L-PCA (the NL-MVU-PCA, respectively) is equal to 0.000239 (0.000458, respectively).

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

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

In this section, we measure the effect of applying our proposed approach to incorporate both dimensionality reduction method and user preference one. In this set of experiments, the HV is calculated with a large population (1000). Adaptive mutation/recombination is employed with neighbor parents mating. The objectives are dependent (a minimum of $N/2$ objectives are dependent for an experiment having N objectives). 8 correlated objectives are used. We use 250 reference points for NSGA-III. After applying reduction approach, the preference is applied on a reduced set of objectives. Tables 19 and 20 show the final solutions specifications (using Two_Arch2 as an EMO). Each run has a different initial population, which is the result of applying our reduction procedure on the concerned EMO. d_s is a user-defined parameter representing the presumed enhancement in the solutions obtained from the actual best solution relying on the value function, and $d_s = 0.01$. $TD_{Max} = 30$ is the maximum number of calls of the preference information introduced by the DM.

After testing the algorithms proposed in our approach with the deployment problem, we also test them using a real experimental prototyping system (Arduino 2018; Van Den Bossche et al. 2016) with the same problem, then using instances of the theoretical DTLZ problems (Deb et al. 2005).

***Comparing with other preference methods: the PI-EMO-VF** By comparing the results obtained in Table 21 with those in Table 20, it is clearly shown that the PI-EMO-PC achieves better results than PI-EMO-VF since it gives better accuracy with less number of calls.

5.6 Comparing with random search and independent objectives

In this section, the performance of the four EMOs applied to our hybrid scheme is compared with the random search and the independent optimization of each objective function separately. The same parameters of the previous section are used. Table 22 shows a comparison of the worst, average and best values of the HV of the EMOs with random search and independent optimization of objectives (for consistency reasons and since three objectives are redundant $[f_1, f_3, f_4]$, only five from the eight objectives are considered).

From the results presented in Table 22, for four and eight objectives, the HV values found when optimizing each objective separately are higher (thus better) than those obtained using our hybrid scheme. However, all the optimization algorithms used with our hybrid scheme outperform the random search.

Table 19 Median obtained solutions (objective values)

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

Table 20 Median distance of the solutions obtained from most preferred solutions, using PI-EMO-PC

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

Table 21 Median distances of the solutions obtained from the most preferred solutions, using PI-EMO-VF

| | MOEA/D | NSGA-III | Two_Arch2 | MOEA/DD |
|--------------------------------|-------------------|----------|-----------|---------|
| Accuracy | 0.201 | 0.382 | 0.424 | 0.019 |
| Number of function evaluations | 6652 | 8096 | 8284 | 5987 |
| Number of required DM calls | TD _{Max} | 32 | 29 | 22 |

Table 22 Comparing the HV values of EMOs with random search and independent optimization of objectives

| Initial Obj Nbr | Obj Nbr after reduction | Our hybrid scheme applied to: | | | | Random search | Independent optimization of each objective function | | | | |
|--------------------|-------------------------------|-------------------------------|----------|----------|-----------|------------------|---|----------|----------|----------|----------|
| | | MOEA/D (PBI) | MOEA/DD | NSGA-III | Two_Arch2 | | f_2 | f_5 | f_6 | f_7 | f_8 |
| 4/5 | 3 | 0.994975 | 0.982897 | 0.982896 | 0.989352 | 0.893512 | 0.992024 | 0.991985 | 0.995256 | 0.994907 | 0.993223 |
| | | 0.991264 | 0.982542 | 0.982251 | 0.987144 | 0.831033 | 0.991397 | 0.990203 | 0.995002 | 0.994484 | 0.992685 |
| | | 0.980023 | 0.982231 | 0.982033 | 0.986021 | 0.830686 | 0.991092 | 0.990579 | 0.994451 | 0.994023 | 0.992022 |
| 8 | 4 | 0.971978 | 0.984986 | 0.969897 | 0.954237 | 0.890631 | 0.989236 | 0.990586 | 0.992643 | 0.990629 | 0.987898 |
| | | 0.970951 | 0.984158 | 0.961152 | 0.953654 | 0.889367 | 0.988638 | 0.990123 | 0.991245 | 0.990016 | 0.987025 |
| | | 0.970236 | 0.983943 | 0.960364 | 0.953028 | 0.887236 | 0.988022 | 0.989651 | 0.991026 | 0.988028 | 0.986094 |

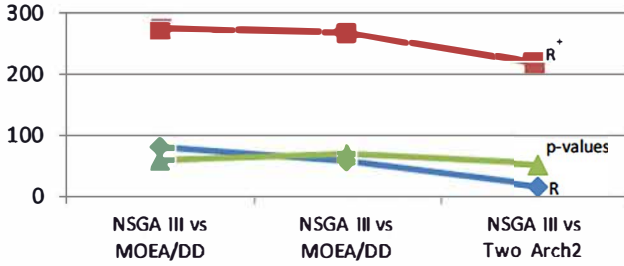


Fig. 5 p values, R^+ and R^- of the algorithm's pairs

Other results (which are not illustrated here due to the limited space) prove that the performance of the random search is enhanced when using independent objectives (without reduction).

6 Statistical and complexity analysis

6.1 Statistical analysis

In this section, we assess the difference in the distributions of indicator values obtained by the different algorithms using a statistical signed ranks test. NSGA-III is used as a control method and the same values for the algorithm's common parameters are used. We suppose two hypotheses in order to apply the statistical tests: A null hypothesis H_0 which imply that there is no difference between the compared algorithms. An alternative hypothesis H_1 which imply that a difference exists between the compared algorithms. We determine the level of discarding the hypothesis using a significance level (named α). The signs R^- and R^+ between NSGA-III and two other algorithms (MOEA/DD and Two_Arch2) are calculated. Then, the associated p -values of the algorithms are computed for each pair of algorithms. Figure 5 shows the p -values, R^- and R^+ of the pairs of the algorithms computed by the Wilcoxon Test. According to the Fig. 5, Two_Arch2 outperforms the control method NSGA-III (with a level of significance $\alpha = 0.1$), and outperforms MOEA/DD (with $\alpha = 0.05$).

6.2 Computational complexity and runtime analysis

An analysis of the computational and temporal complexity is important to evaluate the behavior of the proposed algorithms in terms of computing time.

***Computational complexity of algorithms** The computational complexity reflects the increase rate of the execution time in relation to the population. Table 23 shows an average execution time of 25 runs of the algorithms on the 3-objectives DTLZ1 problem. The MOEA/DD has the best (lowest) execution time with a comparable performance for MOPSO and Two_Arch2; while NSGA-III is the worst algorithm in terms of temporal complexity.

| Table 23 Average of execution time on the 3-objective DTLZ1 problem | Two_Arch2 | NSGA-III | MOPSO | MOEA/DD |
|---|-----------|----------|----------|----------|
| | | 6.37e+01 | 1.64e+02 | 7.44e+01 |

Table 24 Computational complexity of the algorithms on the M-objective DTLZ1

| Two_Arch2 | NSGA-III | MOEA/D | | MOPSO (M ≤ 3) | Two-stage MaOPSO [28] | MOEA/DD |
|-----------|-----------|-----------|--------------|----------------|---|--|
| | | M ≤ 3 | M > 3 | | | |
| $O(Mn^2)$ | $O(Mn^2)$ | $O(Mn^3)$ | $O(e^{M*n})$ | $O(Mn \log n)$ | $O(M(n_1 + L^2 n_2)) \cong O(Mn^3)$ L: the archive bounded-size, n ₁ , n ₂ : population sizes at stages I & II | Authors do not indicate its exact complexity |

Table 25 Execution time in relation with the number and dependence between objectives

| | | Average execution time (in seconds) | | | |
|--------------|------------------|-------------------------------------|---------|-----------|---------|
| | | NSGA-III | MOEA/DD | Two-Arch2 | MOEA/DD |
| 3 objectives | Dependent objs | 164 | 62 | 63.7 | 74.4 |
| | Independent objs | 152 | 50 | 78 | 71 |
| 5 objectives | Dependent objs | 171 | 77 | 72 | 87 |
| | Independent objs | 268 | 135 | 256 | 139 |

However, a low execution time does not necessary indicate a low computation complexity. Table 24 shows the computational complexity of the used algorithms on the M-objective test problem DTLZ1 (n is the number of individuals in the population).

Table 24 illustrates that MOEA/D has the worst (highest) computational complexity while other algorithms shows a similar performances.

***Testing the influence of the number and dependence between objectives on the execution time** Table 25 shows the effect of changing the number and the dependence between objectives on the overall execution time of the tested algorithms.

Table 25 clearly shows that the number and dependencies between the objectives affect the execution time. NSGA-III is the worst algorithm in term of the average needed execution time. Moreover, another interesting observation is recorded concerning the similar performances of the algorithms with three independent objectives and five dependent objectives: This indicated the advantage of the reduction of objectives.

To sum up, the complexity analysis assesses the behavior of the different tested algorithms. It shows that the execution and computation time can be influenced by different parameters such as the number and dependency between objectives. Further

tests can be performed regarding the influence of the size of the initial population and the number of nodes on the execution time.

7 Experimental tests on real testbeds

To evaluate the robustness and the efficiency of the protocols, technologies and models in real environments, real platforms called testbeds can be used. Indeed, simulations and theoretical calculations fail to reproduce the physical characteristics of the real-world environments; hence the current tendency to reduce the differences between theory and practice by testing the new algorithms and solutions in real environments with experiments carried out on testbeds. In this study, we propose a testbed composed of 36 nodes, called Ophelia. Using this personal testbed, different advantages are envisaged:

- *Conformity to reality* A personal testbed like Ophelia is based on tests in a real context of use unlike test platforms such as FIT/IoT-Lab (IoTLab 2019) which offers tests with a large number of nodes that are aligned or uniform on a grid.
- *Reproducibility* since Ophelia relies on open-source tools, such as OpenWiNo and Arduino, it is easy to manage and reproduce the obtained results by other research teams. Indeed, Ophelia supports different physical layers and various types of sensors, which facilitates the deployment of nodes and the prototyping task.
- *Heterogeneity of nodes* Ophelia supports three different types of nodes (DecaWino, WiNoLoRa and TeensyWiNo). Thanks to its compliance with open hardware and software, the WiNo architecture allows integrating foreign libraries to manage the deployed nodes, which enables it to support a wide variety of nodes.
- *A distributed deployment* Ophelia consists of 36 nodes deployed in several buildings and locations in a campus of $200 * 200 \text{ m}^2$.
- *Easy use and deployment* The nodes in our Ophelia testbed are manipulated (erasing data, updating) using OpenWino and the execution of the protocols stack is done via the usb interface of the nodes or by executing a command line from the console. Moreover, WiNo nodes are compatible with revolutionary transmission modes (UWB, LoRa...) and most standard physical layers, which makes the design and customization of the network as well as the replacement of the physical layer easier and more realistic.
- *Real use* WiNos nodes have small size, low power consumption rate and easy attachment to a mobile system or a person, which makes them an ideal component for the IoT and the prototyping of communicating objects.

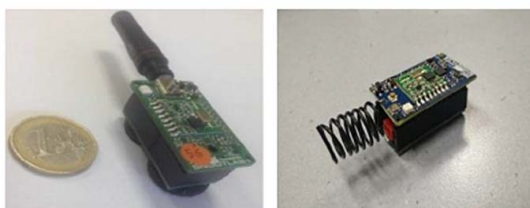
7.1 Experimental parameters

An Intel Core i5-6600 K 3.5 GHz computer is used to test the algorithms. Physical layer is based on a 433 MHz implementation. The applied access method is the non-coordinated CSMA/CA of the IEEE 802.15.4 protocol, and the routing layer relies on the reactive Ad hoc On-demand Distance Vector protocol. The parameters considered in our experiments are illustrated in Table 26.

Table 26 Parameters of the experiments

| | |
|---|-------------------------------------|
| Nodes repartition | 6 sites on 200 * 200 m ² |
| Nodes number | 36 (29 fixed, 6 nomad, 1 mobile) |
| Sensing range | 8 m |
| Transmission range | 7 m |
| Frame error rate (FER) | 0.01 (initially) |
| Received signal strength indicator (RSSI) | 100 (initially) |
| Average number of runs | 25 experiments |
| Bit rate | 256 kbps |
| Modulation model | 125 kbit/s GFSK |
| Antenna model | transceiver RFM22 |
| Modem configuration | 12 # GFSK_Rb2Fd5 |
| Frequency | 434.79 MHz |
| Tx power | 7 (the max of RFM22) |
| Message-number | 1000 |
| Message-length | 16 |
| Message-wait | 5 |
| Reception gain | 50 mA |
| Transmission power | 100 mW |

Fig. 6 The Teensy WiNo used nodes



7.2 Ophelia testbed nodes and used tools

The deployed TeensyWiNo nodes are WiNoRF22 nodes equipped with brightness and temperature sensors to which other sensors are added (gyrometer, acceleration or pressure). They give access to low layers in order to manage the access time to the medium, the sleep, the awakening and CPU time; and the management of the restricted memory. WiNo nodes represent a hardware platform able to host different protocols with real-time constraints (several months of use using two AAA batteries). The installed TeensyWiNo nodes are shown in Fig. 6 and their technical characteristics are illustrated in Table 27.

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

The following tools are used:

Table 27 Technical characteristics of the used TeensyWiNo nodes

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

- *Arduino 1.6.1* (Arduino 2018) is an open hardware and software platform employed by the “WiNo” sensors to prototype modules and transfer the sketches. Teensyduino is an Arduino added module used to run these sketches.
- *OpenWiNo* (Van Den Bossche et al. 2016) is an open tool applied to prototype and evaluate the performance of WSN and IoT protocols in different layers (MAC, NWK...) and run them on real WiNos nodes. The simplicity of using OpenWiNo lies in changing the physical layer where it is sufficient to modify the transceiver. This is very practical in open-hardware environments as it is the case in the context of IoT.
- *Ophelia* relies on Openwino, Arduino, the deployed Teensywino sensors and a web user interface used to remote the access to the testbed and to execute sketches on the nodes.

7.3 Experimental scenario and results

We use 30 stationary sensors with known positions and deployed initially. Positions depend on the users application needs. The number of nomad nodes to add is limited to six. The positions of the latter nodes are to be determined by the used optimization methods. A mobile node is used. In order to measure the influence of the selected locations of the nomad sensors on the overall performance of the network, the following scenario of the experiments is repeated several times: At first, all sensors are flashed and the parameters of the initial configuration (such as the power of transmission) are sent. Afterwards, the mobile sensor sends a broadcast to all nodes. The RSSI and FER rates issued from and received by each node are considered. After a predefined waiting time, another transmitter is chosen and other nodes receive. These steps are repeated until performing 36 experiments. At the end, a couple of connectivity matrices, combining the FER and RSSI means between the nodes, are created. The average number of neighbors of each node is deduced from these two matrices. In our experiments, two nodes are considered neighbors if and only if the mean rate of the RSSI (and FER, respectively) recorded between these two nodes is greater (lower, respectively) than a pre-defined threshold equal to 100 (0.1, respectively). Due to the stochastic aspect of the used optimization algorithms, the use of a statistical test with several executions is necessary to assess their behavior. Therefore, the mean values in our tests are calculated relying on 25 runs of the algorithms. Figure 7 shows the 3D indoor deployment of nodes in one of the sites used in experiments. In fact, blue nodes are the nomad ones while red nodes are the fixed ones.¹

¹ Color should be used for Fig. 7 in print.

Fig. 7 The indoor 3D layout of the sensors in one of the used sites (Color figure online)

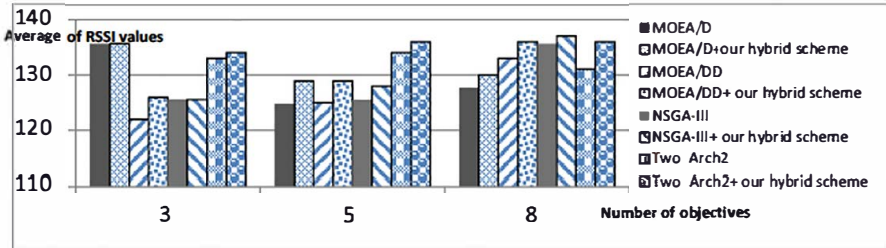
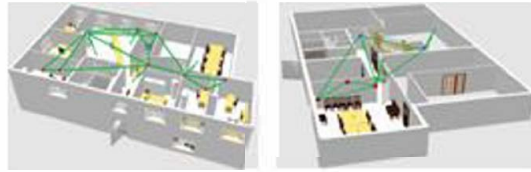


Fig. 8 Average values of RSSI, for different objective's number

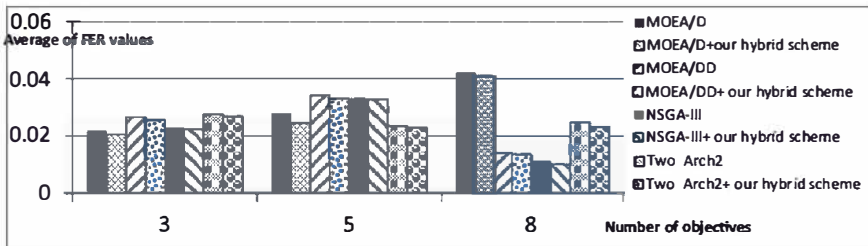


Fig. 9 Average FER values, for different number of objectives

Comparing the RSSI values To assess different objectives, like the localization, the connectivity or the quality of links, the RSSI metric is calculated. Our experiments are based on a hybrid localization model which combines the RSSI information and the Distance-Vector Hop protocol. Indeed, the higher the RSSI value, the better the localization will be. Figure 8 shows the RSSI average values (a convertible to dBm measure ranging from 0 to 256) exchanged between the nodes for different numbers of objectives.

Comparing the FER rates To measure the coverage state and the quality of links among nodes, the FER metric is calculated. Indeed, the lower the FER value, the better the coverage will be. To assess FER values for each pair of nodes, an average value deduced based on four values is considered. 10 s of wait is used between the four taken values. Figure 9 shows the average FER values between nodes for different numbers of objectives.

Comparing the neighbor's number To evaluate the connectivity and the degree of use of the network, the average number of each node neighbors is computed. Indeed, the previous concept of neighboring based-on RSSI and FER is used. Figure 10 shows the average number of neighbors of nodes and for different numbers of objectives.

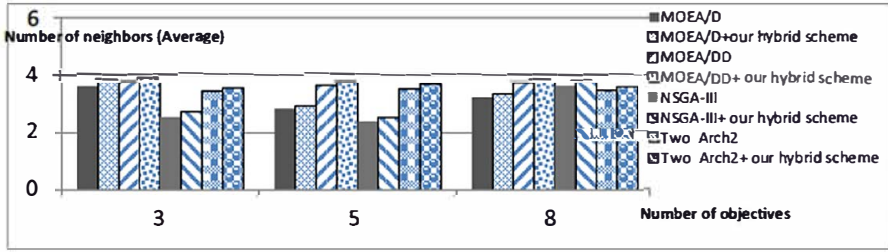


Fig. 10 Average number of neighbors, for different number of objectives

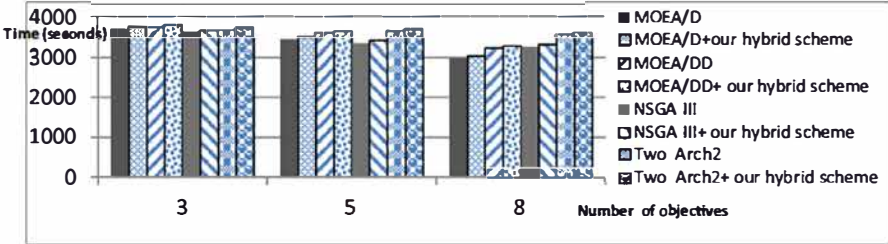


Fig. 11 Average network lifetime according to the number of objectives

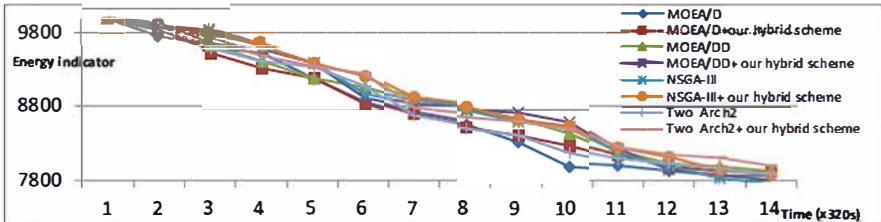


Fig. 12 Average energy consumption levels, as a function of time

Comparing the network lifetime Figure 11 indicates, for a set of number of objectives, the network lifetime which is taken as the time needed by the first node of the network to become out of energy.

Comparing the amount of consuming energy Figure 12 shows the variation of the network energy consumption according to the time. The average of the energy rates of nodes is calculated after the use of the nomad nodes.

7.4 Interpretations and discussion

After performing the experiments, several findings can be considered:

- These tests prove the suitability of the optimization paradigms for real-world contexts with real experiments. It shows the theoretical findings obtained by applying the tested algorithms.

-
- Our experiments show that a link between two nodes can have a high FER and a good RSSI at the same time. Thus, the FER and RSSI rates are not always inversely proportional.
 - By studying the behavior of the tested methods before and after the application of the proposed hybridization scheme, the prototyping results demonstrate that NSGA-III is often better than MOEA/DD on FER and RSSI amounts. Thus, the NSGA-III is considered more efficient in guaranteeing the localization, the coverage and the link quality, while MOEA/DD is better used to satisfy the average number of neighbors and the network lifetime.
 - Consistent with our numerical results in Sect. 5, the experiments assert that the effectiveness of the algorithms is related to the number of objectives to be optimized. Indeed, Figs. 8, 9, 10 and 11 show that if the objective number does not exceed three, the behavior of the MOEA/D will be better than that of the NSGA-III. In the case of four objectives or more, the behavior of the NSGA-III becomes better than that of the MOEA/D. This statement is explained by the fact that, unlike MOEA/DD and Two_Arch2, the NSGA-III is only dedicated to many-objective problems.
 - Contrary to different studies such as Li et al. (2015) affirming that the decomposition paradigms are generally more efficient than the NSGA-III, our findings show that MOEA/DD is not always better than NSGA-III because our problem is a real-world complex one having some features which differ from those characterizing the theoretical problems used to evaluate these algorithms.
 - Finally, it is proven that the incorporation of the dimensionality reduction and the user preferences improves the results (higher HV, higher coverage and localization) and enhance the behavior of the tested algorithms.
 - The Two_Arch2 generally has a constant comportment which is not affected by varying the objectives number.

8 Conclusion

This study proposed to resolve the problem of indoor 3D redeployment of the connected objects in IoT collection networks by adding new objects on the chosen locations and guaranteeing a set of objectives. For this purpose, we developed a hybridization scheme that overcomes the problems of the computational complexity and the considerable time spent by the MaOAs to solve the MaOPs. This scheme integrated the dimensionality reduction and the user's preferences to various recent many-objective algorithms like MOEA/DD, NSGA-III and Two_Arch2. Subsequently, to prove their effectiveness in finding solutions for the 3D redeployment problem. The new proposed algorithms were evaluated using the HV metric on our real problem. Moreover, a set of experimental tests were performed to validate the theoretical observations. The results proved that the modified hybrid algorithms are more performing compared to the originals ones. In fact, other various interesting findings are obtained such as the superior performance of MOEA/D compared with NSGA-III if the objectives are correlated.

In the future, several research directions can be envisaged to improve this work. We intend to integrate other recent MOEAs such as KnEA (Zhang et al. 2015) into

our platform to test their behaviors on our real world problem. Moreover, although the proposed approach reduces the complexity and the objectives of the problem, it seems to be complex. Hence, the study of the algorithmic complexity of our hybrid approach can be considered to show the contribution of its use. At the application level, in order to prove the large-scale suitability of our method and to measure the effect of the density of network on the results, our experiments may be re-evaluated by testing the proposed hybridization scheme using a larger number of nodes. This is possible via real prototyping platforms such as IoTLab (IoTLab 2019) (having more than 1000 nodes) permitting to measure the same metrics as in our tests (RSSI, FER, number of neighbors). Thus, a consistent comparison with our experiments and an evaluation of the behavior of the proposed hybridization scheme in different real world contexts can be done. Besides, other than the 802.15.4 nodes used in our experiment tests, we aim to support other recent protocols and technologies of transmission (Ultra-Wide Band and LORA) and apply them on test our approach in a smart-home dedicated to aged persons (MIB of the IUT of Blagnac in Toulouse).

Funding None.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.


References

- Argany, M., Karimipour, F., Mafi, F., Afghantoloe, A.: Optimization of wireless sensor networks deployment based on probabilistic sensing models in a complex environment. *J. Sens. Actuator Netw.* 7(2), 20 (2018). <https://doi.org/10.3390/jsan7020020>
- Arduino platform: <https://www.arduino.cc/en/main/software> (2018). Accessed 5 Jan 2018
- Bechikh, S., Ben Said, L., Ghédira, K.: Searching for knee regions of the Pareto front using mobile reference points. *Soft Comput.* 15(9), 1807–1823 (2011). <https://doi.org/10.1007/s00500-011-0694-3>
- Branke, J., Deb, K., Miettinen, K., Slowinski, R.: *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer, Berlin (2008)
- Cheng, X., Du, D.Z., Wang, L., Xu, B.: Relay sensor placement in wireless sensor networks. *ACM/Springer J. Wirel. Netw.* 14(3), 347–355 (2008). <https://doi.org/10.1007/s11276-006-0724-8>
- Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference point-based non-dominated sorting approach, part I: solving problems with box constraints. *IEEE Trans. Evol. Comput.* 18(4), 577–601 (2014). <https://doi.org/10.1109/TEVC.2013.2281535>
- Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. *Evolutionary multiobjective optimization*. In: Abraham, A., Jain, L., Goldberg, R. (eds.) *Advanced Information and Knowledge Processing*, pp. 105–145. Springer, London (2005)
- Deb, K., Chaudhuri, S., Miettinen, K.: Towards estimating nadir objective vector using evolutionary approaches. In: *8th Genetic and Evolutionary Computation Conference (GECCO)*, pp. 643–650 (2006). <https://doi.org/10.1145/1143997.1144113>
- Domingo-Perez, F., Lazaro-Galilea, J.L., Bravo, I., Gardel, A., Rodriguez, D.: Optimization of the coverage and accuracy of an indoor positioning system with a variable number of sensors. *Sensors (Basel, Switzerland)* 16(6), 934 (2016). <https://doi.org/10.3390/s16060934>
- Drechsler, N., Sülflow, A., Drechsler, R.: Incorporating user preferences in many-objective optimization using relation e-preferred. *Nat. Comput.* 14, 469 (2015). <https://doi.org/10.1007/s11047-014-9422-0>
- Elhabyan, R., Shi, W., St-Hilaire, M.: Coverage protocols for wireless sensor networks: review and future directions. *J. Commun. Netw.* 21(1), 45–60 (2019). <https://doi.org/10.1109/JCN.2019.000005>

-
- Fonseca, C.M., Paquete, L., López-Ibáñez, M.: An improved dimension—sweep algorithm for the hyper-volume indicator. In: Congress on Evolutionary Computation, pp. 1157–1163. IEEE Press, Piscataway (2006). <https://doi.org/10.1109/CEC.2006.1688440>
- Gong, D., Wang, G., Sun, X.: Set-based genetic algorithms for solving many-objective optimization problems. In: 13th UK Workshop on Computational Intelligence (UKCI), Guildford, pp. 96–103 (2013). <https://doi.org/10.1109/UKCI.2013.6651293>
- Guo, J., Jafarkhani, H.: Movement-efficient sensor deployment in wireless sensor networks with limited communication range. *IEEE Trans. Wirel. Commun.* **18**(7), 3469–3484 (2019). <https://doi.org/10.1109/TWC.2019.2914199>
- Huang, B., Liu, W., Wang, T., Li, X., Song, H., Liu, A.: Deployment optimization of data centers in vehicular networks. *IEEE Access* **7**, 20644–20663 (2019a). <https://doi.org/10.1109/ACCESS.2019.2897615>
- Huang, X., Cheng, S., Cao, K., Cong, P., Wei, T., Hu, S.: A survey of deployment solutions and optimization strategies for hybrid SDN networks. *IEEE Commun. Surv. Tutor.* **21**(2), 1483–1507 (2019b). <https://doi.org/10.1109/COMST.2018.2871061>
- Ishibuchi, H., Akedo, N., Nojima, Y.: EMO algorithms on correlated many-objective problems with different correlation strength. *World Automation Congress 2012*, Puerto Vallarta, Mexico, pp. 1–6 (2012) IoTLab platform: <https://www.iot-lab.info> (2019). Accessed 22 June 2019
- Ko, A.H.R., Gagnon, F.: Process of 3D wireless decentralized sensor deployment using parsing crossover scheme. *Appl. Comput. Inform.* **11**(2), 89–101 (2015). <https://doi.org/10.1016/j.aci.2014.11.001>
- Li, K., Deb, K., Zhang, Q., Kwong, S.: An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Trans. Evol. Comput.* **19**(5), 694–716 (2015). <https://doi.org/10.1109/TEVC.2014.2373386>
- Liu, X., Qui, T., Zhou, X., Wang, T., Yang, L., Chang, V.: Latency-aware anchor-point deployment for disconnected sensor networks with mobile sinks. *IEEE Trans. Ind. Inf.* (2019). <https://doi.org/10.1109/9/TII.2019.2916300>
- Luo, X., Li, X., Wang, J., Guan, X.: Potential-game based optimally rigid topology control in wireless sensor networks. *IEEE Access* **6**, 16599–16609 (2018). <https://doi.org/10.1109/ACCESS.2018.2814079>
- Mitra, P., Murthy, C.A., Pal, S.K.: Unsupervised feature selection using feature similarity. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(3), 301–312 (2002). <https://doi.org/10.1109/34.990133>
- Mnasri, S., Nasri, N., Van Den Bossche, A., Val, T.: The 3D deployment multi-objective problem in mobile WSN: optimizing coverage and localization. *Int. Res. J. Innov. Eng. (IRJIE)* **1**(5), 1–14 (2015)
- Mnasri, S., Nasri, N., Van Den Bossche, A., Val, T.: A hybrid ant-genetic algorithm to solve a real deployment problem: a case study with experimental validation. In: Puliafito, A., Bruneo, D., Distefano, S., Longo, F. (eds.) *Ad hoc, Mobile, and Wireless Networks. ADHOC-NOW 2017*. Lecture Notes in Computer Science, vol. 10517. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67910-5_30
- Mnasri, S., Nasri, N., Van Den Bossche, A., Val, T.: 3D indoor redeployment in IoT collection networks: a real prototyping using a hybrid PI-NSGA-III-VF. In: *The 14th International Wireless Communications and Mobile Computing Conference IWCNC 2018*, pp. 780–785 (2018)
- Qu, B.Y., Suganthan, P.N., Liang, J.J.: Differential evolution with neighborhood mutation for multimodal optimization. *IEEE Trans. Evol. Comput.* **16**(5), 601–614 (2012). <https://doi.org/10.1109/TEVC.2011.2161873>
- Rostami, S.: Preference focussed many-objective evolutionary computation. Ph.D. dissertation (chapter 2), School of Engineering, Manchester Metropolitan University, Manchester, UK, M15 6HB (2014)
- Saul, L.K., Weinberger, K.Q., Ham, J.H., Sha, F., Lee, D.D.: Spectral methods for dimensionality reduction. In: Schoelkopf, O.C.B., Zien, A. (eds.) *Semisupervised Learning*. MIT Press, Cambridge (2006)
- Savkin, A.V., Huang, H.: A method for optimized deployment of a network of surveillance aerial drones. *IEEE Syst. J.* (2019). <https://doi.org/10.1109/jsyst.2019.2910080>
- Saxena, D.K., Duro, J.A., Tiwari, A., Deb, K., Zhang, Q.: Objective reduction in many-objective optimization: linear and nonlinear algorithms. *IEEE Trans. Evol. Comput.* **17**(1), 77–99 (2013). <https://doi.org/10.1109/TEVC.2012.2185847>
- Shlens, J.: A tutorial on principal component analysis. Center for Neural Science, New York University, Tech. Rep (2009)
- Sinha, A., Korhonen, P., Wallenius, J., Deb, K.: An improved progressively interactive evolutionary multi-objective optimization algorithm with a fixed budget of decision maker calls. *Eur. J. Oper. Res.* **233**(3), 674–688 (2014). <https://doi.org/10.1016/j.ejor.2013.08.046>

- Sinha, A., Saxena, D.K., Deb, K., Tiwari, A.: Using objective reduction and interactive procedure to handle many-objective optimization problems. *Appl. Soft Comput.* 13(1), 415–427 (2013). <https://doi.org/10.1016/j.asoc.2012.08.030>
- Tian, Y., Cheng, R., Zhang, X., Jin, Y.: PlatEMO: a MATLAB platform for evolutionary multi objective optimization. *IEEE Comput. Intell. Mag.* 12(4), 73–87 (2017). <https://doi.org/10.1109/MCI.2017.2742868>
- Tsang, Y.P., Choy, K.L., Wu, C.H., Ho, G.T.S.: Multi-objective mapping method for 3D environmental sensor network deployment. *IEEE Commun. Lett.* 23(7), 1231–1235 (2019). <https://doi.org/10.1109/LCOMM.2019.2914440>
- 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, Thessaloniki, Greece, pp. 1–6 (2016). <https://doi.org/10.1109/ICT.2016.7500490>
- Wang, H., Jiao, L., Yao, X.: Two_Arch2: an improved two-archive algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* 19(4), 524–541 (2015). <https://doi.org/10.1109/TEVC.2014.2350987>
- Weinberger, K.Q., Saul, L.K.: Unsupervised learning of image manifolds by semidefinite programming. *Int. J. Comput. Vis.* 70(1), 77–90 (2006). <https://doi.org/10.1109/CVPR.2004.1315272>
- Xu, H., Lai, Z., Liang, H.: A novel mathematical morphology based antenna deployment scheme for indoor wireless coverage. In: IEEE80th Vehicular Technology Conference (VTC Fall), pp. 1–5 (2014). <https://doi.org/10.1109/VTCFall.2014.6965828>
- Yuan, Y., Xu, H., Wang, B., Zhang, B., Yao, X.: Balancing convergence and diversity in decomposition-based many-objective optimizers. *IEEE Trans. Evol. Comput.* 20(2), 180–198 (2016). <https://doi.org/10.1109/TEVC.2015.2443001>
- Zhang, X., Tian, Y., Jin, Y.: A knee point driven evolutionary algorithm for many objective optimization. *IEEE Trans. Evol. Comput.* 19(6), 761–776 (2015). <https://doi.org/10.1109/TEVC.2014.2378512>
- Zhang, H., Liu, Y., Zhou, J.: Balanced-evolution genetic algorithm for combinatorial optimization problems: the general outline and implementation of balanced evolution strategy based on linear diversity index. *Nat. Comput.* (2018). <https://doi.org/10.1007/s11047-018-9670-5>

Affiliations

Sami Mnasri¹  · Nejah Nasri^{2,3} · Malek Alrashidi² · Adrien van den Bossche¹ · Thierry Val¹

✉ Sami Mnasri
Sami.Mnasri@fsgf.mu.tn

Nejah Nasri
nejah.nasri@isecs.mu.tn

Malek Alrashidi
mqalrashidi@ut.edu.sa

Adrien van den Bossche
vandenbo@irit.fr

Thierry Val
val@irit.fr

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

² Department of Computer Science, Community College, University of Tabuk, Tabuk, Saudi Arabia

ENIS, LETI, University of Sfax, Sfax, Tunisia