

# Assuming Multiobjective Metaheuristics to Solve a Three-Objective Optimisation Problem for Relay Node Deployment in Wireless Sensor Networks

Jose M. Lanza-Gutierrez\*, Juan A. Gomez-Pulido

Department of Technologies of Computers and Communications, University of Extremadura,  
Polytechnic School, Campus Universitario s/n, 10003 Caceres (Spain)

## Abstract

This paper deals with how to efficiently deploy energy-harvesting relay nodes in previously-established low-cost static Wireless Sensor Networks (WSNs), assuming a single-tiered network model. The purpose is to optimise three conflicting objectives: average energy cost, average sensitivity area, and network reliability. This is the so-called Relay Node Placement Problem (RNPP), which is an NP-hard optimisation problem. We find many works assuming heuristics in the current literature. However, it is not the case for metaheuristics, which usually provide good results solving such complex problems. This situation led us to consider a wide range of MultiObjective (MO) metaheuristics: the two standard genetic algorithms NSGA-II and SPEA2, the trajectory algorithm MO-VNS, the algorithm based on decomposition MOEA/D, and two novel swarm intelligence algorithms MO-ABC and MO-FA, which are based on the behaviour of honey bees and fireflies, respectively. These metaheuristics are applied to optimise a freely available data set. The results obtained are analysed considering two MO metrics: hypervolume and set coverage. Through a widely accepted statistical methodology, we conclude that MO-FA provides the best performance on average. We also study the efficiency of this approach, verifying that it is a good strategy to optimise such networks, including some limitations. Finally, we compare this proposal to another author approach, which assumes a heuristic.

**Keywords:** wireless sensor networks, optimization, energy efficiency, coverage, metaheuristics, reliability, swarm intelligence

## 1. Introduction

Wireless Sensor Networks (WSNs) are widely considered in many fields of application in the last years. For example, intensive agriculture, environmental control, industrial monitoring, forest fire detection, and home automation, among others [1].

Traditionally, a WSN is composed of a set of sensors capturing information about the environment and a sink node (also called collector node), which collects all the information provided by the network. The sensors have some important features, which encourage the use of this technology. For example, they are small, cheap, power-autonomous, and able to capture different types of information in a same device. In addition, the use of wireless technology facilitates the network deployment, reducing its cost. These features, among others, allow considering WSNs in environments where the deployment of wired technologies would be impossible or very expensive [2].

Nevertheless, WSNs have also important shortcomings affecting critical features, e.g. energy consumption and Quality of Service (QoS). Traditionally, the sensors are powered by batteries to assume cheap devices and to avoid the use of wires. All the information captured by the sensors is sent to the collector node, which involves an energy cost. Consequently, WSNs are particularly sensitive to energy consumption, affecting the network behaviour. If the network assumes a star topology, all the sensors follow a similar energy distribution. However, if we consider a habitual multi-hop protocol, the energy distribution could be unbalanced, involving the existence of bottlenecks, i.e. sensors subject to higher energy costs.

With the goal of avoiding the bottlenecks, a new device specialised in communication tasks is added to traditional WSNs [3]. This device is called Relay Node (RN) or router, which forwards all the information received to the sink node, so reducing the workload of the sensors. The routers have higher energy capacity than the sensors. They can be plugged into the grid, have greater batteries, or be energy harvesting devices (e.g. powered by solar energy). Hence, they are not limited to be placed close to a plug, being possible to deploy the network in difficult terrains, such as jungles, farmland, and war zones. Consequently, the routers are more expensive than the sensors and their deployment must be carefully studied.

The efficient deployment of WSNs depends on many factors. Hence, this issue was defined as an NP-hard optimisation problem [4] by several authors [5] [6]. This type of problem cannot

\*Corresponding author

Email addresses: jmlanza@unex.es (Jose M. Lanza-Gutierrez),  
jangomez@unex.es (Juan A. Gomez-Pulido)

This paper is an extended, improved version of the paper *A Trajectory-based Heuristic to Solve a Three-Objective Optimization Problem for Wireless Sensor Network Deployment* presented at EvoComNet2014 and published in: Applications of Evolutionary Computing, Proceedings of 17th European Conference, EvoApplications 2014, Granada, Spain, April 23-25, 2014, LNCS YYYY (to appear soon), Springer, 2014.

be solved through exact techniques, but non-conventional ones, such as Evolutionary Computation (EC) [7].

In this paper, we study how to deploy energy-harvesting RNs in previously-established low-cost static traditional WSNs, with the purpose of optimising three important factors. This is the so-called Relay Node Placement Problem (RNPP). The following contributions are presented in this work:

- A formal statement of the RNPP is presented, assuming three conflicting objectives to optimise: Average Energy Consumption (AEC) of the sensors, Average Sensitivity Area (ASA) provided by the network, and Network Reliability (NR), including a trade-off study between them.
- Six different MultiObjective (MO) metaheuristics are considered to solve the problem. Five of them belong to EC: NSGA-II [8], SPEA2 [9], MO-ABC [10], MO-FA [11], and MOEA/D [12]. NSGA-II and SPEA2 are two standard Genetic Algorithms (GAs). MO-ABC and MO-FA are two novel swarm intelligence algorithms, being based on the behaviour of honey bees and fireflies, respectively. MOEA/D is a generic algorithm framework, which decomposes an MO optimization problem into different single objective optimization subproblems. The remainder is a trajectory algorithm called MO-VNS [13].
- These metaheuristics are applied to solve a freely available data set proposed by ourselves in [14], since standard test suites or benchmark cases do not exist for the problem being studied. Instead, papers in the recent literature assume non-public or randomly generated data sets. Consequently, this contribution could be replicated and enhanced by other authors in future works.
- Another author approach [3] is assumed to solve the data set. We study the advantages which accrue from deploying RNs to traditional WSNs following both approaches.

The remainder of this paper is structured as follows. The next section discusses the literature review. The WSN model considered is exposed in Section 3. The optimisation problem is explained in Section 4. The metaheuristics assumed are discussed in Section 5. The experimental methodology is presented in Section 6. The computational results appear in Section 7. Section 8 compares our methodology to another author approach. Concluding remarks are left to Section 9. The acronyms considered in all the sections are listed in Table 1.

## 2. Literature review

We find two main lines of research for WSN deployment: authors studying how to efficiently deploy traditional WSNs, i.e. a set of sensors and a collector node, and works adding RNs to traditional WSNs to optimise them.

Taking the first approach, some authors considered heuristics. Cheng et al [5] minimised the energy cost by assigning different power transmission levels to the sensors. Chang and Tasoulas [6] maximised the network lifetime optimising the routing protocol. Cardei and Du [15] split the WSN into disjoint

sets of sensors, maximising the network lifetime by deciding which set had to be in the active mode. Zhang and Hou [16] optimised coverage and connectivity by keeping a minimum number of active sensors. Liu et al [17] maximised the network lifetime, maintaining connectivity and coverage, by following the approach proposed by Cardei and Du [15]. Mahboubi et al. [18] studied deployment strategies based on Voronoi polygons to increase coverage in wireless mobile sensor networks.

Other authors assumed metaheuristics to this end. Konstantinidis et al [19] maximised network lifetime and coverage by considering MOEA/D. Jia et al [20] optimised the coverage rate and the number of sensors deployed; assuming a GA to decide which sensors had to be active. Jia et al [21] considered NSGA-II, for the same purpose as Jia et al [20], for sensors with adjustable sensing radius. Hu et al [22] assumed a GA to maximise the network lifetime by splitting the WSNs as Cardei and Du [15]. Konstantinidis and Yang [23] improved the work done in [19] ensuring a certain connectivity. Le Berre et al [24] applied some MultiObjective Evolutionary Algorithms (MOEAs) to optimise lifetime, coverage, and financial cost. Martins et al [25] maximised the network lifetime assuming an MO hybrid algorithm. Yoon et al. [26] proposed a GA to deploy the sensor nodes optimising the network coverage. Mini et al. [27] considered a two-step algorithm to optimise the network lifetime. Firstly, they deployed the sensors at theoretical optimal locations such that the network lifetime was maximum. Secondly, they scheduled the WSN by a swarm intelligence algorithm optimising the network lifetime.

The deployment of traditional WSNs with many sensors has an important shortcoming. Being as the sensors are energy limited and the number of relayed packets is high, it is common to consider redundant sensors to enhance the performance of the network. This situation implies additional costs. The efficient addition of RNs to traditional WSNs is presented in the literature as a possible solution to this issue. Taking this line, we find two different approaches: Single-Tiered and Two-Tiered WSNs, ST-WSNs and TT-WSNs, respectively. In the first one, all the devices can communicate among them following a multi-hop routing protocol. The second approach is a cluster-based network, where the sensors send the data to the cluster head (an RN) in one hop, and then the RNs forward the information to the collector node in one or more hops.

Following the second line, some authors considered heuristics. Hou et al [3] maximised the network lifetime and mitigated the network geometric deficiencies in TT-WSNs. Tang et al [28] ensured connectivity and fault-tolerance deploying the minimum number of routers in TT-WSNs; to this end, they assumed two polynomial time approximation algorithms. Wang et al [29] minimised the network cost in TT-WSNs with constraints on lifetime and connectivity, assuming two models: the routers were not energy constrained and all the nodes were energy limited. Lloyd and Xue [30] optimised the network lifetime in ST-WSNs, while they preserved the network connectivity; to this end, they followed two different approaches: on the one hand, there was a connecting path composed of routers and/or sensors between each pair of sensors, and on the other hand, the path was composed solely of routers. Han et al [31]

Table 1: Acronyms.

AEC	Average Energy Consumption	MOEA/D	Multiobjective Evolutionary Algorithm Based on Decomposition
AFN	Aggregation and Forwarding Node	MSN	Micro Sensor node
ASA	Average Sensitivity Area	NBI	Normal Boundary Intersection
BS	Base Station	NR	Network Reliability
C-RNPP	Constrained Relay Node Placement Problem	NSGA-II	Non-dominated Sorting Genetic Algorithm-II
CHIM	Convex Hull of Individual Minima	QoS	Quality of Service
EC	Evolutionary Computation	RN	Relay Node
GA	Genetic Algorithm	RNPP	Relay Node Placement Problem
IQR	InterQuartile Range	SPEA2	Strength Pareto Evolutionary Algorithm 2
MAC	Medium Access Control	SPINDS	Smart Pairing and INtelligent Disc Search
MO	MultiObjective	ST-WSN	Single-Tiered Wireless Sensor Network
MO-ABC	MultiObjective Artificial Bee Colony	TT-WSN	Two-Tiered Wireless Sensor Network
MO-FA	MultiObjective Firefly Algorithm	WSN	Wireless Sensor Network
MO-VNS	MultiObjective Variable Neighborhood Search		
MOEA	MultiObjective Evolutionary Algorithm		

optimised the fault-tolerance in ST-WSNs considering sensors with adjustable transmission radius. Xu et al [32] studied the impacts of random device placement on connectivity and life-time in TT-WSNs. Misra et al [33] ensured connectivity by deploying a minimum number of RNs in ST-WSNs, where the routers were constrained to be placed at a subset of candidate locations, this is the so-called Constrained RNPP (C-RNPP). Following this constrained approach, Misra et al. [34] placed a minimum number of routers to ensure connectivity and survivability in energy-harvesting ST-WSNs, where the energy harvesting potential of the candidate locations was known a priori. Nigam et al. [35] proposed a branch-and-cut algorithm to deploy the minimum number of routers at a subset of candidate locations in ST-WSNs, such that the sensors were communicated to the sink node within a prespecified delay bound.

Some works assumed metaheuristics to this end. Zhao and Chen [36] applied a particle swarm algorithm to optimise the energy efficiency by minimising the average path length in ST-WSNs. Perez et al [37] assumed an MO algorithm, optimising both the energy cost and the number of routers in ST-WSNs, considering the C-RNPP. Peiravi et al. [38] proposed a method of clustering homogeneous TT-WSNs using an MO two-nested GA, with the aim of obtaining clustering schemes, where the network lifetime was optimised for different delay values.

This work differs from the papers described before in the following: i) We consider an unconstrained single-tiered approach of the RNPP. ST-WSNs are usually assumed in medium-size networks, where devices are low-cost, such as in intensive agriculture [39]. In this model, the RNs are similar to sensors, but they do not have circuits for capturing data, having additional energy capacity. In our approach, the RNs are energy-harvesting devices. Hence, they can be placed in anywhere of the surface. Consequently, the search space is greater than in the C-RNPP. ii) We propose to analyse the behaviour of several MO metaheuristics solving this problem. It is well known that MO metaheuristics provide a trade-off between the conflicting objective functions. Thus, the network designer will have several possibilities to deploy the WSN. This is not the case for heuristics, providing a unique solution. As described

above, many works assumed heuristics for ST-WSNs [3] [28] [29] [30] [31] [32] [33] [34] [35]. However, only a few papers studied the single-tiered approach by assuming metaheuristics [36] [37]. Regarding these works, the authors in [36] considered a singleobjective problem, but ours is MO. Hence, this is a more realistic approach. In [37], they assumed an MO C-RNPP; however, our approach is unconstrained.

This work is partially inspired by some early papers. In [40] and [41], we considered NSGA-II and SPEA2 to solve a similar problem for two objectives. In [42] and [43], we improved [40] by including MO-VNS and MO-ABC, respectively. An early version of this three-objective problem was presented in [44], considering NSGA-II and SPEA2. Finally, we solved the same problem with MO-VNS in [45]. Now, we present a more complete work, including two novel swarm intelligence algorithms, a state-of-the-art MOEA based on decomposition, a bigger data set, and additional comparative studies.

### 3. The wireless sensor network model assumed

In this section, and according to the notation shown in Table 2, we describe the WSN model assumed in the RNPP. Firstly, we present the general assumptions of the model. Next, we discuss some important aspects: the energy cost, the sensitivity area provided by the WSN, and the network lifetime concept.

#### 3.1. Assumptions of the wireless sensor network model

The general assumptions of the model are:

1. The network is composed of three types of wireless static devices: a collector node,  $\tilde{s}_s$  sensors, and  $\tilde{s}_r$  RNs. All of them are placed on a same 2D-surface of size  $d_x \times d_y$ .
2. The sensors are powered by batteries. The collector and the RNs have enough energy capacity for operating over the network lifetime. They are energy-harvesting devices.
3. Initially, at time  $t = 0$ , all the sensors start with the same energy capacity  $iec$  in their batteries. If during operation,  $t > 0$ , a sensor is exhausted, it cannot be linked again.

Table 2: Notation for modelling the WSN and the optimisation problem.

$\alpha$	path loss exponent, $\alpha \in [2, 4]$ .	$h_l^{ic}$	number of hops in the $l$ -th disjoint path between $i \in \tilde{S}_s$ and the sink node.
$\beta$	transmission quality parameter, $\beta > 0$ .	$iec$	initial energy charge of the sensors, $iec > 0$ .
$\tau$	set of time periods, $\tau = \{0, 1, 2, \dots\}$ .	$k$	information packet size in bits, $k > 0$ .
$A(t)$	sensitivity area provided by the WSN at time $t > 0 \in \tau$ .	$t_n$	network lifetime of the WSN based on coverage threshold.
$a_p(t)$	variable assuming 1 if there is at least one sensor $i \in S_s(t)$ at a distance to the demand point $p \in \tilde{D}_p(t)$ lower than $r_s$ .	$P_i(t)$	number of packets sent by the sensor $i \in S_s(t)$ at time $t > 0$ .
$amp$	energy cost per bit of the power amplifier, $amp > 0$ .	$re_i$	reliability of the sensor $i \in \tilde{S}_s$ .
$c$	collector coordinates, $c = (x, y)$ where $x \in [0, d_x]$ and $y \in [0, d_y]$ .	$r_c$	communication radius, $r_c > 0$ .
$co_{th}$	coverage threshold, $co_{th} \in [0, 1]$ .	$r_s$	sensitivity radius, $r_s > 0$ .
$\tilde{D}_p(t)$	set of demand points at time $t > 0$ , $\forall p \in \tilde{D}_p$ , $p = (x, y)$ where $x \in [0, d_x]$ and $y \in [0, d_y]$ .	$\tilde{S}_r$	set of router coordinates, $\forall r \in \tilde{S}_r$ , $r = (x, y)$ where $x \in [0, d_x]$ and $y \in [0, d_y]$ .
$\tilde{d}_p(t)$	number of demand points. It is the cardinal of $\tilde{D}_p(t)$ .	$\tilde{s}_r$	number of routers. It is the cardinal of $\tilde{S}_r$ .
$d_j p_i^c$	number of disjoint paths between the sensor $i \in \tilde{S}_s$ and the collector node.	$\tilde{S}_s$	set of initial sensor coordinates, $\forall i \in \tilde{S}_s$ , $i = (x, y)$ , where $x \in [0, d_x]$ and $y \in [0, d_y]$ .
$d_x$	width of the surface, $d_x > 0$ .	$\tilde{s}_s$	number of initial sensors. It is the cardinal of $\tilde{S}_s$ .
$d_y$	height of the surface, $d_y > 0$ .	$S_s(t)$	set of sensor coordinates with an energy charge greater than 0 at time $t > 0$ , $S_s(t) \subseteq \tilde{S}_s$ .
$Ec_i(t)$	energy charge of a sensor $i \in S_s(t)$ at time $t$ .	$s_s(t)$	number of sensors with an energy charge greater than 0 at time $t > 0$ . It is the cardinal of $S_s(t)$ , $s_s(t) \leq \tilde{s}_s$ .
$Ee_i(t)$	energy expenditure of a sensor $i \in S_s(t)$ at time $t > 0$ .	$w_i^c(t)$	variable which provides the next device in the minimum path between $i \in S_s(t)$ and the collector node at $t > 0$ , $w_i^c(t) \in \{S_s(t) \cup \tilde{S}_r\} + c - i$ .
$err$	local channel error, $err \in [0, 1]$ .	$z_{ji}^c(t)$	variable assuming 1 if $i \in S_s(t)$ is in the minimum path between $j \in S_s(t)$ and the collector node at $t > 0$ .
$f_1$	AEC of the sensors over the network lifetime.		
$f_2$	ASA provided by the WSN over the network lifetime.		
$f_3$	NR based on disjoint paths.		

4. The collector node is the only connection point of the network to the outside.
5. The sensors capture information about the environment on a regular basis and with a sensitivity radius  $r_s$ , i.e. each sensor covers a circumference of radius  $r_s$ . The information is immediately sent to the collector node.
6. The RNs only forward all the information received to the collector node. They do not capture information.
7. Any two devices are linked, if they are at a distance lower than the communication radius  $r_c$ .
8. All the devices consider the same multi-hop routing protocol provided by Dijkstra's Algorithm [46], for minimum path length among devices.
9. We suppose a perfect synchronisation among devices and a perfect medium access, as defined by S-MAC [47].

### 3.2. Energy expenditure

As stated before, the sensors are the only devices powered by batteries. Hence, we have to simulate the energy consumption of the sensors during operation. To this end, we consider the energy model proposed by Konstantinidis et al [19]. Thus, the energy expenditure is only due to the most expensive task: the sending. Consequently, processing, receiving, and sensing tasks are considered negligible.

Following this model, considering a time  $t$ , for  $t > 0$  and  $t \in \tau$ , a sensor  $i$ , with  $i \in S_s(t)$ , sends a number of information packets  $P_i(t)$  which is expressed as

$$P_i(t) = 1 + \sum_{j \in [S_s(t) - i]} z_{ji}^c(t), \quad (1)$$

where this value is given by the number of packets that the sensor relays and the number of packets that the sensor generates. In that case, each sensor generates a packet per time period.

Thus, the energy expenditure suffered by a sensor  $i$  at time  $t > 0$  is denoted by

$$Ee_i(t) = P_i(t) \cdot \beta \cdot amp \cdot k \cdot \|i - w_i^c(t)\|^\alpha, \quad (2)$$

where  $\|\cdot\|$  is the Euclidean distance between any two devices. This way, the energy charge of a sensor  $i$  at time  $t$  is given by

$$Ec_i(t) = \begin{cases} Ec_i(t-1) - Ee_i(t) & \text{if } t > 0 \\ iec & \text{if } t = 0 \end{cases}, \quad (3)$$

where if this value equals zero, the sensor is out of energy. Otherwise, we consider that the sensor is active.

### 3.3. Sensitivity area provided by the network

We find two main ways to calculate the sensitivity area provided by a WSN in the literature [48]. On the one hand, some authors assume that each sensor covers a circumference of radius  $r_s$  and area  $\pi \cdot r_s^2$ . Thus, the global coverage is the union of the  $s_s(t)$  areas. On the other hand, some works consider a set of demand points uniformly distributed on the surface. Then, they count the number of demand points having an active sensor at a distance lower than  $r_s$ .

Although the first method is a little bit more accurate, we opted for the second approach, because it is less hard to compute. Thus, we express the sensitivity area provided by a WSN at time  $t > 0$  as

$$A(t) = \frac{\sum_{p \in \tilde{D}_p(t)} a_p(t)}{\tilde{d}_p(t)}, \quad (4)$$

where  $a_p(t)$  is defined as

$$a_p(t) = \begin{cases} 1 & \text{if } \exists i \in \tilde{S}_s(t) / \|p - i\| < r_s \\ 0 & \text{otherwise} \end{cases}, \quad (5)$$

that means,  $a_p(t)$  equals 1 if there is a sensor  $i$  such that the distance between the demand point  $p$  and  $i$  is lower than  $r_s$ , and 0 otherwise.

### 3.4. Network lifetime and coverage

The network lifetime is defined as the number of time periods over which a WSN is able to provide useful information. There are several approaches in the literature to check this requirement, e.g. the time when all the sensors are exhausted. In this work, we assume a coverage threshold ( $co_{th}$ ) for this purpose. Thus, if the sensitivity area provided by the network is lower than  $co_{th}$ , we consider that the amount of information provided is not enough. That is expressed as

$$t_n = |\{t > 0 \in \tau / A(t) > co_{th}\}|, \quad (6)$$

where  $|\cdot|$  is the cardinal of the set.

### 4. Definition of the three-objective optimisation problem

Let  $f_1$  be the AEC of the sensors over the network lifetime. That is expressed as

$$f_1 = \frac{\sum_{t=1}^{t_n} \left( \sum_{i \in S_s(t)} \frac{Ee_i(t)}{s_s(t)} \right)}{t_n}, \quad (7)$$

where  $f_1 \in \mathbb{R}^+$  and both  $Ee_i(t)$  and  $t_n$  are given by (2) and (6), respectively. Let  $f_2$  be the ASA provided by the network, which is expressed as

$$f_2 = \frac{\sum_{t=1}^{t_n} A(t)}{t_n}, \quad (8)$$

where  $f_2 \in [0, 1]$  and  $A(t)$  is given by (4). And let  $f_3$  be the NR, showing the probability that the sensors successfully send information to the sink node. That is

$$f_3 = \sum_{i \in \tilde{S}_s} \left( \frac{re_i}{\tilde{s}_s} \right), \quad (9)$$

where  $f_3 \in [0, 1]$  and  $re_i$  is the reliability of the sensor  $i$ , defined in [49] as

$$re_i = 1 - \prod_{l=1}^{djp_i^c} \left( 1 - (1 - err)^{h_l^{ic}} \right), \quad (10)$$

where  $djp_i^c$  is the number of disjoint paths between  $i$  and the sink node,  $h_l^{ic}$  is the number of hops in the  $l$ -th disjoint path between both devices, and  $err$  is the local channel error. The disjoint paths are calculated through Suurballe's Algorithm [50].

This way, we define the RNPP as an MO optimisation problem, where given a previously-established traditional WSN, i.e.

Table 3: Trade-off study among the fitness functions.

case	$f_1$	$f_2$	$f_3$
#1	Increase	Decrease	<b>Increase</b>
#2	Increase	<b>Increase</b>	<b>Increase</b>
#3	<b>Decrease</b>	Decrease	Decrease
#4	<b>Decrease</b>	<b>Increase</b>	Decrease

$\tilde{s}_s$  sensors and a collector node, the objective is to place  $\tilde{s}_r$  routers by assuming a single tiered network model to

$$\min(f_1), \max(f_2), \text{ and } \max(f_3), \quad (11)$$

subject to

$$\forall r \in \tilde{S}_r, r = (x, y) / x \in [0, d_x] \text{ and } y \in [0, d_y]. \quad (12)$$

These objectives are related to three important problems in the WSN literature: i) Energy efficiency problem [51], whose aim is to reduce the energy cost, increasing the network lifetime and balancing the energy distribution. ii) Coverage problem [52], its purpose is to optimise the amount and diversity of the information provided by the network. ii) Reliability problem [53], the objective is to deploy a reliable WSN, this means that the network is able to recover from the failure of nodes, which is crucial to reduce maintenance costs. Although these problems are widely considered in the literature, these objectives were not considered all together for the deployment of ST-WSNs, applied to the unconstrained RNPP.

It is well-known that one fundamental requirement for a problem to be considered as an MO optimisation problem is that the objectives should be conflicting [7]. Other authors assumed that both energy cost and coverage were conflicting objectives in WSNs [19] [23] [24] [54] [55], where [23] and [54] also tackled the connectivity issue. With the purpose of showing that our definition of the RNPP is an MO problem, we analyse the trade-off among the fitness functions in Table 3, where *Increase* implies that the value of the fitness function is increased and *Decrease* otherwise. Note that *Increase* and *Decrease* are in boldface if the behaviour of the fitness function fits with the purpose of the optimization problem, i.e. it appears in boldface if the fitness function is increased for a maximisation problem or decreased for a minimisation one, respectively.

According to this trade-off study, we find four different cases in which at least one of the objectives is optimised:

- Case #1: on the basis of Eq. (9), a high reliability implies the existence of many disjoint paths between the sink node and the sensors, which allows that the network is able to recover from the collapse of nodes, avoiding that sensors having energy in their batteries are disconnected from the WSN. This means that more sensors can send packets over the network lifetime. As shown in Eq. (7), the energy consumption depends on the number of packets sent by the sensors, including forwards. Hence, more sensors implies more packets, increasing the energy cost. The average coverage is low due to the coverage is decreasing slowly over the network lifetime. As detailed in Eq. (8), this value depends on the number of sensors that can be

connected to the sink node over the lifetime. A bad distribution of the energy cost (which is high) could mean that the network loses coverage. However, as the reliability is high, the other sensors can reach the sink node and the coverage decreases slowly, instead of reaching the  $co_{th}$ .

- Case #2: it is similar to the previous case. The difference is that the deployment of the RNs is more efficient in energy terms, i.e. the energy distribution is homogeneous. Consequently, the coverage is not decreasing slowly, instead the sensors are exhausted at the same time, reaching the  $co_{th}$ .
- Case #3: a network with a low reliability implies that sensors having energy can be disconnected from the WSN, because other sensors are exhausted. This situation decreased the number of packets sent over the network lifetime, decreasing the energy cost. A bad distribution of this energy cost means that the network loses sensors slowly, decreasing the average coverage, as in case #1.
- Case #4: it is similar to #3. As in case #2, a better deployment of the RNs allows that the energy distribution is homogeneous. Hence, the sensors are exhausted at the same time, increasing the average coverage.

Analysing the behaviour of the fitness functions, we reach the following conclusions: we cannot optimise an objective without degrading at least one of the other two, and on the other hand, enhancing an objective does not necessarily imply that another is optimised. Thus, we conclude that the objectives are in conflict. Hence, this is an MO optimisation problem.

## 5. Multiobjective metaheuristics

As stated before, we consider six different MO metaheuristics to solve the RNPP. The operation details of some of these algorithms were exposed in prior works: NSGA-II and SPEA2 in [44], and MO-VNS in [45]. Thus, in this section, we discuss the state-of-the-art algorithm based on decomposition MOEA/D and the two novel swarm intelligence algorithms MO-ABC and MO-FA. Note that we consider the same encoding for all the algorithms. A chromosome is a possible solution to the optimisation problem, which is composed of  $\tilde{s}_r$  genes. As shown Fig. 1, a gene is the 2D-coordinate of a router.

### 5.1. Multiobjective artificial bee colony algorithm

The Artificial Bee Colony (ABC) proposed by Karaboga and Basturk [10] is based on the behaviour of honey bees. According to the authors, there are three essential components in a honey bee swarm: i) *Food sources* are set of flowers in the surrounding of the hive. ii) *Employed foragers* are the bees, which exploit known food sources. Each time one of these bees exploits a food source, it returns to the nest to upload the nectar and shares quality information about the source. iii) *Unemployed foragers* are the bees looking out for new food sources. There are two types: *onlookers* and *scouts*. The onlookers wait

### Chromosome definition

$r_1$	$r_2$	$r_{\tilde{s}_r}$	$r_z \in \tilde{S}_r, \quad r_z = (x_z, y_z) \text{ for } z = 1, \dots, \tilde{s}_r$
$(x_1, y_1)$	$(x_2, y_2)$	$\dots$	$(x_{\tilde{s}_r}, y_{\tilde{s}_r})$

where  $x_z \in [0, d_x]$  and  $y_z \in [0, d_y]$

### Chromosome example

Assuming  $d_x=200$ ,  $d_y=200$ , and  $\tilde{s}_r=4$ , a possible solution to the optimisation problem could be

$r_1$	$r_2$	$r_3$	$r_4$
(20.65, 80.40)	(180.31, 5.84)	(120.15, 7.10)	(2.45, 190.40)

Figure 1: Definition of the chromosome structure.

in the nest to define new food sources based on the information provided by the employed foragers. The scouts search new food sources without considering any experience.

In this algorithm, a food source is a solution to the optimisation problem. The colony consists of two groups of artificial bees: employers and onlookers. Each employer is assigned a different food source. The bee exploits its food source trying to get better solutions in its surrounding. The algorithm includes a parameter called *limit* to detect if a food source is exhausted. Thus, after *limit* times without getting a better solution, the employed considers the source exhausted. Then, the bee becomes a scout, choosing a new food source through a low cost search.

As given in Algorithm 1, we assume a colony  $P_t$  of size  $PS$ , being the number of employed and onlookers constant over generations,  $EMP$  and  $PS - EMP$ , respectively. Initially, each bee of the colony is assigned a random solution of the optimisation problem (line 1). At each iteration and so long as the stop criterion is not reached (line 2), the colony is split into two groups:  $SE_t$  and  $SO_t$ , i.e. the set of employed foragers and onlookers, respectively (line 3). Next, each  $emplB_a \in SE_t$  generates a new solution in its surrounding, assuming a randomly selected individual  $emplB_b \in SE_t$  (lines 4-5), that is

$$v_i = emplB_{ai} + S_F(emplB_{ai} - emplB_{bi}) \quad \forall i \in [1, \dots, \tilde{s}_r], \quad (13)$$

where  $S_F$  is a random value in the interval  $[-1, 1]$ , being  $emplB_{ai}$  and  $emplB_{bi}$  the routers in the  $i$ -th gene of  $emplB_a$  and  $emplB_b$ , respectively. A greedy selection between the new solution and the previous one is performed, increasing the attempt counter if the new solution is dominated (lines 6-9).

The onlookers assume a probability based selection process. As the original ABC is a singleobjective algorithm and the RNPP is an MO problem, we consider same assumptions to establish how good is a food source. Now,  $SE_t$  is sorted based on the elitist crowded-comparison operator  $<_n$  defined in [8] for NSGA-II. This way, the best solution of  $SE_t$  is twice as likely as the second one, the second solution is twice as likely as the third one, and so on. Accordingly, each onlooker bee  $onloB_a \in SO_t$  generates a new solution based on an employed forager  $emplB_a$ , which is randomly selected according to these probabilities (lines 11-14), that is

$$v_i = onloB_{ai} + S_F(onloB_{ai} - emplB_{ai}) \quad \forall i \in [1, \dots, \tilde{s}_r], \quad (14)$$

### Algorithm 1 MO-ABC

```

1:  $P_t \leftarrow \text{initialisePopulation}(PS)$ 
2: while not stop condition do
3:    $SE_t, SO_t \leftarrow \text{selectGroupBees}(P_t, EMP)$ 
4:   for each  $emplB_a \in SE_t$  do ▷ Employed process
5:      $v \leftarrow \text{generateEmployedSolution}(emplB_a)$ 
6:     if  $emplB_a$  dominates  $v$  then
7:        $emplB_a.\text{attemptCounter} + +$ 
8:     end if
9:      $emplB_a \leftarrow \text{greedySelection}(v, emplB_a)$ 
10:  end for
11:   $Probs_{E_t} \leftarrow \text{generateProbabilities}(SE_t)$  ▷ Onlooker process
12:  for each  $onloB_a \in SO_t$  do
13:     $emplB_a \leftarrow \text{selectEmployedBee}(Probs_{E_t}, SE_t)$ 
14:     $v \leftarrow \text{generateOnlookerSolution}(emplB_a, onloB_a)$ 
15:     $onloB_a \leftarrow \text{greedySelection}(v, onloB_a)$ 
16:  end for
17:  for each  $emplB \in SE_t$  do ▷ Scout process
18:    if  $emplB.\text{attemptCounter} > \text{limit}$  then
19:       $emplB_a \leftarrow \text{selectGoodEmployedBee}(SE_t)$ 
20:       $emplB_a \leftarrow \text{generateScoutSolution}(emplB_a)$ 
21:    end if
22:  end for
23:   $P_{t+1} \leftarrow SE_t \cup SO_t$ 
24:   $\text{saveBestSolutionsToFile}(P_{t+1})$  and  $t \leftarrow t + 1$ 
25: end while

```

where  $onloB_{a_i}$  and  $emplB_{a_i}$  are the routers in the  $i$ -th gene of  $onloB_a$  and  $emplB_a$ , respectively. The onlooker bee takes the value of the greedy selection between both solutions (line 15).

If the food source of an employed bee is exhausted, a solution is generated through the scout process. Thus, a solution  $emplB_a \in SE_t$  is randomly selected from the first two Pareto fronts of the set. Next, we get the Euclidean distance between  $emplB_a$  and all the solutions in  $SE_t \cup SO_t - \{emplB_a\}$ . Finally, the  $k$ -nearest solutions are combined to generate a new solution, which replaces the exhausted one (lines 17-22), that is

$$emplB_{a_i} = \frac{\sum_{bee \in NS} bee_i}{k} \quad \forall i \in [1, \dots, \tilde{s}_r], \quad (15)$$

where  $k$  is a random value between  $[2, 11]$ ,  $NS$  is the set of  $k$ -nearest solutions, and  $emplB_{a_i}$  and  $bee_i$  are the routers placed in the  $i$ -th gene of  $emplB_a$  and  $bee$ , respectively. Finally, the attempt counter of the food source is reset.

### 5.2. Multiobjective firefly algorithm

The Firefly Algorithm (FA) designed by Yang [11] is inspired by the idealised behaviour of fireflies. These nocturnal beetles generate flashing lights to attract mating partners and potential prey, being these signals only visible to a limited distance.

FA considers three basic rules: i) all fireflies are unisex, i.e. a firefly can attract another regardless of their sex. ii) attractiveness is proportional to light intensity. Thus, given two fireflies, the less bright one will move towards the brighter one. iii) the light intensity of a firefly depends on its qualities.

The authors define two important issues in the algorithm: the formulation of the attractiveness and the variation of brightness. A firefly is a solution to the optimisation problem. This way, the brightness of a firefly depends on the quality of its fitness functions. As FA is a singleobjective algorithm and the RNPP is an MO problem, we consider the  $<_n$  operator as for MO-ABC. Accordingly, given two fireflies  $f_a$  and  $f_b$ ,  $f_a$  is brighter

### Algorithm 2 MO-FA

```

1:  $P_t \leftarrow \text{initialisePopulation}(PS)$  and  $Q_t \leftarrow \{\}$ 
2:  $I_t \leftarrow \text{calculateLightIntensity}(P_t)$ 
3: while not stop condition do
4:   for  $a = 1 \rightarrow PS$  do
5:      $Q_t[a] \leftarrow P_t[a]$  and  $I_{Q_t[a]} \leftarrow I_{P_t[a]}$  ▷ Copy  $P_t$  to  $Q_t$ 
6:     for  $b = 1 \rightarrow PS$  do
7:       if  $I_{P_t[b]} > I_{Q_t[a]}$  then ▷ Attractiveness between fireflies
8:          $Q_t[a] \leftarrow \text{moveFirefly}(Q_t[a], P_t[b])$ 
9:       end if
10:    end for
11:  end for
12:   $Q_t \leftarrow \text{MoveBrightestFirefly}(Q_t)$ 
13:   $Q_t \leftarrow \text{calculateFitnessValues}(Q_t)$ 
14:   $P_{t+1} \leftarrow Q_t \cup P_t$ 
15:  if stagnation condition then ▷ Check stagnation
16:    for  $a = 1 \rightarrow PS$  do
17:      if  $P_{t+1}[a] > \text{mutation}(P_{t+1}[a])$  then
18:         $P_{t+1}[a] \leftarrow \text{randomOffset}(P_{t+1}[a])$ 
19:      end if
20:    end for
21:  end if
22:   $I_{t+1} \leftarrow \text{calculateLightIntensity}(P_{t+1})$ 
23: end while

```

than  $f_b$  if  $f_a <_n f_b$ . In such a case,  $f_b$  is attracted to  $f_a$ , i.e. the routers of  $f_b$  move towards the routers of  $f_a$ . Thus, the router in the  $i$ -th gene of  $f_a$  attracts the router in the same gene position of  $f_b$ .

Let  $d_{ab}$  be the Euclidean distance between  $f_a$  and  $f_b$  in the solution space, that is given by

$$d_{ab} = \sqrt{\sum_{i=1}^{\tilde{s}_r} (f_{a_i,x} - f_{b_i,x})^2 + (f_{a_i,y} - f_{b_i,y})^2}, \quad (16)$$

where  $f_{a_i}$  is the router in the  $i$ -th gene of  $f_a$ , being  $f_{a_i,x}$  and  $f_{a_i,y}$  the coordinates  $x$  and  $y$  of  $f_{a_i}$ , respectively. Definitions of  $f_{b_i}$ ,  $f_{b_i,x}$ , and  $f_{b_i,y}$  are similar to the previously discussed.

Let  $bd_{ab}$  be the bounded Euclidean distance between  $f_a$  and  $f_b$ , which is denoted by

$$bd_{ab} = \frac{d_{ab}}{\sqrt{\tilde{s}_r(d_x^2 + d_y^2)}}. \quad (17)$$

Then,  $f_a$  attracts  $f_b$  causing a movement in  $f_b$  expressed as

$$f_{b_i} = f_{b_i} + \beta_0 e^{\gamma bd_{ab}^2} (f_{a_i} - f_{b_i}) + \alpha_{FA}(\sigma - 1/2) \quad \forall i \in [1, \dots, \tilde{s}_r], \quad (18)$$

where  $\beta_0$  is the attractiveness at  $d_{ab} = 0$ ,  $\gamma$  is the light absorption coefficient,  $\alpha_{FA}$  is a randomisation parameter, and  $\sigma$  is a random value in the interval  $[0, 1]$ . In this function, we consider the bounded Euclidean distance (17) instead of the Euclidean distance (16) proposed in the algorithm. The aim is to delimit the range of values assumed by the exponential function. Now, distances are changed to the interval  $[0, 1]$ .

As outlined in Algorithm 2, we consider two populations  $P_t$  and  $Q_t$  of the same size  $PS$ .  $P_t$  saves the parents of the generation  $t$  and  $Q_t$  contains the offspring generated through  $P_t$ . Initially,  $P_t$  is set randomly, being  $Q_t$  empty (line 1). So long as the stop criterion is not reached (line 3),  $P_t$  is copied to  $Q_t$  (line 5). Next, each individual in  $Q_t$  is attracted to brighter individuals in  $P_t$ , following the expression (18) (lines 6-10). Afterwards, the brightest firefly in  $Q_t$  is moved randomly, because it was not

attracted to any individual (line 12). Then, we calculate the fitness functions of the new solutions in  $Q_t$  (line 13). Next, a new parent population  $P_{t+1}$  is generated, considering the best  $PS$  individuals of  $Q_t \cup P_t$  assuming  $<_n$  (line 14). Finally, we include a stagnation control based on the percentage of solutions in  $Q_t$ , dominating individuals from  $P_t$ . If this value is lower than the threshold called  $mof_{th}$ ,  $P_{t+1}$  is modified to increase the population diversity. This way, each individual in  $P_{t+1}$  is mutated considering the mutation operator discussed for NSGA-II and SPEA2 in [44]. If the new individual is dominated by the previous one, an offset is performed (lines 15-21).

### 5.3. Multiobjective evolutionary algorithm based on decomposition with NBI-Tchebycheff approach

The MultiObjective Evolutionary Algorithm based on Decomposition (MOEA/D) was proposed by Qingfu Zhang and Hui Liu [12]. In contrast to the majority of the current state-of-the-art MOEAs, which treat an MO optimisation problem as a whole. MOEA/D decomposes an MO optimisation problem into several scalar optimisation subproblems, optimising each one by only considering information from neighbouring subproblems, reducing its computational complexity. This way, MOEA/D provides a general framework allowing the application of any singleobjective technique to solve each subproblem.

Two decomposition methods are commonly assumed in MOEA/D: Weighted Sum Approach and Weighted Tchebycheff Approach [12]. The first one works out well on convex problems, but this is not the case for non-convex ones. The second method is able to deal with both types of problems. However, both approaches have a same shortcoming, they are very sensitive to the scale of the objectives. Note that, in the RNPP, the scale of  $f_1$  is different to  $f_2$  and  $f_3$ , being similar  $f_2$  and  $f_3$ .

A decomposition method independent of the scales of the objectives is Normal Boundary Intersection (NBI) [56]. This approach tries to find the intersection points between the Pareto front and a number of straight lines, which are defined by a set of uniformly-distributed points in the Convex Hull of Individual Minima (CHIM) and a normal vector. Because this method has several constraints applied to MOEA/D [57], the authors in [58] proposed to take the advantages of the NBI approach and the Tchebycheff approach, defining the NBI-Tchebycheff approach for decomposing three-objective optimisation problems. This decomposition method is assumed to solve the RNPP.

Thus, given a set of  $PS$  reference points  $R = \{r^1, r^2, \dots, r^{PS}\}$  evenly distributed on the plane II and the three approximate extreme points of the Pareto front  $F^1 = (F_1^1, F_2^1, F_3^1)$ ,  $F^2 = (F_1^2, F_2^2, F_3^2)$ , and  $F^3 = (F_1^3, F_2^3, F_3^3)$ , which are contained in the set of reference points. Then, the three-objective optimisation problem is decomposed into  $PS$  singleobjective minimisation subproblems according to the NBI-Tchebycheff approach, where the  $i$ -th optimisation function is defined as

$$g(x|r^i, \hat{n}) = \max \left\{ \begin{array}{l} n_1(f_1(x) - r_1^i), \\ n_2(r_2^i - f_2(x)), \\ n_3(r_3^i - f_3(x)) \end{array} \right\}, \quad (19)$$

where  $x$  is a solution to the optimisation problem,  $r^i \in R$ , and  $\hat{n} = (n_1, n_2, n_3)$  is the normal vector to the plane II.

### Algorithm 3 MOEA/D with NBI-Tchebycheff approach

```

1: File  $\leftarrow \{\}$  ▷ Initialisation
2:  $R \leftarrow generateReferenceSet(F^1, F^2, F^3, CHIM_{inc}, crowDistance)$ 
3:  $P_t \leftarrow initialisePopulation(PS)$ 
4:  $ND \leftarrow computeNeighbourhood(R)$ 
5: while not stop condition do
6:   for  $i = 1 \rightarrow PS$  do ▷ Update
7:      $k \leftarrow randInt(1, nSize)$ 
8:      $l \leftarrow randInt(1, nSize)$ 
9:      $new_{ind} \leftarrow crossover(P_t[ND_{i,k}], P_t[ND_{i,l}])$ 
10:     $new_{ind} \leftarrow mutation(new_{ind})$ 
11:    for  $j = 1 \rightarrow nSize$  do
12:      if  $g(new_{ind}|r^{ND_{i,j}}, \hat{n}) \leq g(P_t[ND_{i,j}]|r^{ND_{i,j}}, \hat{n})$  then
13:         $P_t[ND_{i,j}] \leftarrow new_{ind}$ 
14:      end if
15:    end for
16:    File  $\leftarrow updateNonDominatedSet(File, new_{ind})$ 
17:  end for
18:   $P_{t+1} \leftarrow P_t$ 
19: end while

```

As is given in Algorithm 3, MOEA/D assumes a *File* set saving all the non-dominated solutions found over generations. This set is initialised as empty (line 1). The set of reference points  $R$  is generated in line 2 as described in [58]. To this end, the algorithm considers two parameters: *crowDistance* and *CHIM<sub>inc</sub>*. The first one shows the distance between two consecutive reference points. The another is the percentage in which the area defined by  $F^1$ ,  $F^2$ , and  $F^3$  is increased regarding the original size, being this value greater or equal than 1.0. A population  $P_t$  of  $PS$  random individuals is generated in line 3, where  $PS$  is the cardinal of the set  $R$ . Thus, each reference point has associated a unique solution in  $P$ ,  $P_t[i]$  is related to  $r_i$ ,  $i \in 1, 2, \dots, PS$ . Next, the neighbourhood is computed in line 4. To this end, the Euclidean distance between any two points in  $R$  is calculated. Then, we obtain the  $nSize$  closest reference points to each reference point, which is another parameter of the algorithm. Thus,  $ND_{i,j} \in R$ , where  $i \in 1, 2, \dots, PS$  and  $j \in 1, 2, \dots, nSize$ , i.e. the reference point  $ND_{i,j}$  is the  $j$ -th closest reference point to  $i$ .

Once MOEA/D is initialised, the algorithm tries to generate new solutions for each subproblem. To this end and so long as the stop condition is not reached (line 5), for each reference point in  $R$ , the algorithm selects two random solutions in its neighbourhood (lines 7 to 8). Then, it generates a new solution through crossover and mutation operators (lines 9 to 10), which are the same as described for NSGA-II and SPEA2. Next, we update the neighbourhood solutions of the reference point  $i$  according to the new individual generated (lines 11 to 15). Finally, the set of non-dominated solutions is updated adding the new solution and removing all dominated ones (line 16).

## 6. The solving strategy

In this section we discuss some important details about the solving strategy followed. Firstly, we describe the data set considered. Next, we present the experimental methodology and the strategy assumed to configure the algorithms.

### 6.1. The testbed considered

We assume a data set composed of four different scenarios with sizes 50x50, 100x100, 200x200, and 300x300. A tradi-



Table 4: Description of the data set considered.

Instance ( $d_x \times d_y$ )	$\tilde{s}_s$	Fitness values ( $\tilde{s}_r = 0$ )			Test cases ( $\tilde{s}_r > 0$ )	Hyp. reference points (ideal, nadir)		
		$f_1$	$f_2$	$f_3$		$f_1$	$f_2$	$f_3$
50x50	4	0.0353	0.9175	0.9964	1	(0.02,0.04)	(1.00,0.60)	(1.00,0.50)
100x100	15	0.1091	0.8924	0.9567	2,3	(0.02,0.10)	(1.00,0.60)	(1.00,0.50)
200x200	57	0.2791	0.8710	0.9323	2,4,6,9	(0.10,0.30)	(1.00,0.60)	(1.00,0.50)
300x300	128	0.4225	0.7644	0.8528	6,12,18,24	(0.04,0.50)	(1.00,0.60)	(1.00,0.50)

General parameters					
$\alpha = 2$	$\beta = 1$	$amp = 100pJ/bit/m^2$	$co_{th} = 0.70$	$err = 0.1$	
$lec = 5J$	$k = 128KB$	$r_c = 30m$	$r_s = 15m$		

tional WSN is deployed in each of them, considering the lower bound of the minimum number of sensors to cover the whole surface. Thus, if a sensor covers a circumference of radius  $r_s$  and the area of the scenario is  $d_x \times d_y$ , this value is given by

$$\tilde{s}_s = \left\lceil \frac{d_x \cdot d_y}{\pi \cdot r_s^2} \right\rceil. \quad (20)$$

The traditional WSNs are deployed assuming a singleobjective GA, which optimises the global coverage provided by the network, being the collector node placed on the middle of the surface. The sensor coordinates obtained are detailed in [14].

Table 4 describes the data set considered, where the *general parameters* are obtained from [23] and [25]. Being as adding routers increases the network cost, we do not include more than 20% of these devices regarding the number of sensors. In this table, the *test cases* field indicates the number of RNs considered. Thus, we define a test case for the 50x50 instance, two for the 100x100, four for the 200x200, and four for the 300x300. The *fitness values* field shows the values of the objective functions without considering any router.

## 6.2. Experimental methodology

The data set is optimised by considering the metaheuristics previously discussed. For this purpose, we perform 31 independent runs for each algorithm and test case, being 30 runs a widely accepted value to reach statistical conclusions [59]. As stop condition, we assume several criteria in order to study the convergence of the algorithms. Criteria based on the number of evaluations are fairer than others, such as elapsed time, which depends on the computer considered. Accordingly, we assume 50 000, 100 000, 200 000, 300 000, and 400 000 evaluations.

## 6.3. Configuring the algorithms

All the algorithms were configured before performing the experiments. To this end and starting from default values, a parameter of the algorithm is selected to be tuned. Then, 31 independent runs are performed for each configuration of the parameter, considering a reduced stop condition (50 000 evaluations). Next, the configuration which provides the best performance on average is selected, assuming the hypervolume metric [60] as quality indicator. Next, another parameter is selected so long as all of them are fixed. Table 5 shows the range of values considered and the configurations selected. Note that the reference points assumed to calculate the hypervolume are in the *Hyp. reference points* field of Table 4. These values were obtained experimentally, where *ideal* and *nadir* are the best and

Table 5: Parametric swap.

NSGA-II			MO-VNS		
Parameter	Selected	Range	Parameter	Selected	Range
mutation	<b>0.80</b>	[0.05,0.10,...,0.95]	mutation	<b>0.10</b>	[0.05,0.10,...,0.95]
crossover	<b>0.80</b>	[0.05,0.10,...,0.95]	$k_{max}$	<b>10</b>	[3,4,...,14]
PS	<b>50</b>	[25,50,...,300]	$dv$	<b>2.0</b>	[1.0,1.5,...,5.0]

SPEA2			MO-ABC		
Parameter	Selected	Range	Parameter	Selected	Range
mutation	<b>0.70</b>	[0.05,0.10,...,0.95]	limit	<b>15</b>	[5,10,...,60]
crossover	<b>0.60</b>	[0.05,0.10,...,0.95]	EMP	<b>0.25</b>	[0.20,0.25,...,0.80]
PS	<b>50</b>	[25,50,...,300]	PS	<b>50</b>	[25,50,...,300]

MO-FA			MOEA/D		
Parameter	Selected	Range	Parameter	Selected	Range
$\gamma$	<b>0.60</b>	[0.05,0.10,...,0.95]	mutation	<b>0.25</b>	[0.05,0.10,...,0.95]
$\alpha_{FA}$	<b>0.85</b>	[0.05,0.10,...,0.95]	nSize	<b>0.05</b>	[0.05,0.10,...,0.95]
$\beta_0$	<b>0.70</b>	[0.05,0.10,...,0.95]	CHIM <sub>inc</sub>	<b>1.30</b>	[1.00,1.05,...,2.00]
mutation	<b>0.10</b>	[0.05,0.10,...,0.95]	crowDistance	<b>0.015</b>	[0.010,0.015,...,0.050]
mafa <sub>th</sub>	<b>0.25</b>	[0.05,0.10,...,0.95]	crossover	<b>0.15</b>	[0.05,0.10,...,0.95]
PS	<b>100</b>	[25,50,...,300]			

the worst value of a fitness function, respectively. These reference points were also considered to determine the extreme points  $F^1$ ,  $F^2$ , and  $F^3$  in the MOEA/D algorithm.

## 7. Computational results

In this section we apply the metaheuristics to optimise a freely available data set. The results obtained are analysed from two different viewpoints. On the one hand, we study the quality of the results in MO terms. On the other hand, we analyse the impact of the optimisation on the objective functions.

### 7.1. Analysing the algorithms in multiobjective terms

Table 6 shows average hypervolume ( $\overline{Hyp}$ ) and InterQuartile Range (IQR) for each algorithm, test case, and stop condition, where higher hypervolumes are shaded. Analysing this table, some algorithms seem to provide better performance than others. However, these differences could not be significant. To check this, we follow the methodology shown in Fig. 2.

This way, we study if data come from a normal distribution through Kolmogorov-Smirnov-Lilliefors [61] and Shapiro-Wilk's [62] tests, considering the hypothesis:  $H_0$  if data follow a normal distribution and  $H_1$  otherwise. P-values lower than 0.05 were obtained. Hence, we cannot assume  $H_1$ .

Next, we study if there are significant differences among the algorithms. To this end, as samples are independent and data do not follow a normal distribution, we assume the Wilcoxon-Mann-Whitney's [63] test with hypothesis:  $H_0$  if  $\overline{Hyp}_i \leq \overline{Hyp}_j$  and  $H_1$  if  $\overline{Hyp}_i > \overline{Hyp}_j$ , assuming  $i = 1, 2, \dots, 6$ ,  $j = 2, 3, \dots, 6$ ,  $i < j$ , 1=MO-FA, 2=MO-ABC, 3=MO-VNS, 4=SPEA2, 5=NSGA-II, and 6=MOEA/D.

The p-values are analysed assuming a significance level of 0.05. Based on them, Table 7 compares all the algorithms two by two for 50x50, 100x100, 200x200, 300x300 and all the scenarios, showing the percentage of test cases in which the algorithms provide better (rows) and worse (columns) significant performance than each of the others. This table also shows a ranking based on the average performance regarding all the other algorithms. This is a more realistic way to establish the performance ranking than other methods, such as considering only the algorithms having the best significant performance in the largest number of test cases. Following this less realistic

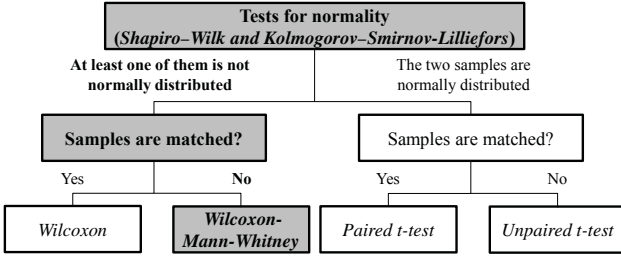


Figure 2: Statistical methodology followed.

approach, it is possible to consider an algorithm which is the best in many test cases, but it is the worst in others. Our aim is to define a ranking based on the average performance.

According to Table 7, MO-FA is the best algorithm for 50x50 and 200x200, MO-VNS for 100x100, and MOEA/D for 300x300. Being MO-FA the best algorithm on average, followed by MOEA/D, MO-VNS, MO-ABC, SPEA2, and NSGA-II. This behaviour is displayed in Fig. 3, where we show the differences among the algorithms over the number of evaluations. In this figure, we note as MO-FA has a good convergence rate, providing good results for reduced stopping criteria.

The set coverage metric [7] is also considered. To this end, we compare the median Pareto fronts previously obtained. Table 8a shows the average set coverage for 50 000, 200 000, and 400 000 evaluations. Higher average values for each stop condition are shaded. Accordingly, MO-FA is the best algorithm for 50 000 evaluations. For the next stop condition, MO-FA is the best in 50x50 and 200x200 scenarios, MO-ABC in 300x300, and MO-VNS in 100x100. Finally, for 400 000 evaluations, MO-FA is the best in 50x50 and 200x200, MO-ABC in 300x300, and MO-VNS in 100x100. Based on this study, Table 8b shows the average set coverage among the algorithms for all the stop conditions and instances. Being MO-FA the algorithm providing the best average behaviour, followed by MO-VNS, MO-ABC, SPEA2, NSGA-II, and MOEA/D.

Analysing the results obtained by the two MO quality metrics, we reach similar conclusions for MO-FA, MO-VNS, MO-ABC, SPEA2, and NSGA-II. However, this is not the case for MOEA/D. This is the second algorithm in hypervolume terms, but it is the last for the set coverage metric. This means that MOEA/D explores a greater part of the search space than the others. Thus, many solutions obtained by MOEA/D are dominated by the other algorithms; nevertheless, MOEA/D is able to find solutions where the other algorithms are unable, increasing the hypervolume metric. We note this behaviour in Fig. 4, especially in the third plot, where we compare the median Pareto fronts obtained by solving some representative test cases.

## 7.2. Impact of the optimisation on the fitness functions

In this subsection, we discuss the impact provided by the addition of RNs to traditional WSNs. Table 9 shows extreme values of the median Pareto front obtained by MO-FA for 400 000 evaluations. Each solution is associated with two quality metrics: *gain* and *eff*. *Gain* denotes the percentage in which a fitness function is increased or decreased, regarding the use of a

traditional WSN (see Table 4). On the other hand, *eff* measures the efficiency of the optimisation by dividing the gain obtained between the number of routers assumed. According to this table, AEC is decreased up to 63.54%, and ASA and NR are increased up to 16.36% and 10.20%, respectively. We note that as the number of routers increases, the efficiency of the optimisation decreases. This way, the maximum efficiency is habitually obtained considering a reduced number of routers.

In summary, the addition of RNs seems to be a good way to optimise traditional WSNs. However, the efficiency of this approach could be reduced if many routers are considered.

## 8. Other approaches

We started this work assuming an important limitation: we did not find any paper fitting this problem definition. Hence, we cannot directly compare the results obtained to other author approaches. To alleviate this fact, we selected a paper from the current literature considering a similar approach: Hou et al [3] implemented a heuristic called SPINDS, which optimises the network lifetime by adding routers to TT-WSNs.

This network model is composed of several clusters and a Base Station (BS). Each cluster consists of a set of MicroSensor Nodes (MSNs) and one Aggregation and Forwarding Node (AFN). MSNs capture and send information via single-hop transmission to the local AFN. Each AFN relays all the received information to the BS via multi-hop routing. Finally, the RNs are additional devices, forwarding all the information received from AFNs to the BS. In this model, all the devices are powered by batteries, excepting the BS. In addition, each device can assume a different initial energy charge.

As both network models are different, we assume some assumptions to adapt this model to our problem definition:

- We consider that a cluster is a sensor node. This way, MSNs are not included and AFNs are the new sensors.
- In the original model, the amount of information relayed by an AFN is given by the number of MSNs in its cluster. Now, we assume that all the AFNs send an information packet of size  $k$  per time unit.
- All the AFNs have a same initial energy charge in their batteries. Having RNs and BS an unlimited power supply.
- We assume the energy model described in our work.

We implemented the SPINDS algorithm following these assumptions. The results obtained optimising our data set appear in Table 10, where the value of the fitness functions have associated the *gain* and *eff* metrics previously defined in Section 7.2. Comparing Tables 9 and 10, we note that MO-FA provides a better behaviour for all the test cases. In fact, all the results provided by SPINDS are dominated by MO-FA.

Table 6: Average hypervolumes obtained. Higher values for each test case and stop condition are shaded.

NSGA-II( $\overline{Hyp} \%IQR$ )						SPEA2( $\overline{Hyp} \%IQR$ )					
$d_x \times d_y(\bar{s}_r)$	50 000	100 000	200 000	300 000	400 000	50 000	100 000	200 000	300 000	400 000	
50x50(1)	64.52%, 0.0000	64.52%, 0.0000	64.52%, 0.0000	64.52%, 0.0000	64.52%, 0.0000	64.52%, 0.0000	64.52%, 0.0000	64.52%, 0.0000	64.52%, 0.0000	64.52%, 0.0000	
100x100(2)	40.99%, 0.0052	41.28%, 0.0032	41.36%, 0.0033	41.47%, 0.0002	41.48%, 0.0002	41.18%, 0.0033	41.28%, 0.0030	41.45%, 0.0002	41.45%, 0.0002	41.46%, 0.0003	
100x100(3)	53.78%, 0.0040	54.18%, 0.0043	54.49%, 0.0026	54.60%, 0.0013	54.60%, 0.0015	54.05%, 0.0063	54.43%, 0.0033	54.71%, 0.0036	54.77%, 0.0047	54.87%, 0.0045	
200x200(2)	32.40%, 0.0125	33.01%, 0.0066	33.44%, 0.0063	33.58%, 0.0042	33.66%, 0.0036	32.54%, 0.0090	33.03%, 0.0076	33.23%, 0.0068	33.38%, 0.0063	33.43%, 0.0061	
200x200(4)	43.40%, 0.0243	44.54%, 0.0257	45.53%, 0.0223	46.08%, 0.0217	46.41%, 0.0221	43.11%, 0.0196	44.25%, 0.0283	45.34%, 0.0222	45.74%, 0.0176	46.00%, 0.0190	
200x200(6)	54.07%, 0.0188	56.14%, 0.0220	57.43%, 0.0167	58.15%, 0.0200	58.58%, 0.0183	54.71%, 0.0199	56.72%, 0.0162	58.00%, 0.0197	58.71%, 0.0177	59.17%, 0.0161	
200x200(9)	63.42%, 0.0254	65.83%, 0.0232	68.04%, 0.0204	68.93%, 0.0123	69.48%, 0.0134	64.93%, 0.0294	67.56%, 0.0242	69.21%, 0.0224	70.08%, 0.0138	70.62%, 0.0133	
300x300(6)	30.03%, 0.0078	30.91%, 0.0081	31.40%, 0.0107	31.71%, 0.0098	31.90%, 0.0111	30.37%, 0.0132	31.21%, 0.0141	31.86%, 0.0139	32.20%, 0.0138	32.35%, 0.0148	
300x300(12)	33.09%, 0.0134	34.32%, 0.0164	35.32%, 0.0165	35.90%, 0.0186	36.37%, 0.0212	33.90%, 0.0145	35.25%, 0.0202	36.54%, 0.0121	37.23%, 0.0123	37.75%, 0.0130	
300x300(18)	34.66%, 0.0096	36.38%, 0.0171	37.67%, 0.0194	38.46%, 0.0201	38.92%, 0.0200	36.01%, 0.0154	37.81%, 0.0127	39.07%, 0.0145	39.84%, 0.0139	40.36%, 0.0167	
300x300(24)	37.55%, 0.0126	38.88%, 0.0146	40.44%, 0.0173	41.30%, 0.0187	41.58%, 0.0167	38.51%, 0.0107	40.08%, 0.0123	41.77%, 0.0101	42.72%, 0.0137	43.31%, 0.0168	
MO-VNS( $\overline{Hyp} \%IQR$ )						MO-ABC( $\overline{Hyp} \%IQR$ )					
$d_x \times d_y(\bar{s}_r)$	50 000	100 000	200 000	300 000	400 000	50 000	100 000	200 000	300 000	400 000	
50x50(1)	64.60%, 0.0002	64.61%, 0.0001	64.62%, 0.0002	64.62%, 0.0001	64.63%, 0.0001	64.60%, 0.0000	64.60%, 0.0000	64.60%, 0.0000	64.60%, 0.0000	64.60%, 0.0001	
100x100(2)	41.75%, 0.0005	41.79%, 0.0004	41.81%, 0.0003	41.81%, 0.0002	41.82%, 0.0001	41.40%, 0.0068	41.40%, 0.0053	41.50%, 0.0055	41.50%, 0.0049	41.50%, 0.0048	
100x100(3)	54.96%, 0.0062	55.17%, 0.0039	55.45%, 0.0071	55.56%, 0.0063	55.61%, 0.0065	54.70%, 0.0070	55.00%, 0.0082	55.10%, 0.0098	55.10%, 0.0099	55.10%, 0.0099	
200x200(2)	31.76%, 0.0451	34.00%, 0.0078	34.60%, 0.0180	35.22%, 0.0149	35.49%, 0.0135	32.40%, 0.0103	32.70%, 0.0105	32.80%, 0.0079	32.90%, 0.0082	32.90%, 0.0083	
200x200(4)	42.81%, 0.0232	44.38%, 0.0284	45.24%, 0.0245	45.78%, 0.0213	46.14%, 0.0198	39.80%, 0.0289	40.70%, 0.0230	41.40%, 0.0236	41.40%, 0.0256	41.40%, 0.0256	
200x200(6)	54.27%, 0.0300	56.20%, 0.0192	56.80%, 0.0208	57.13%, 0.0219	57.47%, 0.0208	47.00%, 0.0509	49.20%, 0.0429	50.80%, 0.0430	51.00%, 0.0426	51.20%, 0.0413	
200x200(9)	63.48%, 0.0225	64.40%, 0.0198	65.33%, 0.0143	65.87%, 0.0195	66.45%, 0.0179	60.20%, 0.0443	62.30%, 0.0426	63.30%, 0.0337	63.50%, 0.0403	63.70%, 0.0408	
300x300(6)	30.39%, 0.0067	30.93%, 0.0105	31.23%, 0.0082	31.34%, 0.0079	31.40%, 0.0079	30.00%, 0.0103	31.00%, 0.0142	31.50%, 0.0110	31.50%, 0.0108	31.80%, 0.0127	
300x300(12)	33.88%, 0.0110	34.56%, 0.0113	35.31%, 0.0110	35.68%, 0.0096	35.83%, 0.0099	34.80%, 0.0150	36.50%, 0.0136	37.70%, 0.0115	38.20%, 0.0119	38.40%, 0.0118	
300x300(18)	37.04%, 0.0101	37.83%, 0.0107	38.48%, 0.0080	38.77%, 0.0066	39.01%, 0.0078	39.20%, 0.0130	41.10%, 0.0136	42.60%, 0.0120	43.50%, 0.0145	44.00%, 0.0118	
300x300(24)	40.14%, 0.0159	40.85%, 0.0126	41.48%, 0.0099	41.79%, 0.0084	41.95%, 0.0068	42.90%, 0.0164	44.90%, 0.0127	46.80%, 0.0114	47.70%, 0.0115	48.30%, 0.0121	
MO-FA( $\overline{Hyp} \%IQR$ )						MOEA/D( $\overline{Hyp} \%IQR$ )					
$d_x \times d_y(\bar{s}_r)$	50 000	100 000	200 000	300 000	400 000	50 000	100 000	200 000	300 000	400 000	
50x50(1)	64.63%, 0.0000	64.63%, 0.0000	64.63%, 0.0000	64.63%, 0.0000	64.63%, 0.0000	64.62%, 0.0001	64.62%, 0.0000	64.63%, 0.0000	64.63%, 0.0000	64.63%, 0.0000	
100x100(2)	41.66%, 0.0013	41.71%, 0.0010	41.75%, 0.0005	41.77%, 0.0003	41.78%, 0.0004	41.07%, 0.0027	41.20%, 0.0021	41.31%, 0.0016	41.35%, 0.0015	41.39%, 0.0010	
100x100(3)	54.79%, 0.0055	55.19%, 0.0013	55.29%, 0.0011	55.35%, 0.0014	55.38%, 0.0014	54.82%, 0.0047	55.12%, 0.0046	55.36%, 0.0063	55.42%, 0.0069	55.48%, 0.0068	
200x200(2)	35.05%, 0.0155	35.57%, 0.0049	35.98%, 0.0019	36.00%, 0.0020	36.06%, 0.0019	32.32%, 0.0180	32.76%, 0.0186	33.27%, 0.0147	33.54%, 0.0116	33.71%, 0.0127	
200x200(4)	43.65%, 0.0272	44.58%, 0.0231	45.21%, 0.0162	45.56%, 0.0117	45.91%, 0.0173	43.85%, 0.0481	44.82%, 0.0383	46.18%, 0.0234	46.72%, 0.0213	46.69%, 0.0198	
200x200(6)	55.22%, 0.0346	56.54%, 0.0383	57.89%, 0.0358	58.38%, 0.0409	58.96%, 0.0358	57.48%, 0.0160	58.62%, 0.0155	59.56%, 0.0192	60.01%, 0.0180	60.38%, 0.0171	
200x200(9)	65.87%, 0.0400	67.83%, 0.0269	69.82%, 0.0250	70.50%, 0.0239	70.94%, 0.0261	69.60%, 0.0167	70.96%, 0.0163	72.16%, 0.0177	72.87%, 0.0155	73.35%, 0.0150	
300x300(6)	30.28%, 0.0130	31.39%, 0.0202	32.59%, 0.0251	32.90%, 0.0173	33.02%, 0.0168	30.54%, 0.0057	31.25%, 0.0077	31.82%, 0.0074	32.08%, 0.0069	32.26%, 0.0053	
300x300(12)	34.63%, 0.0228	36.24%, 0.0238	37.76%, 0.0215	38.25%, 0.0192	39.17%, 0.0064	36.39%, 0.0152	37.56%, 0.0174	38.48%, 0.0174	38.85%, 0.0159	39.11%, 0.0154	
300x300(18)	37.97%, 0.0188	39.63%, 0.0187	41.18%, 0.0190	41.90%, 0.0189	42.54%, 0.0196	40.53%, 0.0257	42.22%, 0.0266	43.92%, 0.0219	44.74%, 0.0168	45.21%, 0.0178	
300x300(24)	40.83%, 0.0313	42.88%, 0.0207	44.35%, 0.0200	45.13%, 0.0217	45.59%, 0.0176	45.09%, 0.0202	47.51%, 0.0187	49.82%, 0.0242	51.04%, 0.0308	51.75%, 0.0275	

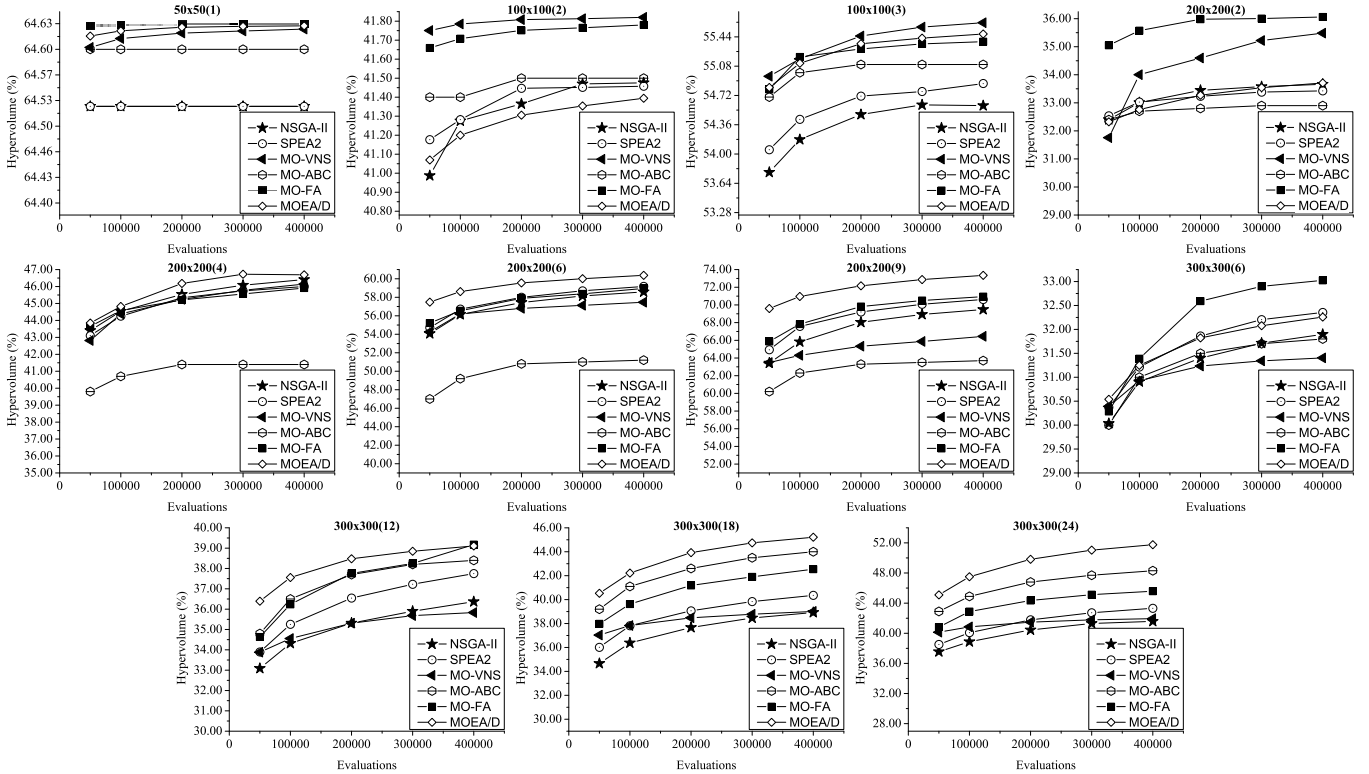


Figure 3: Convergence study based on the hypervolume indicator.

Table 7: Percentage of test cases in which the algorithms are significant better and worse, based on the hypervolume metric.

		It is worse (50x50)								It is worse (100x100)							
It is better	MOEA/D	MO-FA	MO-VNS	MO-ABC	NSGA-II	SPEA2	Percentage	It is better	MOEA/D	MO-FA	MO-VNS	MO-ABC	NSGA-II	SPEA2	Percentage	It is better	MOEA/D
	—	0.00%	0.00%	3.03%	7.58%	7.58%	24.24%		—	0.00%	0.00%	1.63%	4.07%	4.07%	9.76%		—
	MO-FA	7.58%	—	7.58%	7.58%	7.58%	37.88%		MO-FA	4.88%	—	0.00%	4.07%	8.13%	25.20%		MO-FA
	MO-VNS	0.00%	0.00%	—	0.00%	7.58%	15.15%		MO-VNS	5.69%	5.69%	—	6.50%	8.13%	34.15%		MO-VNS
	MO-ABC	3.03%	0.00%	—	—	7.58%	22.73%		MO-ABC	4.07%	0.00%	0.00%	—	8.13%	20.33%		MO-ABC
	NSGA-II	0.00%	0.00%	0.00%	—	—	0.00%		NSGA-II	2.44%	0.00%	0.00%	0.00%	1.63%	4.07%		NSGA-II
	SPEA2	0.00%	0.00%	0.00%	0.00%	—	0.00%		SPEA2	2.44%	0.00%	0.00%	0.00%	4.07%	6.50%		SPEA2
Percentage		10.61%	0.00%	18.18%	10.61%	30.30%	100.00%	Percentage		19.51%	5.69%	0.00%	12.20%	32.52%	100.00%	Percentage	
		It is worse (200x200)								It is worse (300x300)							
It is better	MOEA/D	MO-FA	MO-VNS	MO-ABC	NSGA-II	SPEA2	Percentage	It is better	MOEA/D	MO-FA	MO-VNS	MO-ABC	NSGA-II	SPEA2	Percentage	It is better	MOEA/D
	—	3.20%	5.02%	8.22%	4.57%	5.94%	26.94%		—	4.74%	5.93%	5.93%	5.93%	5.93%	28.46%		—
	MO-FA	3.20%	—	6.85%	9.13%	4.57%	30.59%		MO-FA	1.98%	—	7.51%	1.58%	7.51%	25.69%		MO-FA
	MO-VNS	1.83%	0.00%	—	8.68%	1.83%	14.16%		MO-VNS	0.40%	0.00%	0.40%	5.93%	1.19%	5.53%		MO-VNS
	MO-ABC	0.00%	0.00%	0.00%	—	0.00%	0.00%		MO-ABC	0.00%	3.95%	6.32%	5.93%	5.93%	22.13%		MO-ABC
	NSGA-II	0.00%	0.00%	2.74%	8.68%	0.91%	12.33%		NSGA-II	0.00%	1.19%	0.00%	5.93%	0.00%	2.37%		NSGA-II
	SPEA2	0.00%	0.00%	3.65%	8.68%	—	15.98%		SPEA2	1.98%	0.00%	4.74%	1.58%	7.51%	15.81%		SPEA2
Percentage		5.02%	3.20%	18.26%	43.38%	16.89%	100.00%	Percentage		5.53%	8.70%	25.69%	9.49%	20.16%	100.00%	Percentage	
		It is worse (all the instances)															
It is better	MOEA/D	MO-FA	MO-VNS	MO-ABC	NSGA-II	SPEA2	Percentage	It is better	MOEA/D	MO-FA	MO-VNS	MO-ABC	NSGA-II	SPEA2	Percentage	It is better	MOEA/D
	—	2.87%	4.54%	5.60%	5.30%	5.75%	24.05%		—	3.48%	—	5.90%	5.14%	7.41%	28.44%		—
	MO-FA	3.48%	—	5.90%	5.14%	7.41%	14.67%		MO-FA	1.82%	1.06%	—	4.24%	3.33%	14.67%		MO-FA
	MO-VNS	1.82%	1.06%	—	4.24%	3.33%	14.67%		MO-VNS	1.06%	1.51%	2.87%	4.54%	4.54%	14.52%		MO-VNS
	MO-ABC	0.91%	0.00%	1.36%	2.87%	0.61%	5.75%		MO-ABC	0.91%	0.00%	1.36%	2.87%	0.61%	5.75%		MO-ABC
	NSGA-II	1.21%	0.00%	3.03%	3.48%	—	12.56%		NSGA-II	1.21%	0.00%	3.03%	3.48%	—	12.56%		NSGA-II
Percentage		8.47%	5.45%	17.70%	21.33%	26.32%	100.00%	Percentage		8.47%	5.45%	17.70%	21.33%	26.32%	100.00%	Percentage	

Table 8: Set coverage study among the algorithms. Notation assumed A=MO-ABC, D=MOEA/D, F=MO-FA, N=NSGA-II, S=SPEA2, and V=MO-VNS. (a) for 50 000, 200 000, and 400 000 evaluations. (b) for all the stop conditions and instances.

NSGA-II (50 000)						
$d_x \times d_y$	S	V	A	F	D	Avg.
50x50	100.00%	8.11%	5.56%	2.13%	7.89%	24.74%
100x100	34.24%	0.00%	0.68%	0.25%	38.31%	14.70%
200x200	38.42%	15.53%	40.60%	1.34%	43.41%	27.86%
300x300	31.53%	3.74%	23.30%	5.12%	23.57%	17.45%
Avg.	51.05%	6.84%	17.53%	2.21%	28.30%	
SPEA2 (50 000)						
$d_x \times d_y$	N	V	A	F	D	Avg.
50x50	100.00%	8.11%	5.56%	2.13%	7.89%	24.74%
100x100	54.00%	1.93%	8.17%	2.09%	48.17%	22.87%
200x200	45.61%	14.95%	44.22%	2.79%	39.57%	29.43%
300x300	55.82%	16.37%	31.71%	15.75%	40.84%	32.10%
Avg.	63.86%	10.34%	22.41%	5.69%	34.12%	
MO-VNS (50 000)						
$d_x \times d_y$	S	N	A	F	D	Avg.
50x50	77.07%	81.77%	55.56%	25.53%	65.79%	61.14%
100x100	88.10%	94.57%	57.94%	31.27%	78.78%	70.13%
200x200	57.74%	60.94%	62.03%	2.07%	37.96%	44.15%
300x300	75.03%	88.95%	35.13%	28.81%	49.21%	55.43%
Avg.	74.48%	81.56%	52.66%	21.92%	57.93%	
MO-ABC (50 000)						
$d_x \times d_y$	S	N	V	F	D	Avg.
50x50	92.99%	90.06%	37.84%	6.38%	50.00%	55.45%
100x100	68.33%	85.81%	11.21%	11.71%	72.51%	49.91%
200x200	23.51%	30.26%	21.21%	4.55%	29.31%	21.77%
300x300	59.75%	70.62%	45.14%	29.75%	43.60%	49.77%
Avg.	61.15%	69.19%	28.85%	13.10%	48.85%	
MO-FA (50 000)						
$d_x \times d_y$	S	N	V	A	D	Avg.
50x50	100.00%	100.00%	70.27%	94.44%	81.58%	89.26%
100x100	87.10%	95.44%	30.40%	55.65%	87.83%	71.28%
200x200	87.41%	93.66%	91.40%	95.07%	67.28%	86.96%
300x300	75.92%	94.52%	48.84%	43.93%	41.15%	60.87%
Avg.	87.61%	95.90%	60.23%	72.27%	69.46%	
MOEA/D (50 000)						
$d_x \times d_y$	S	N	V	A	F	Avg.
50x50	73.89%	80.66%	21.62%	36.11%	12.77%	45.01%
100x100	17.76%	22.51%	1.23%	8.14%	1.46%	10.22%
200x200	17.20%	15.00%	9.31%	16.11%	3.72%	12.27%
300x300	10.86%	25.79%	7.02%	21.49%	15.02%	16.03%
Avg.	29.93%	35.99%	9.79%	20.46%	8.24%	
NSGA-II (200 000)						
	S	V	A	F	D	Avg.
100.00%	2.33%	4.44%	2.04%	2.33%	22.23%	
57.53%	0.00%	18.42%	2.59%	26.79%	21.07%	
41.77%	39.24%	59.14%	8.73%	66.35%	43.05%	
30.51%	45.41%	6.95%	13.91%	36.61%	26.68%	
57.45%	21.75%	22.24%	6.82%	33.02%		
SPEA2 (200 000)						
	N	V	A	F	D	Avg.
100.00%	2.33%	4.44%	2.04%	2.33%	22.23%	
56.28%	0.14%	19.97%	2.85%	35.09%	22.87%	
36.90%	31.93%	57.03%	3.68%	59.05%	37.72%	
56.42%	63.58%	4.90%	15.40%	40.78%	36.21%	
62.40%	24.49%	21.58%	5.99%	34.31%		
MO-VNS (200 000)						
	S	N	A	F	D	Avg.
91.07%	90.48%	82.22%	38.78%	72.09%	74.93%	
92.84%	92.57%	50.40%	51.99%	80.27%	73.61%	
57.69%	52.75%	66.63%	13.01%	59.63%	49.94%	
36.71%	38.47%	7.48%	15.82%	35.16%	26.73%	
69.58%	68.57%	51.68%	29.90%	61.79%		
MO-ABC (200 000)						
	S	N	V	F	D	Avg.
91.52%	90.48%	9.30%	12.24%	32.56%	47.22%	
54.18%	56.37%	7.55%	23.63%	54.02%	39.15%	
21.04%	28.43%	23.70%	2.95%	40.52%	23.33%	
80.02%	81.66%	86.85%	60.99%	49.57%	71.82%	
61.69%	64.23%	31.85%	24.95%	44.17%		
MO-FA (200 000)						
	S	N	V	A	D	Avg.
100.00%	100.00%	58.14%	91.11%	88.37%	87.52%	
86.82%	80.37%	22.15%	48.24%	69.01%	61.32%	
78.67%	77.02%	71.28%	93.07%	85.76%	81.16%	
73.05%	73.04%	75.94%	13.83%	46.31%	56.43%	
84.64%	82.61%	56.88%	61.56%	72.36%		
MOEA/D (200 000)						
	S	N	V	A	F	Avg.
91.07%	90.48%	18.60%	60.00%	10.20%	54.07%	
39.99%	39.83%	2.02%	13.97%	4.90%	20.14%	
12.10%	3.39%	11.08%	8.65%	0.00%	7.04%	
9.81%	11.22%	18.12%	3.20%	15.42%	11.55%	
38.24%	36.23%	12.46%	21.46%	7.63%		
NSGA-II (400 000)						
	S	V	A	F	D	Avg.
100.00%	7.55%	5.00%	2.04%	2.13%	23.34%	
64.70%	1.49%	16.97%	6.14%	43.87%	26.63%	
33.07%	40.22%	43.33%	3.18%	66.67%	37.29%	
14.21%	60.15%	3.42%	19.59%	42.70%	28.01%	
53.22%	18.10%	17.26%	5.68%	34.19%		
SPEA2 (400 000)						
	N	V	A	F	D	Avg.
100.00%	7.55%	5.00%	2.04%	2.13%	23.34%	
70.89%	0.37%	18.33%	6.39%	47.39%	28.68%	
59.94%	38.79%	51.12%	5.21%	66.03%	44.22%	
68.12%	75.86%	11.21%	37.09%	45.18%	47.49%	
67.71%	30.64%	21.42%	12.68%	40.18%		
MO-VNS (400 000)						
	S	N	A	F	D	Avg.
80.08%	80.17%	87.50%	75.51%	82.98%	81.25%	
96.69%	93.48%	63.03%	52.46%	88.94%	78.92%	
49.91%	47.13%	56.65%	9.60%	60.37%	44.73%	
16.45%	32.01%	5.91%	17.59%	38.85%	22.16%	
60.78%	72.69%	53.27%	38.79%	67.78%		
MO-ABC (400 000)						
	S	N	V	F	D	Avg.
90.68%	90.72%	16.98%	10.20%	34.04%	48.52%	
51.45%	54.64%	11.59%	14.76%	58.42%	38.17%	
23.19%	34.18%	26.70%	9.06%	37.44%	26.11%	
73.88%	83.35%	85.11%	67.65%	49.81%	71.96%	
59.80%	66.52%	35.09%	25.42%	44.93%		
MO-FA (400 000)						
	S	N	V	A	D	Avg.
100.00%	100.00%	30.19%	92.50%	93.62%	83.26%	
82.72%	80.76%	29.85%	56.21%	81.39%	66.19%	
88.46%	84.58%	73.07%	78.95%	88.21%	82.65%	
43.42%	64.98%	71.92%	13.76%	53.16%	49.45%	
78.65%	86.95%	51.26%	60.35%	79.09%		
MOEA/D (400 000)						
	S	N	V	A	F	Avg.
90.68%	90.72%	20.75%	57.50%	6.12%	53.15%	
16.58%	18.07%	1.27%	3.64%	1.90%	8.29%	
12.42%	8.35%	17.80%	14.60%	0.00%	10.63%	
9.31%	24.26%	16.71%	11.08%	16.87%	15.65%	
32.25%	34.99%	14.13%	21.70%	6.22%		

MO-FA						
S	N	V	A	D	Avg.	
84.05%	86.95%	53.35%	63.98%	73.85%	72.44%	
MO-VNS						
S	N	V	A	F	D	Avg.
68.56%	72.69%	53.11%	32.87%	64.61%	58.37%	
MO-ABC						
S	N	V	F	D	Avg.	
61.57%	66.52%	31.94%	21.76%	48.70%	46.10%	
SPEA2						
N	V	A	F	D	Avg.	
67.71%	21.43%	20.57%	7.81%	37.41%	30.99%	
NSGA-II						
S	V	A	F	D	Avg.	
53.22%	18.10%	17.26%	5.68%	34.19%	25.69%	
MOEA/D						
S	N	V	A	F	Avg.	
33.77%	34.99%	11.27%	21.30%	7.46%	21.76%	

Table 9: Studying the extreme values of the median Pareto front obtained by MO-FA for 400 000 evaluations.

$d_x \times d_y(\bar{s}_i)$	max( $f_1$ :AEC)			min( $f_1$ :AEC)			max( $f_2$ :ASA)			min( $f_2$ :ASA)			max( $f_3$ :NR)			min( $f_3$ :NR)		
	value	gain	eff	value	gain	eff	value	gain	eff	value	gain	eff	value	gain	eff	value	gain	eff
50x50(1)	0.0302	-14.39%	-14.39%	0.0235	-33.31%	-33.31%	92.07%	0.35%	0.35%	88.98%	-3.02%	-3.02%	99.93%	0.29%	0.29%	99.86%	0.22%	0.22%
100x100(2)	0.0712	-34.72%	-17.36%	0.0563	-48.38%	-24.19%	91.56%	2.60%	1.30%	87.34%	-2.13%	-1.06%	98.65%	3.11%	1.56%	98.06%	2.50%	1.25%
100x100(3)	0.0572	-47.54%	-15.85%	0.0430	-60.63%	-20.21%	91.61%	2.66%	0.89%	82.40%	-7.66%	-2.55%	99.21%	3.70%	1.23%	98.18%	2.63%	0.88%
200x200(2)	0.2381	-14.71%	-7.35%	0.1874	-32.87%	-16.44%	88.79%	1.94%	0.97%	85.46%	-1.88%	-0.94%	94.91%	1.81%	0.90%	94.00%	0.83%	0.41%
200x200(4)	0.2514	-9.92%	-2.48%	0.1631	-41.57%	-10.39%	89.44%	2.69%	0.67%	85.18%	-2.21%	-0.55%	95.73%	2.68%	0.67%	94.40%	1.26%	0.31%
200x200(6)	0.2075	-25.64%	-4.27%	0.1257	-54.96%	-9.16%	90.01%	3.34%	0.56%	86.39%	-0.82%	-0.14%	96.52%	3.53%	0.59%	94.58%	1.44%	0.24%
200x200(9)	0.2085	-25.29%	-2.81%	0.1018	-63.54%	-7.06%	90.67%	4.09%	0.45%	85.61%	-1.71%	-0.19%	97.07%	4.12%	0.46%	95.18%	2.09%	0.23%
300x300(6)	0.3474	-17.79%	-2.96%	0.2306	-45.41%	-7.57%	88.57%	15.86%	2.64%	78.20%	2.31%	0.38%	91.33%	7.09%	1.18%	85.95%	0.79%	0.13%
300x300(12)	0.3218	-23.84%	-1.99%	0.1962	-53.57%	-4.46%	88.73%	16.08%	1.34%	77.97%	2.00%	0.17%	92.54%	8.51%	0.71%	86.40%	1.31%	0.11%
300x300(18)	0.3146	-25.53%	-1.42%	0.1819	-56.94%	-3.16%	88.85%	16.23%	0.90%	78.04%	2.10%	0.12%	93.41%	9.54%	0.53%	86.84%	1.83%	0.10%
300x300(24)	0.3719	-11.97%	-0.50%	0.1591	-62.34%	-2.60%	88.95%	16.36%	0.68%	75.77%	-0.88%	-0.04%	93.98%	10.20%	0.43%	86.61%	1.56%	0.07%

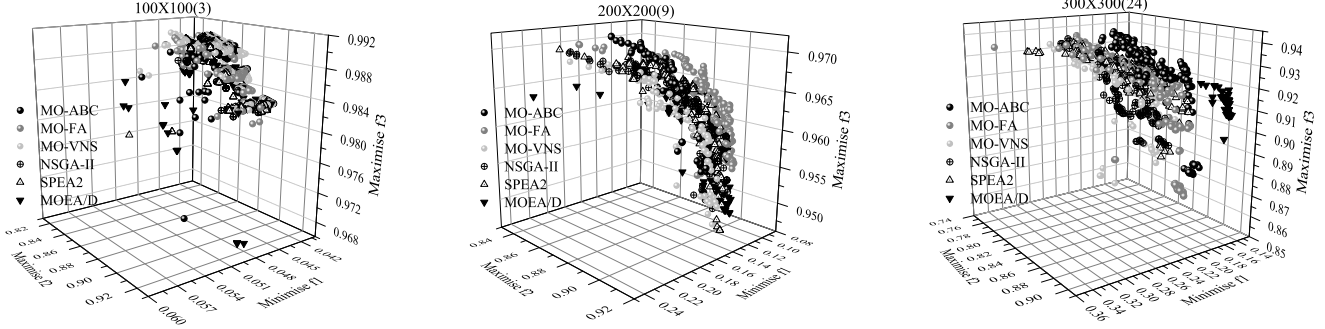


Figure 4: Comparing some representative median Pareto fronts for 400 000 evaluations.

Table 10: Solutions obtained through SPINDS heuristic.

$d_x \times d_y(\bar{s}_i)$	$f_1$ :AEC			$f_2$ :ASA			$f_3$ :NR		
	value	gain	eff	value	gain	eff	value	gain	eff
50x50(1)	0.0247	-29.89%	-29.89%	88.98%	-3.02%	-3.02%	99.86%	0.22%	0.22%
100x100(2)	0.0610	-44.11%	-22.06%	89.46%	0.25%	0.12%	98.54%	3.00%	1.50%
100x100(3)	0.0534	-51.06%	-17.02%	88.80%	-0.49%	-0.16%	98.95%	3.43%	1.14%
200x200(2)	0.2655	-4.89%	-2.44%	86.45%	-0.75%	-0.37%	94.13%	0.96%	0.48%
200x200(4)	0.1851	-33.69%	-8.42%	86.22%	-1.01%	-0.25%	94.46%	1.32%	0.33%
200x200(6)	0.1653	-40.77%	-6.80%	85.68%	-1.63%	-0.27%	94.46%	1.32%	0.22%
200x200(9)	0.1171	-58.05%	-6.45%	86.15%	-1.09%	-0.12%	94.65%	1.52%	0.17%
300x300(6)	0.3853	-8.80%	-1.47%	80.38%	5.16%	0.86%	90.16%	5.72%	0.95%
300x300(12)	0.3853	-8.80%	-0.73%	80.38%	5.16%	0.43%	90.17%	5.74%	0.48%
300x300(18)	0.3301	-21.86%	-1.21%	78.04%	2.10%	0.12%	89.62%	5.09%	0.28%
300x300(24)	0.3301	-21.86%	-0.91%	78.04%	2.10%	0.09%	89.62%	5.08%	0.21%

## 9. Final remarks

This paper deals with how to optimise traditional WSNs by adding energy-harvesting RNs, assuming a single-tiered network model. The purpose is to optimise three relevant factors: AEC, ASA, and NR. This problem is known as RNPP and was defined as an NP-hard optimisation problem in the literature. Hence, non-traditional techniques are required. We find many works assuming heuristics to this end. However, metaheuristics usually have a good behaviour solving such problems, providing a set of trade-off solutions, which allows the network designer to have several possibilities to deploy the network. This is not the case for heuristics, providing a unique solution.

This situation led us to assume several MO metaheuristics. Five of them belong to EC: MOEA/D, NSGA-II, SPEA2, MO-ABC, and MO-FA. Being MO-ABC and MO-FA swarm intelligence algorithms based on the behaviour of honey bees and fireflies, respectively. NSGA-II and SPEA2 are two standard GAs. MOEA/D is a state-of-the-art algorithm based on decomposition. The remainder is the trajectory algorithm MO-VNS.

The metaheuristics are applied to optimise a freely available

data set, considering four different scenarios where a traditional WSN is deployed in each of them. The results obtained are analysed assuming two standard MO tools: hypervolume and set coverage. Through a widely accepted statistical methodology, we conclude that MO-FA provides the best behaviour in medium and very small instances, MOEA/D in big instances, and MO-VNS in small instances. Being MO-FA the algorithm providing the best performance on average. In addition, we note that MOEA/D is able to explore a greater part of the search space than the other algorithms.

We also analyse the impact provided by the optimisation process. To this end, we calculate the percentage in which the fitness functions increase or decrease, regarding the routers considered. This way, we reach that the addition of RNs seems to be a good way to optimise such networks. However, this approach is limited, because the efficiency of the optimisation decreases if many routers are deployed.

We assume an important limitation. We did not find any work considering a similar problem. Hence, we cannot directly compare our results to other author methodologies. To solve this, we adapted the heuristic proposed by Hou et al [3] in order to optimise our data set. Analysing the results obtained, we note that our approach provides a better behaviour.

As future lines of research, it would be interesting to consider new metaheuristics and a bigger data set. Conducting real-world experiments would be also a good future extension.

## Acknowledgments

The authors thank the anonymous referees for comments and suggestions which have led to an improved version of this pa-

per. This research was funded by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund, under the contract TIN2012-30685 (BIO project).

## References

- [1] J. Y. B. Mukherjee, D. Ghosal, Wireless sensor network survey, *Computer Networks* 52 (2008) 2292–2330.
- [2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, *IEEE Communications Magazine* 40 (2002) 102–114.
- [3] Y. Hou, Y. Shi, H. Sherali, S. Midkiff, On energy provisioning and relay node placement for wireless sensor networks, *IEEE Transactions on Wireless Communications* 4 (2005) 2579–2590.
- [4] M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., 1979.
- [5] X. Cheng, B. Narahari, R. Simha, M. Cheng, D. Liu, Strong minimum energy topology in wireless sensor networks: Np-completeness and heuristics, *IEEE Transactions on Mobile Computing* 2 (2003) 248–256.
- [6] J.-H. Chang, L. Tassiulas, Maximum lifetime routing in wireless sensor networks, *IEEE/ACM Transactions on Networking* 12 (2004) 609–619.
- [7] E. Zitzler, *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. (Doctoral dissertation), Swiss Federal Institute of Technology (ETH), 1999.
- [8] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast elitist multi-objective genetic algorithm: Nsga-ii, *IEEE Transactions on Evolutionary Computation* 6 (2000) 182–197.
- [9] E. Zitzler, M. Laumanns, L. Thiele, *Spea2: Improving the strength pareto evolutionary algorithm*, Tech. rep., Computer Engineering and Networks Laboratory (TIK), ETH Zurich (2001).
- [10] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm, *J. of Global Optimization* 39 (2007) 459–471.
- [11] X.-S. Yang, Firefly algorithms for multimodal optimization, in: *Stochastic Algorithms: Foundations and Applications*, Vol. 5792 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 169–178.
- [12] Q. Zhang, H. Li, Moea/d: A multiobjective evolutionary algorithm based on decomposition, *IEEE Transactions on Evolutionary Computation* 11 (6) (2007) 712–731.
- [13] M. J. Geiger, Randomised variable neighbourhood search for multi objective optimisation, in: *Proceedings of EU/ME Workshop*, Vol. 0809.0271, 2008, pp. 34–42.
- [14] J. M. Lanza-Gutierrez, J. A. Gomez-Pulido, M. A. Vega-Rodriguez, J. M. Sanchez-Perez, Instance sets for optimization in wireless sensor networks, <http://arco.unex.es/wsnopt> (2011).
- [15] M. Cardei, D.-Z. Du, Improving wireless sensor network lifetime through power aware organization, *Wireless Networks* 11 (2005) 333–340.
- [16] H. Zhang, J. Hou, Maintaining sensing coverage and connectivity in large sensor networks, *Ad Hoc & Sensor Wireless Networks* 1 (2005) 89–124.
- [17] L. Liu, B. Hu, L. Li, Energy conservation algorithms for maintaining coverage and connectivity in wireless sensor networks, *IET Communications* 4 (2010) 786–800.
- [18] H. Mahboubi, K. Moezzi, A. Aghdam, K. Sayrafian-Pour, V. Marbukh, Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors, *IEEE Transactions on Industrial Informatics* 10 (2014) 163–174.
- [19] A. Konstantinidis, K. Yang, Q. Zhang, An evolutionary algorithm to a multi-objective deployment and power assignment problem in wireless sensor networks, in: *Proceedings of IEEE GLOBECOM*, 2008, pp. 1–6.
- [20] J. Jia, J. Chen, G. Chang, Z. Tan, Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm, *Computers and Mathematics with Applications* 57 (2009) 1756–1766.
- [21] J. Jia, J. Chen, G. Chang, Y. Wen, J. Song, Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius, *Computers and Mathematics with Applications* 57 (2009) 1767–1775.
- [22] X.-M. Hu, J. Zhang, Y. Yu, H.-H. Chung, Y.-L. Li, Y.-H. Shi, X.-N. Luo, Hybrid genetic algorithm using a forward encoding scheme for lifetime maximization of wireless sensor networks, *IEEE Transactions on Evolutionary Computation* 14 (2010) 766–781.
- [23] A. Konstantinidis, K. Yang, Multi-objective k-connected deployment and power assignment in wsns using a problem-specific constrained evolutionary algorithm based on decomposition, *Computer Communications* 34 (2011) 83–98.
- [24] M. Le Berre, F. Hnaïen, H. Snoussi, Multi-objective optimization in wireless sensors networks, in: *Proceedings of ICM*, Vol. 1, 2011, pp. 1–4.
- [25] F. Martins, E. Carrano, E. Wanner, R. Takahashi, G. Mateus, A hybrid multiobjective evolutionary approach for improving the performance of wireless sensor networks, *IEEE Sensors Journal* 11 (2011) 545–554.
- [26] Y. Yoon, Y.-H. Kim, An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks, *IEEE Transactions on Cybernetics* 43 (2013) 1473–1483.
- [27] S. Mini, S. Udgata, S. Sabat, Sensor deployment and scheduling for target coverage problem in wireless sensor networks, *IEEE Sensors Journal* 14 (2014) 636–644.
- [28] J. Tang, B. Hao, A. Sen, Relay node placement in large scale wireless sensor networks, *Computer Communications* 29 (2006) 490–501.
- [29] Q. Wang, K. Xu, G. Takahara, H. Hassanein, Device placement for heterogeneous wireless sensor networks: Minimum cost with lifetime constraints, *IEEE Transactions on Wireless Communications* 6 (2007) 2444–2453.
- [30] E. L. Lloyd, G. Xue, Relay node placement in wireless sensor networks, *IEEE Transactions on Computers* 56 (2007) 134–138.
- [31] X. Han, X. Cao, E. L. Lloyd, C.-C. Shen, Fault-tolerant relay node placement in heterogeneous wireless sensor networks, *IEEE Transactions on Mobile Computing* 9 (2010) 643–656.
- [32] K. Xu, H. Hassanein, G. Takahara, Q. Wang, Relay node deployment strategies in heterogeneous wireless sensor networks, *IEEE Transactions on Mobile Computing* 9 (2010) 145–159.
- [33] S. Misra, N. Majd, H. Huang, Constrained relay node placement in energy harvesting wireless sensor networks, in: *Proceedings of IEEE MASS*, Vol. 1, 2011, pp. 25–34.
- [34] S. Misra, N. E. Majd, H. Huang, Approximation algorithms for constrained relay node placement in energy harvesting wireless sensor networks, *IEEE Transactions on Computers* 99 (2013) PrePrints.
- [35] A. Nigam, Y. K. Agarwal, Optimal relay node placement in delay constrained wireless sensor network design, *European Journal of Operational Research* 233 (2014) 220–233.
- [36] C. Zhao, P. Chen, Particle swarm optimization for optimal deployment of relay nodes in hybrid sensor networks, in: *Proceedings of IEEE CEC*, Vol. 1, 2007, pp. 3316–3320.
- [37] A. Perez, M. Labrador, P. Wightman, A multiobjective approach to the relay placement problem in wsns, in: *Proceedings of IEEE WCNC*, Vol. 1, 2011, pp. 475–480.
- [38] A. Peiravi, H. R. Mashhadi, S. Hamed Javadi, An optimal energy-efficient clustering method in wireless sensor networks using multi-objective genetic algorithm, *International Journal of Communication Systems* 26 (2013) 114–126.
- [39] A. ur Rehman, A. Z. Abbasi, N. Islam, Z. A. Shaikh, A review of wireless sensors and networks: applications in agriculture, *Computer Standards and Interfaces* 36 (2) (2014) 263–270.
- [40] J. M. Lanza-Gutierrez, J. A. Gomez-Pulido, M. A. Vega-Rodriguez, J. M. Sanchez-Perez, Relay node positioning in wireless sensor networks by means of evolutionary techniques, in: *Autonomous and Intelligent Systems*, Vol. 7326 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2012, pp. 18–25.
- [41] J. M. Lanza-Gutierrez, J. A. Gomez-Pulido, M. A. Vega-Rodriguez, J. M. Sanchez-Perez, A parallel evolutionary approach to solve the relay node placement problem in wireless sensor networks, in: *Proceeding of GECCO*, 2013, pp. 1157–1164.
- [42] J. M. Lanza-Gutierrez, J. A. Gomez-Pulido, M. A. Vega-Rodriguez, A trajectory algorithm to solve the relay node placement problem in wireless sensor networks, in: *Theory and Practice of Natural Computing*, Vol. 8273 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 145–156.
- [43] J. M. Lanza-Gutierrez, J. A. Gomez-Pulido, M. A. Vega-Rodriguez, Intelligent relay node placement in heterogeneous wireless sensor networks for energy efficiency, *International Journal of Robotics and Automation* 29 (2014) 1–13.
- [44] J. M. Lanza-Gutierrez, J. A. Gomez-Pulido, M. A. Vega-Rodriguez, A new realistic approach for the relay node placement problem in wireless

sensor networks by means of evolutionary computation (accepted), *Ad Hoc and Sensor Wireless Networks*.

- [45] J. M. Lanza-Gutierrez, J. A. Gomez-Pulido, M. A. Vega-Rodriguez, A trajectory-based heuristic to solve a three-objective optimization problem for wireless sensor network deployment, in: *Applications of Evolutionary Computation (EVO\*2014-accepted)*, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2014.
- [46] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*(Third Edition), The MIT Press, 2009.
- [47] W. Ye, J. Heidemann, D. Estrin, An energy-efficient mac protocol for wireless sensor networks, in: *Proceedings of INFOCOM*, Vol. 3, 2002, pp. 1567–1576.
- [48] B. Wang, Coverage problems in sensor networks: A survey, *ACM Computing Surveys* 43 (2011) 32:1–32:53.
- [49] B. Deb, S. Bhatnagar, B. Nath, Reliable information forwarding using multiple paths in sensor networks, in: *Proceedings of IEEE LCN*, 2003, pp. 406–415.
- [50] J. W. Suurballe, Disjoint paths in a network, *Networks* 4 (1974) 125–145.
- [51] G. Anastasi, M. Conti, M. D. Francesco, A. Passarella, Energy conservation in wireless sensor networks: A survey, *Ad Hoc Networks* 7 (3) (2009) 537–568.
- [52] C. Zhu, C. Zheng, L. Shu, G. Han, A survey on coverage and connectivity issues in wireless sensor networks, *Journal of Network and Computer Applications* 35 (2) (2012) 619–632.
- [53] K. Islam, W. Shen, X. Wang, Wireless sensor network reliability and security in factory automation: A survey, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 42 (6) (2012) 1243–1256.
- [54] A. Konstantinidis, K. Yang, Multi-objective energy-efficient dense deployment in wireless sensor networks using a hybrid problem-specific moea/d, *Applied Soft Computing* 11 (2011) 4117–4134.
- [55] A. Konstantinidis, K. Yang, Q. Zhang, D. Zeinalipour-Yazti, A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks, *Computer Networks* 54 (2010) 960–976.
- [56] I. Das, J. Dennis, Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems, *Journal on Optimization* 8 (3) (1998) 631–657.
- [57] Q. Zhang, H. Li, D. Maringer, E. Tsang, Moea/d with nbi-style tchebycheff approach for portfolio management, in: *IEEE Congress on Evolutionary Computation (CEC)*, 2010, pp. 1–8.
- [58] A. Rubio-Largo, Q. Zhang, M. A. Vega-Rodriguez, A multiobjective evolutionary algorithm based on decomposition with normal boundary intersection for traffic grooming in optical networks, *Information Sciences* 289 (0) (2014) 91–116.
- [59] W. Hays, R. Winkler, *Statistics: probability, inference, and decision*, Holt, Rinehart and Winston, 1970.
- [60] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, *IEEE Transactions on Evolutionary Computation* 3 (1999) 257–271.
- [61] H. W. Lilliefors, On the kolmogorov-smirnov test for normality with mean and variance unknown, *Journal of the American Statistical Association* 62 (1967) 399–402.
- [62] S. S. Shapiro, M. B. Wilk, An analysis of variance test for normality (complete samples), *Biometrika* 52 (1965) 591–611.
- [63] H. B. Mann, D. R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, *Annals of Mathematical Statistics* 1 (1947) 50–60.