



**HAL**  
open science

# An efficient local improvement operator for the multi-objective wireless sensor network deployment problem

Guillermo Molina, Francisco Luna, Antonio Jesús Nebro, Enrique Alba

► **To cite this version:**

Guillermo Molina, Francisco Luna, Antonio Jesús Nebro, Enrique Alba. An efficient local improvement operator for the multi-objective wireless sensor network deployment problem. *Engineering Optimization*, 2011, 10.1080/0305215X.2010.546840 . hal-00683843

**HAL Id: hal-00683843**

**<https://hal.science/hal-00683843>**

Submitted on 30 Mar 2012

**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.



**An efficient local improvement operator for the multi-objective wireless sensor network deployment problem**

Journal:	<i>Engineering Optimization</i>
Manuscript ID:	GENO-2010-0078.R2
Manuscript Type:	Original Article
Date Submitted by the Author:	29-Nov-2010
Complete List of Authors:	Molina, Guillermo; University of Malaga, Lenguajes y Ciencias de la Computacion Luna, Francisco; University of Málaga, Computer Science Nebro, Antonio; University of Málaga, Computer Science Alba, Enrique; University of Malaga, Lenguajes y Ciencias de la Computacion
Keywords:	Sensor network design, Operators, Multi-objective optimization
<p>Note: The following files were submitted by the author for peer review, but cannot be converted to PDF. You must view these files (e.g. movies) online.</p> <p>Molina.et.al.WSN.R2.latex.sources.tar.gz</p>	

SCHOLARONE™  
Manuscripts

## RESEARCH ARTICLE

**An efficient local improvement operator for the multi-objective wireless sensor network deployment problem**

Guillermo Molina, Francisco Luna, Antonio J. Nebro and Enrique Alba

Departamento de Lenguajes y Ciencias de la Computación

Universidad de Málaga, Málaga, Spain

email: {guillermo,flv,antonio,eat}@lcc.uma.es

(Received: )

Wireless sensor network layout, also known as sensor node deployment, is a complex NP-complete optimization task that determines most of the functioning features of the wireless sensor network. Coverage, connectivity and lifetime (handled through its opposing parameter, power consumption), are three of the most important characteristics of the service, and are taken into consideration in this work within a multi-objective approach of the problem. Leveraging on the specific properties of the wireless sensor nodes and networks, the Proximity Avoidance Coverage-preserving Operator (PACO) for local improvement is presented, described and tested. The testbed consists of a set of state-of-the-art multi-objective optimization algorithms with different configurations, and problem instances of varying size. In all the scenarios, and more specially in the algorithmic settings that already produce high performance solutions, PACO has proven to be a robust enhancement to the raw optimization technique without requiring additional computation, that easily scales through problem complexity.

**Keywords:** sensor network design, operators, multi-objective optimization

## Nomenclature

HECN	High Energy Communication Node
HV	Hipervolume quality indicator
MOCcell	Multi-Objective Cellular genetic algorithm
NSGA-II	Non Sorting Genetic Algorithm II
PACO	Proximity Avoidance Coverage-preserving Operator
PAES	Pareto Archived Evolution Strategy
$R_{COMM}$	Communication radius of a sensor node
$R_{SENS}$	Sensing radius of a sensor node
RGX	Rectangular Geographic Crossover
SBX	Simulated Binary Crossover
SPEA2	Strenght Pareto Evolutionary Algorithm 2
TTFF	Time To First Failure criterion
WSN	Wireles Sensor Network
WSNL	Wireless Sensor Network Layout problem

## 1. Introduction

Wireless Sensor Networks (WSNs) have become a hot topic in research (Akyildiz *et al.* (2002), Culler *et al.* (2004)). Their capabilities for monitoring large areas, accessing remote places, reacting in real-time, together with their relative ease of use have brought scientists a whole new horizon of possibilities. WSNs have so far been employed in many applications (Dargie and Poellabauer (2010)): militar activities such as reconnaissance, surveillance and target acquisition, environmental activities such as forest fire prevention, geophysical activities such as volcano activity study, biomedical purposes such as health data monitoring or artificial retina, or civil engineering such as structural health measurement. Their uses increase by the day and their potential applications seem boundless. The wide variety of applications results in a wide variety of networks bearing different constraints and having different features, yet most of them share some common issues that allow them to be treated homogenously.

One of the main features of WSNs is their geographical ubiquity, this makes the deployment of the nodes a critical task (Nan and Li (2008)). The coverage of the network, which depends directly on the positions of the nodes, is one such feature. For instance, in the countersniper system (Lédeczi *et al.* (2005)), the physical distribution of the nodes determine their capability to locate the shooter. In a forest fire prevention, the origin and evolution of the fire can also be known if the nodes are properly deployed (Mladineo and Knezic (2000)).

Another feature of the uttermost importance in WSNs, that also depends on the node deployment, is node energy consumption. In most scenarios, it is practically unfeasible to substitute nodes or recharge their energy: the high number of nodes or the hostility of the environment they are deployed in makes the task impossible. However, WSNs should work for the longest possible time. This causes energy saving to be one of the principal policies in a WSN in order to increase the network lifetime. The main source of energy consumption for WSNs is widely considered to be wireless communication (Ganesan *et al.* (2006), Li *et al.* (2006)), which depends on the communication structure of the network, which in turn depends on the node deployment. Therefore, an optimal layout of nodes involves considering several conflicting design objectives. In the adopted approach, these objectives are the network coverage, the lifetime, and the cost of the network (taken as the number of nodes). Thus, the problem at hand is a multi-objective optimization problem.

The WSN deployment or layout problem (WSNL problem for short), which was proven

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

to be NP-complete in (Wu *et al.* (2007)), is a complex task that has received much attention in the literature. The NP-completeness of the WSNL problem makes using metaheuristics (Blum and Roli (2003)) mandatory so as to deal with the increasingly-sized, real-world instances in affordable times. The point is that, even though metaheuristics have been used to some extent, the use of these optimization techniques is limited to almost canonical forms of the algorithms, with little-to-no adaptation to the problem particularities. Yet the use of specific problem knowledge is an important issue, and should not be overlooked.

Therefore, the main contributions of this work are as follows. First, a new local improvement operator, the “Proximity Avoidance Coverage-preserving Operator” (PACO), is presented here that takes advantage of specific problem knowledge to solve the WSNL problem. The operator is characterized and its parameters are tuned. Second, it is combined with different state-of-the-art multi-objective optimization techniques, and its effectivity and robustness are assessed. Finally, a scalability study is carried out on the problem instance size (size of the terrain, number of nodes), demonstrating not only that the operator scales well, but also that its efficiency increases for larger problem instances. This last effect is specially important, since the number of nodes of a WSN is expected to grow in the future.

The rest of the paper is structured as follows. The next section is devoted to providing the reader with a review of the related literature. The WSNL problem is formally depicted in Section 3. Then the PACO operator is presented and described in Section 4. The multi-objective optimization techniques, as well as the integration of PACO into them, are presented in Section 5. Then in Section 6, the results of the experimental evaluation of PACO are shown and discussed. Finally, the main conclusions are drawn in Section 7, where future lines of research are also sketched.

## 2. Related work

In its most basic form, the WSNL problem amounts to selecting the geographic locations for the deployment of each single node of the network. This problem is widely considered one of the fundamental tasks in WSNs (Nan and Li (2008)) and, as such, has been extensively studied in the literature. The point is that, in this research field, each author has used a formulation which is strongly scenario dependent and, as a consequence, there exist many papers that use different approaches to the problem, make different assumptions, set different optimization objectives, and use different models for the problem, the network, and the sensor nodes. A comprehensive review of the main models used in the existing literature for dealing with the problem is presented in (Molina (2010)). This section is therefore devoted to presenting the most popular approaches to solve the WSNL problem and their related works.

There are many works in the literature that tackle the WSNL problem. Interesting surveys on coverage problems defined for WSNs, that are mostly related to the defined WSNL problem can be found in (Cardei and Wu (2006), Younis and Akkaya (2008), Thai *et al.* (2008)). Specifically, in (Younis and Akkaya (2008)), the authors classify node placement problems into two categories: static and dynamic. This work belongs to the first category. As stated previously, different proposals have used different approaches, different models, different objectives and constraints, etc., what makes the related literature to be quite heterogeneous. The most popular approaches to the WSN node deployment problem are distinguished among:

- Assume that nodes follow a random deployment.
- Use a regular geometric deployment.
- Define the problem as a continuous optimization problem in which the location of the nodes to be deployed have to be selected.

Most of the early work on node deployment assume that nodes cannot be placed deterministically, but occupy random positions instead. This line of work usually follows one of two leads: in the first, the authors assume a given distribution function and get the resulting performance statistics from the network (usually, expected values, and upper/lower bounds). Examples of such approach are the works of Brass (2007), Cevher and M.Kaplan (2009), Lazos and Poovendran (2006), Manohar *et al.* (2009), or Shrivastava *et al.* (2009). In the second, the distribution function of the random node deployment can be optimized (for instance, a parametric function may be defined) so that the resulting network has the best possible performance statistics (Wettergren and Costa (2009)).

Regular or systematic node deployment strategies have also been researched, as they present the advantage of simplicity and scalability. Examples for this kind of deployment are the placement of the nodes according to a regular lattice, such as a square or hexagonal grid. The goal of these approaches is to reach efficient deployments in the sense of using the minimum number of nodes to provide full area coverage (Bai *et al.* (2006), Jain and Liang (2005), Kar and Banerjee (2003), Zheng *et al.* (2007)), but robustness (to be considered as looking for the maximum number of paths between two nodes) is also an issue (Biagioni and Sasaki (2003), Esseghir *et al.* (2005)).

However, this work is focused on non-systematic deterministic node placement, that is, the location of the nodes to be deployed have to be determined. This is the most general and interesting approach to the WSNL problem since random and regular deployments either do not address the problem (random deployments) or rely on a very simple problem model (regular deployments). So, when the problem lies in finding the location of the nodes, a very large body of research knowledge can be found because some forms of the optimization problem defined have been demonstrated to be NP-complete (Wu *et al.* (2007)). Also, the problem can be reduced to the set covering problem, by restricting the available positions of the sensor nodes to a set of discrete locations (for instance a regular point grid); and the set covering problem is well known to be a NP-complete problem (Cheng *et al.* (2005)). Therefore, the WSNL problem is NP-complete as well. This fact has made researchers to use heuristic algorithms for tackling large instances of the problem (exact or complete algorithms are discarded due to the time and/or memory required for them to find the optimal solution).

The heuristic algorithms applied to solve the WSNL problem can be mainly classified into two types. The first group includes works that use specific methods, often referred to as *ad-hoc heuristic* methods, tailored after the specifics of the problem instances at hand. A recurrent case is the use of greedy methods. In (Dhillon and Chakrabarty (2003)), a regular grid is used to compute the detection probability of a WSN and to place the nodes in order to obtain differentiated coverage. The authors propose two greedy strategies for the node deployment: the first one places a node at each step in the position that maximally reduces the accumulated probability of non-detection, and the second one places a node at each step in the position with minimal detection probability. Zhang and Wicker (2005) study the positioning of sensors in a terrain from the point of view of data transmission. They divide the terrain into cells, then analyze how  $N$  sensors should be distributed among the cells, in a way that avoids network bottlenecks and data loss. An *ad-hoc heuristic* algorithm is proposed for node distribution. The deployment of the nodes to reduce the distortion and the energy consumption (due to transmissions) is

1  
2  
3  
4  
5 studied in (Ganesan *et al.* (2006)). Two codification systems for the data, joint-entropy  
6 and Slepian-Wolf, are considered. An ad-hoc heuristic solution based on concentric circles  
7 is proposed. A sensor placement for perimeter coverage is presented in (Jourdan and Roy  
8 (2008)), with the purpose of detecting a moving agent. The field is assumed convex, and  
9 the moving agent has to be detected as it enters or leaves the field. The Position Error  
10 Bound (PEB) is obtained, and a greedy method that minimizes the PEB is proposed.  
11 An estimation of the detection of moving targets by a WSN is given in (Lazos *et al.*  
12 (2009)), along with a node deployment strategy. Based on the analogy with the line set  
13 intersection problem, the detection probability is obtained for a single node, and it is  
14 found to depend only on the perimeter of its coverage. The proposed deployment strategy  
15 seeks to maximize the internode distance so as to minimize the overlap between coverage  
16 cells; it achieves so by solving the circle packing problem. A set of base stations for node  
17 location purposes has to be selected from a pool of deployed nodes in (Paschalidis and  
18 Guo (2009)). The basic idea is to divide the network in as many regions as possible,  
19 where for every region pair there is one base station that can discriminate with low error  
20 probability using the received signal from the new node. Lifetime is also the main concern  
21 in (Chen *et al.* (2005)), but instead of raw lifetime, they study the lifetime per node, that  
22 is, the ratio between the network's lifetime and the number of nodes in the network. The  
23 authors propose a greedy algorithm for node placement along the WSN axis, and derive  
24 the optimal number of nodes and their positions.  
25

26  
27 The second group includes the works that use general-purpose flexible optimization  
28 methods, namely metaheuristics (Blum and Roli (2003)). This body of research contains  
29 a high number of publications, among which the most relevant ones that tackle prob-  
30 lems resembling the WSNL problem have been selected. Jourdan and de Weck solved an  
31 instance of WSNL using a multi-objective genetic algorithm in (Jourdan and de Weck  
32 (2004)). In their formulation a fixed number of sensors has to be placed in order to  
33 maximize the coverage and the lifetime of the network. Dijkstra's algorithm is repeat-  
34 edly applied to determine the number of rounds that can be performed provided each  
35 node has a predefined starting energy. The NP-completeness of the WSNL problem with  
36 heterogeneous sensor nodes is demonstrated in (Wu *et al.* (2007)), by assimilating it the  
37 knapsack problem. The authors use a grid model of the terrain and propose a genetic  
38 algorithm (GA) to obtain the optimal deployment to maximize the average detection  
39 probability over the sensor field, with budget constraints on the number and types of  
40 nodes. Specific genetic crossover and mutation operators are proposed as well. The pro-  
41 posed GA outperforms two greedy algorithms which are based on an uniform placement  
42 of the nodes. A multi-objective GA is used in (Kang and Chen (2009)) to obtain 3D dif-  
43 ferentiated coverage by placing  $N$  sensors in a 3D field and selecting the sensing radius for  
44 the nodes. The coverage achieved has to be maximized, while the total energy consump-  
45 tion in the network has to be minimized. A similar problem definition, the differentiated  
46 coverage in 2D, is solved in (Aitsaadi *et al.* (2008)) with a Tabu Search (TS). Instead  
47 of reducing the consumed energy, the number of nodes placed has to be minimized. The  
48 proposed TS is shown to outperform several greedy algorithms. A GA to deploy sensors  
49 on a planar grid with obstacles and differentiated coverage is proposed in (Xu and Yao  
50 (2006)). The results have pointed out that the proposed GA has reached more accurate  
51 solutions than previously proposed heuristic algorithms. A multi-objective approach to  
52 the WSN layout, where the coverage and lifetime are the opposing objectives, and the  
53 number of nodes is fixed, is adopted in (Pradhan *et al.* (2009)); a multi-objective parti-  
54 cle swarm optimization algorithm (MOPSO) is used to solve this problem. The authors  
55 of Woehrle *et al.* (2010) used a multi-objective evolutionary algorithm (MOEA), named  
56  
57  
58  
59  
60

1  
2  
3  
4  
5 IBEA, to solve a multi-objective sensor placement problem where the optimization ob-  
6 jectives are the cost (measured by the number of sensor nodes) and the transmission  
7 reliability (measured by the expected transmission failure rate). The authors employ  
8 a geographic crossover operator, and two types of mutation: Voronoi-based and Gaus-  
9 sian. The deployment and power assignment problem is solved using a multi-objective  
10 evolutionary algorithm called MOEA/D in (Konstantinidis *et al.* (2010)). The authors  
11 propose a decomposition of the problem into several scalar problems in which the ob-  
12 jectives, coverage and lifetime, are merged with different weights, and reconstruct the  
13 Pareto set from the solutions to the different problems. Specific genetic operators are  
14 proposed that operate in a different manner depending on the current objective weight-  
15 ing, to guide the search process towards the specific region of interest. The technique is  
16 shown to outperform the well-known NSGA-II algorithm.

17  
18 To the best of our knowledge, the next work describes the only algorithm which is en-  
19 dowed with a improvement operator that uses problem specific knowledge for the WSNL  
20 problem (all the algorithms described above are mostly used in their canonical forms).  
21 This work is presented in (Ferentinos and Tsiligiridis (2010)), in which a GA-based  
22 memetic algorithm is proposed to solve the dynamic design of WSNs. In this problem  
23 formulation, the WSN, which operates by rounds, consists of regular grid-deployed nodes;  
24 for each round, every node must be assigned one state out of four possibilities: cluster  
25 head, high energy operation, low energy operation, and non active. A set of objectives  
26 including active node density, energy consumption, and connectivity, are aggregated into  
27 a single weighted fitness function, and a mono-objective approach is adopted. An initial  
28 GA solution method is improved by adding a local search process that operates on a  
29 threshold basis: at each round, every node state has a corresponding remaining battery  
30 threshold; nodes that do not surpass the threshold cannot be in the corresponding state.  
31 This problem is a bit different from the one addressed in this work (regular deployment  
32 of the nodes), but it has been included because of the use of a local search operator.

33  
34 In summary, the first type of techniques, i.e., ad-hoc heuristics, regroups specific tech-  
35 niques to solve a particular type of WSNL problem. This group includes, among others,  
36 several greedy-like techniques; these techniques are very scenario-specific and thus hard  
37 to extrapolate to a different scenario, but leverage on problem knowledge and show  
38 high performances. The second type contains high-level optimization techniques, i.e.,  
39 metaheuristic algorithms. These techniques are robust and versatile and can be used to  
40 solve a wide range of –arbitrarily large– problem instances. The NP-completeness of the  
41 WSNL problem makes these algorithms to become the most appropriate choice. In all  
42 the works in which metaheuristics have been compared to ad-hoc heuristics, the results  
43 have shown that the formers usually outperform the latter, thus indicating the suitability  
44 of metaheuristics to address the WSNL problem. However, the proposed metaheuristic  
45 approaches also lack deep knowledge of the problem features that could help enhancing  
46 their performance. Indeed, the use of problem-specific knowledge is restricted to just  
47 the use of special genetic operators or different fitness functions in some works, and a  
48 particular single-objective local search operator in only one single contribution (which  
49 deals with an optimization problem that is slightly different from the WSNL problem  
50 addressed in this work). The contribution of this paper is therefore to propose a com-  
51 bined use of versatile metaheuristic solvers with a problem-specific heuristic to enhance  
52 their performances. To the best of our knowledge, the Proximity Avoidance Coverage-  
53 preserving Operator (PACO) is the first attempt at presenting such a problem-specific  
54 heuristic that is targeted to multi-objective metaheuristics.



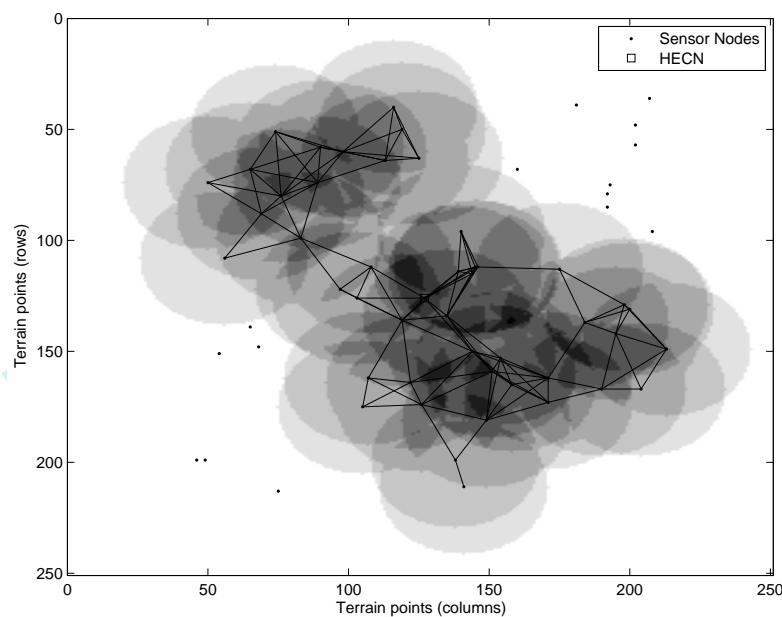


Figure 1.: Example WSN.

### 3. The wireless sensor network design problem

This section details the formulation of WSNL problem addressed in this work. A WSN is a wireless network composed of sensor nodes which sense or monitorize an area around itself called its sensing area. A parameter called *sensing radius* ( $R_{SENS}$ ) determines the sensitivity range of the sensor node and thus the sensing area. The nodes communicate among themselves using wireless communication links. These links are determined by the parameter *communication radius* ( $R_{COMM}$ ), the maximum distance at which two nodes can establish a link. A special node in the WSN, called High Energy Communication Node (HECN), is the gateway for external access to the network. The administrator of the network gathers the measured data and sends commands through it. Therefore every sensor node in the network must have communication with the HECN. An example WSN graphical representation is shown in Figure 1. In this illustration the HECN is located at the centre of the terrain, and the nodes are represented using dots. The network topology (communication links between nodes) is represented by lines and the covered terrain is shown in grey. In this topology, nodes may be connected to the HECN, or to all nodes within communication range that are 1 hop closer to the HECN, and also to all nodes within communication range that are 1 hop farther from the HECN. For convenience, when two nodes are connected, the closest to the HECN is referred to as the “parent”, and the farthest from the HECN is referred to as the “child”.

The definition of the WSNL problem used here has adopted the following models for the different WSN elements (Molina (2010)): binary coverage at node level, area coverage for the sensor network with a discrete grid model used for the terrain (each point in the grid represents one square meter of the terrain), and N-to-1 communications over a flat network structure. A simple routing algorithm is considered: every node sends its (re)transmitted information packets to the HECN itself if it is within communication

range, or distributes them among all neighbors that are closer (in hop count) to the HECN. When there are several neighbors closer to the HECN, each of them receives a traffic share proportional to the inverse of the link power (see Eq. 2). Every node has a traffic (number of packets to send) equal to the packets received from nodes farther from the HECN, and additionally produces one data packet per round (corresponding to its own sensed data, see Eq. 3). Formally:

$$Sent(x_i, x_j) = Traffic(x_i) \cdot ProbSend(x_i, x_j) \quad (1)$$

$$ProbSend(x_i, x_j) = \frac{\frac{1}{LinkPower(x_i, x_j)}}{\sum_{x_k} \frac{1}{LinkPower(x_i, x_k)}} \quad (2)$$

$$Traffic(x_i) = 1 + \sum_{x_j} Sent(x_j, x_i) \quad (3)$$

where  $LinkPower(x_i, x_j)$  is defined by the wave propagation model detailed below (Eq. 8).

With these models, the objectives of the WSNL problem are to obtain a full coverage network (set as a constraint) with minimum cost and maximum lifetime. The lifetime of a WSN is the period of time during which the network functions properly. As time passes, nodes will eventually run out of energy and stop operating, which results in a degradation of the network performance. The exact moment when the WSN stops functioning properly is subjective, but a broadly used measure for it is the time until the first node fails (*Time To First Failure* or TTF, Singh *et al.* (1998)). Formally, let  $\vec{x}$  be a non-fixed length vector of nodes  $x_i$  where each node is a 2D coordinate representing the node location, then the WSNL problem is defined as:

$$f_1(\vec{x}) = Length(\vec{x}) \quad (4)$$

$$f_2(\vec{x}) = Max \left( \{EnergyConsumed(x_i)\}_{i=1}^{f_1(\vec{x})} \right) \quad (5)$$

subject to

$$C(\vec{x}) = 100, \quad (6)$$

where the coverage function,  $C(\vec{x})$ , is defined as

$$C(\vec{x}) = 100 \cdot \left( \frac{CoveredPoints(\vec{x})}{TotalGridPoints} \right), \quad (7)$$

and  $CoveredPoints(\vec{x})$  is the function that, for any given solution  $\vec{x}$ , returns the number of grid points covered by some node  $x_i$  in  $\vec{x}$ . The wave propagation model defines the function  $EnergyConsumed(x_i)$  (Eq. 9).

That is, the number of sensor nodes and their locations have to be chosen in a way that minimizes the cost of the network which, in this case, is calculated as the number of deployed sensor nodes ( $f_1$ ), and the energy spent in communications by the most loaded node in the network ( $f_2$ ). The load in the most loaded node of the network is minimized since this node constitutes the bottleneck of the network with respect to the network lifetime; the most loaded node will be the first node to run out of energy,

1  
2  
3  
4  
5 hence determining the network lifetime according to the TTFF criterion. The two  
6 objectives are opposed, since the higher the number of nodes, the lower the share of  
7 retransmissions. The WSNL problem definition also includes a constraint (Eq. 6) so that  
8 any feasible solution has to provide full sensing coverage.

9 In order to determine the energy spent in communications by any node of the WSN,  
10 the number of transmissions performed is calculated. The WSN operates by rounds: in  
11 a round every node collects the data from its measurements and sends it to the HECN  
12 encapsulated in a packet; between rounds the nodes are in a low-energy state. It is  
13 assumed that the main source of energy consumption is packet transmission; besides,  
14 packet (re)transmission is the sole energy-consuming process of the WSN that is di-  
15 rectly affected by node deployment (and its resulting topology), and thus susceptible  
16 of being optimized in order to extend network lifetime. Therefore, all sources of energy  
17 consumption are neglected besides packet transmissions in this work.

18 To calculate the energy spent by transmissions the simple wave propagation model  
19 shown in Eq. 8 is applied for the power required per data packet to be transmitted over  
20 from node  $x_i$  to node  $x_j$ . Assuming free-space path loss sets  $\alpha = 2$ . Since the  $\beta$  constant  
21 value does not affect the optimization problem results, it will be neglected. The total  
22 energy consumed by a node  $x_i$  is shown in Eq. 9, where  $\beta = 1$  and  $\alpha = 2$ . The function  
23  $Sent(a, b)$  indicates the number of data packets sent from node  $a$  to node  $b$  (see Eq. 1).  
24  
25

$$26 \quad LinkPower(x_i, x_j) = \beta \cdot \|x_i - x_j\|^\alpha \quad (8)$$

$$27 \quad EnergyConsumed(x_i) = \sum_{x_j \in neighbors(x_i)} Sent(x_i, x_j) \cdot \|x_i - x_j\|^2 \quad (9)$$

28  
29  
30  
31 These models and problem objectives have been chosen because of two main facts.  
32 On the one hand, they have been widely used in the literature and represent a rather  
33 general approach to the problem, that is, they avoid strong problem-specificness that  
34 would hinder this work from drawing useful and relevant conclusions about using PACO.  
35 On the other hand, defining an improvement operator for the more general case will  
36 surely make easier its adaptation to the specific scenarios that may appear in this field.  
37  
38

#### 39 4. The PACO operator

40  
41 This paper presents a new operator for local improvement in a WSN conceived to be in-  
42 tegrated into an optimization algorithm: the “Proximity Avoidance Coverage-preserving  
43 Operator” (PACO). The basis of its functioning is identifying locally suboptimal config-  
44 urations and trying to fix them.  
45

##### 46 4.1. Operator description

47  
48 It is understandable that, for the purpose of an efficient WSN deployment, having nodes  
49 too close to one another produces inefficiency due to two reasons:

- 50 • An extra node is deployed (increased cost) that provides little-to-no coverage improve-  
51 ment (since most of its sensing area is already covered by the other node).
- 52 • An extra information packet (reduced energy efficiency) containing the extra node’s  
53 data has to be relayed.  
54  
55

56 Thus, the purpose of PACO is to replace the pair of nodes that are close to one another  
57 by a single node, provided that this single node can safely replace them:

- 58 • The node guarantees that the area covered by the two initial nodes is still covered.  
59  
60

- The connectivity of the WSN is maintained.

Thus PACO has to find an “equivalent deployment area” for the node pair, such that any node placed inside this area is capable of maintaining both the coverage and connectivity of the network after the pair has been removed. This area is found as the intersection of two zones: the “coverage preserving zone”, which is the area where a single node guarantees coverage, and the “connectivity preserving zone”, which is the area where a single node maintains the network connectivity.

It has to be pointed out that node position and covered area points are subject to a reciprocity property. If a sensor node covers a disk-shaped area around it, then any given terrain point can be covered by a sensor node placed anywhere inside that same disk-shaped area around it. This property shall be used to define a reciprocal WSN whose coverage will identify the coverage equivalent area. The same property holds for the connectivity.

The operation of PACO can be summed up in the following steps:

- (1) Choose a pair of close nodes. The PACO operator first explores the whole WSN in search for all pairs of close nodes; this can be considered as a preliminary step. A *threshold* parameter defines which pairs of nodes are considered to be *close*: all nodes  $n_a, n_b$  whose Euclidean distance is below it. This threshold value should typically be some fraction of  $R_{SENS}$ .
- (2) Obtain the “coverage preserving zone” for that pair. To do so, PACO identifies the area that is exclusively covered by the selected pair (note that the connectivity constraint is not taken into account here). A reciprocal WSN is then created with a node in every terrain point of this area, and the coverage of this reciprocal network is computed; the area that is covered by all the nodes in the reciprocal WSN is the “coverage preserving zone”. Thus, a single node placed in this zone can effectively replace the selected pair in terms of coverage.
- (3) Obtain the “connectivity preserving zone” for that pair. Regarding the connectivity, the node has to fulfill the following constraints:
  - All children nodes of the two nodes removed must be within communicating range of the placed node.
  - At least one of the parent nodes must be within communicating range of the placed node.To locate this “connectivity preserving zone” the same principle as before is applied: each child and each parent defines a disk-shaped connectivity zone around itself (with radius  $R_{COMM}$ ). The intersection or overlap zone (if any) of all the zones defined by the children guarantees that a single node will keep all the children connected. The union of all the zones defined by the parent nodes guarantees that at least one parent one is connected. The final “connectivity preserving zone” is the intersection of the children and parent zones.
- (4) Obtain the “equivalent deployment area” as the intersection of both the coverage and connectivity preserving zones.
- (5) If the “equivalent deployment area” is empty, i.e., no overlap is found between the two previous zones, the two removed nodes must be restored and the operator does nothing. Otherwise, when there is an overlap zone (non-empty “equivalent deployment area”), then a single node is placed inside it that effectively replaces the two initially chosen nodes.

The general PACO procedure is an iterative procedure (Algorithm 1). The steps above are performed for each pair of close nodes found in the WSN.

**Algorithm 1** Pseudocode for PACO.

---

```

1: input: a WSN layout  $wsn = n_1n_2 \dots n_k$ ,  $n_i \in WSN$ , a threshold value  $th$ 
2:  $wsnBackup \leftarrow wsn$  // Store a copy of the current layout
3:  $stop \leftarrow false$ 
4: for All  $(n_a, n_b) \leftarrow \text{NodePair}(wsn)$  do
5:   if  $\text{NearbyNodes}(n_a, n_b, th)$  then
6:      $CovEq \leftarrow \text{ComputeCovEq}(wsn, n_a, n_b)$  // Step 1
7:      $ConEq \leftarrow \text{ComputeConEq}(wsn, n_a, n_b)$  // Step 2
8:      $CovConEq \leftarrow CovEq \cap ConEq$  // Step 3
9:     if  $CovConEq \neq \emptyset$  then
10:       $n_p \leftarrow \text{ChooseNode}(wsn, CovConEq)$ 
11:       $wsn \leftarrow \text{Remove}(wsn, n_a, n_b)$ 
12:       $wsn \leftarrow \text{Deploy}(wsn, n_p)$ 
13:       $\text{Evaluate}(wsn)$ 
14:    end if
15:  end if
16: if  $\text{NodesDeployed}(wsn) < \text{NodesDeployed}(wsnBackup)$  &
17:    $\text{EnergyConsumption}(wsn) < \text{EnergyConsumption}(wsnBackup)$  then
18:    $wsnBackup \leftarrow wsn$  //  $wsn$  dominates  $wsnBackup$ 
19: else
20:    $wsn \leftarrow wsnBackup$  // restore the assignment
21: end if
22: end for
23: return  $wsnBackup$ 
24: output: a possibly improved WSN layout

```

---

**4.2. PACO formal specification**

A formal description of PACO's operation is as follows. Let  $T$  be the set of terrain points  $p$  (the discretized terrain grid), and let  $WSN$  be the points where a sensor node is deployed ( $WSN \subseteq T$ ). Assume the functions  $coverage()$  that for each node  $n \in WSN$  returns the set of points in  $T$  covered by that node,  $parentNodes()$  that for each node  $n \in WSN$  returns the set of nodes in  $WSN$  that are parent nodes of  $n$ , and  $childNodes()$  that for each node  $n \in WSN$  returns the set of nodes in  $WSN$  that are children nodes of  $n$ . Select a pair of nodes  $n_a$  and  $n_b$  such that  $n_a, n_b \in WSN$  and  $\|n_a - n_b\| < threshold$ .

- Step 1 Define  $E$  as the set of points covered only by  $\{n_a, n_b\}$ , i.e.  $p \in E \equiv p \in coverage(n_a) \cup coverage(n_b)$ ;  $\forall n \in WSN, n \neq n_a, n_b, p \notin \cup coverage(n)$ . Find the set of points  $CovEq$  that guarantee coverage to the set  $E$ :  $n \in CovEq \equiv \forall p \in T : p \in E \rightarrow p \in coverage(n)$ .
- Step 2 Define the sets  $P$  and  $C$  such that:  $P = parentNodes(n_a) \cup parentNodes(n_b)$  and  $C = childNodes(n_a) \cup childNodes(n_b)$ . Then find the set  $ConEq$  that maintains the connectivity of the network:  $n \in ConEq \equiv \forall n_c \in WSN : n_c \in C \rightarrow n_c \in childNodes(n), \exists n_p \in P : n_p \in parentNodes(n)$ .
- Step 3 Define  $CovConEq$  as the set of points that guarantee both coverage and connectivity:  $CovConEq = CovEq \cap ConEq$ .

Then as long as  $CovConEq \neq \emptyset$ , a single sensor placed in any  $n \in CovConEq$  may replace the pair  $\{n_a, n_b\}$  without loss of coverage or connectivity.

## 5. Algorithms

This section provides the reader with a general background on multi-objective optimization required to later describe the multi-objective algorithms used in the experimental section. The solution encoding and the genetic operators used by the algorithms are presented next. Finally, the integration of the PACO operator into these algorithms is detailed.

### 5.1. Multi-objective background concepts

A general multi-objective optimization problem (MOP) can be formally defined as follows (assuming minimization without loss of generality):

**DEFINITION 5.1 MOP** Find a vector  $\vec{x}^* = (x_1^*, x_2^*, \dots, x_n^*)$  which satisfies the  $m$  inequality constraints  $g_i(\vec{x}) \geq 0, i = 1, 2, \dots, m$ , the  $p$  equality constraints  $h_i(\vec{x}) = 0, i = 1, 2, \dots, p$ , and minimizes the vector function  $\vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x}))$ , where  $\vec{x} = (x_1, x_2, \dots, x_n)$  is the vector of decision variables.

The set of all values satisfying the constraints defines the *feasible region*  $\Omega$  and any point  $\vec{x} \in \Omega$  is a *feasible solution*.

As mentioned before, the *Pareto Optimal Set* is sought. Before its definition some concepts must be introduced.

**DEFINITION 5.2 Pareto Optimality** A point  $\vec{x}^* \in \Omega$  is *Pareto optimal* if for every  $\vec{x} \in \Omega$  and  $I = \{1, 2, \dots, k\}$  either  $\forall_{i \in I} f_i(\vec{x}) = f_i(\vec{x}^*)$  or there is at least one  $i \in I$  such that  $f_i(\vec{x}) > f_i(\vec{x}^*)$ .

This definition states that  $\vec{x}^*$  is Pareto optimal if no feasible vector  $\vec{x}$  exists which would improve some criterion without causing a simultaneous worsening in at least one other criterion. Other important definitions associated with Pareto optimality are the following:

**DEFINITION 5.3 Pareto Dominance** A vector  $\vec{u} = (u_1, \dots, u_k)$  is said to *dominate*  $\vec{v} = (v_1, \dots, v_k)$  (denoted by  $\vec{u} \preceq \vec{v}$ ) if and only if  $\vec{u}$  is partially smaller than  $\vec{v}$ , i.e.,  $\forall i \in I, u_i \leq v_i \wedge \exists i \in I : u_i < v_i$ .

**DEFINITION 5.4 Pareto Optimal Set** For a given MOP  $\vec{f}(\vec{x})$ , the *Pareto optimal set* is defined as  $\mathcal{P}^* = \{\vec{x} \in \Omega | \neg \exists \vec{x}' \in \Omega, \vec{f}(\vec{x}') \preceq \vec{f}(\vec{x})\}$ .

**DEFINITION 5.5 Pareto Front** For a given MOP  $\vec{f}(\vec{x})$  and its Pareto optimal set  $\mathcal{P}^*$ , the *Pareto front* is defined as  $\mathcal{PF}^* = \{\vec{f}(\vec{x}) | \vec{x} \in \mathcal{P}^*\}$ .

### 5.2. Algorithm description

Four multi-objective evolutionary algorithms (EAs, Coello Coello *et al.* (2007)) are used in this study, namely NSGA-II, SPEA2, PAES, and MOCeII. The first two algorithms are the two most widely used ones in the literature, while PAES is a simple trajectory-based algorithm, and MOCeII is a fairly new proposal that achieves state-of-the-art performances for some problems. The implementation of these algorithms provided by jMetal (Durillo *et al.* (2010)), an object-oriented Java-based framework aimed at the development, experimentation, and study of metaheuristics for solving multi-objective

---

**Algorithm 2** Pseudocode for a generic multi-objective EA.

---

```
1:  $P(0) \leftarrow \text{GenerateInitialPopulation}()$ 
2:  $\text{EvaluateObjectives}(P(0))$ 
3:  $PF \leftarrow \text{CreateParetoFront}()$  //Create an empty front
4:  $t \leftarrow 0$ 
5: while not  $\text{Termination\_Condition}()$  do
6:   parents  $\leftarrow \text{Selection}(P(t))$ ;
7:   offspring  $\leftarrow \text{EvolutionaryOperators}(\text{parents})$ ;
8:    $\text{Improvement}(\text{offspring})$ ; /* PACO goes here */
9:    $\text{EvaluateObjectives}(\text{offspring})$ ;
10:   $P(t+1) \leftarrow \text{UpdatePopulation}(P(t), \text{offspring})$ 
11:   $\text{UpdateFront}(PF, P(t+1))$ 
12:   $t \leftarrow t+1$ 
13: end while
```

---

optimization problems<sup>1</sup>, is used in this work.

Starting from the pseudocode of a generic multi-objective EA included in Algorithm 2, the main features of the algorithms used in this work are outlined. For a detailed description, interested readers are referred to the references provided for each solver.

Both NSGA-II (Deb *et al.* (2002)) and SPEA2 (Zitzler *et al.* (2002)) use the scheme of Algorithm 2. They differ one each other in the mechanism used to keep a diverse approximated Pareto front. PAES (Knowles and Corne (2000)) in turn has a population with one single solution that it is iteratively modified by using a mutation operator only (no crossover is required). MOCcell (Nebro *et al.* (2009)) is a structured (cellular) EA, where each solution has a neighborhood of solutions inside with which it can cross. Though none of these algorithms includes an improvement operator (line 8 of Algorithm 2) in their canonical definition, the position where PACO comes in has been indicated in the pseudocode nonetheless.

In order to deal with constrained optimization problems such as the WSNL problem, all the algorithms have used the constraint domination principle presented in (Deb (2001)). It is based on considering feasible solutions as better solutions than non-feasible ones. Among non-feasible solutions, those with a smaller overall constraint violation are better (constraints are normalized to be greater than or equal to zero).

### 5.3. Solution encoding

According to the definition of the problem provided in Section 3, the following solution encoding is adopted: a WSN is represented by an array of sensor nodes. Each sensor node in this array has a two-level definition: a single bit determines whether the node is deployed or not (first layer), then two coordinates determine the location of the node in the terrain should it be deployed (second layer). This coding scheme uses a fixed-length variable (the array) to represent a non fixed-sized solution (the deployed WSN); this means only those nodes that are selected for deployment (in the first layer) should compute for the network features and cost, and that the coordinate values of the non-deployed nodes have no significance whatsoever. The length of the array should be large enough to enable the

---

<sup>1</sup>jMetal is freely available for download at the following URL:  
<http://jmetal.sourceforge.net/>.

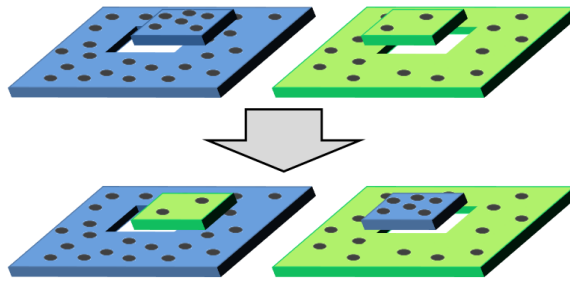


Figure 2.: Example rectangular geographical crossover. All nodes in the extracted rectangles are exchanged between solutions.

handle of large WSNs: in this article the value will be  $4 \times TerrainArea / (R_{SENS} \cdot R_{COMM})$  approximately.

#### 5.4. Genetic operators

This section presents the different crossover and mutation operators used to evaluate the suitability of PACO under different operating conditions.

##### 5.4.1. Crossover operators

Two crossover operators are used: SBX (Deb and Agrawal (1995)) crossover and a geographic crossover (Wu *et al.* (2007)). Whereas the former is the most widely applied operator in the evolutionary multi-objective community, the latter is engineered to capture the particularities of the WSNL problem. In a crossover, two solutions called parents produce one or more new solutions called offsprings by exchanging information with some probability (the *crossover probability*,  $p_c$ ).

The main issue when adapting the SBX crossover to the solution encoding presented in Section 5.3 concerns the management of deployed vs. non-deployed sensors. Let  $p_1$  and  $p_2$  be the individuals to be crossed and let  $s_i^{p_1}$  and  $s_i^{p_2}$  be the sensors at position  $i$  of each individual, at which SBX is operating. Let  $o_1$  and  $o_2$  also be the two generated offsprings and  $s_i^{o_1}$  and  $s_i^{o_2}$  be the corresponding sensors at the same position ( $i$ ). The following cases may arise:

- Neither  $s_i^{p_1}$  nor  $s_i^{p_2}$  is deployed: neither  $s_i^{o_1}$  nor  $s_i^{o_2}$  are deployed either.
- Either  $s_i^{p_1}$  or  $s_i^{p_2}$  is deployed, but not both: the deployed sensor in the parent ( $s_i^{p_1}$  or  $s_i^{p_2}$ ) is independently copied to each offspring with a chance of 50%.
- Both  $s_i^{p_1}$  and  $s_i^{p_2}$  are deployed: the coordinates of  $s_i^{o_1}$  and  $s_i^{o_2}$  are computed by using the coordinates of  $s_i^{p_1}$  and  $s_i^{p_2}$  and the standard SBX operations. The distribution index is set to  $\eta_c = 20$ , a widely used value in the literature.

The other crossover operator used is the geographic crossover, called RGX (Rectangular Geographic Crossover, Wu *et al.* (2007)). In it, nodes are exchanged among solutions based on their geographic locations. A rectangular-shaped area is defined, and all nodes belonging to that area are exchanged between the two solutions (see Figure 2).



#### 5.4.2. Mutation operators

Two different mutation operators have been used: a fully random mutation and a geographic mutation which is based on the polynomial mutation defined in (Deb and Agrawal (1995)). Both mutation operators modify each potential node (which can be either deployed or not) of a given solution with some probability (the *mutation probability*,  $p_m$ ); different nodes are affected by the mutation independently. When a node is chosen to be modified, the performed procedure differs, depending on the mutation operator that is used. Both first check whether the node is deployed or not. If not, it is placed in a random location. Otherwise, it is either removed or repositioned with equal probability:

- Random mutation: The node is moved to any terrain point with uniformly distributed probability.
- Geographic mutation: The node is moved to a point in the surrounding area of the node's current position. This bounded movement is computed by using the polynomial mutation operator separately on the two coordinates of the node.

#### 5.5. PACO integration

The approach used to include PACO in the multi-objective algorithms frame is straightforward, as can be seen in the Algorithm 2. Whenever a new solution is produced by the evolutionary operators (line 7), PACO may be applied onto it (line 8). An elitist criterion is applied: the solution produced by PACO is kept if and only if both the number of deployed sensors and the energy consumption are reduced, i.e., the new solution is said to dominate the older one (see Section 5.1). Otherwise it is rejected and the previous one is kept.

Another important remark has to be done here. Note that each replacement operation by PACO consumes one function evaluation. As explained in Section 4, the function evaluations consumed by PACO are properly accounted for into the computational effort, to ensure fairness in the comparisons between configurations using PACO and not using PACO.

### 6. Experimental study

A set of experiments are conducted to assess the performance of PACO on different scenarios. The base problem instance is defined with the following properties:

- Square terrain  $250 \times 250 m^2$
- Maximum number of nodes 250
- Initial node deployment probability 50%, uniform distribution
- Sensor node features:  $R_{SENS} = 30 m$ ,  $R_{COMM} = 30 m$

First, the methodology and tools used to analyze the experimental results are described in Section 6.1. The initial tuning of PACO's own parameters is performed in Section 6.2. The robustness and performance of PACO are studied in Section 6.3, where the operator is applied on a wide set of algorithmic configurations (with varying algorithm and genetic operator configuration). As an additional contribution, high-performing algorithmic configurations will be detected. Finally, the scalability is tested by using PACO-equipped algorithms on problem instances of growing sizes in Section 6.4.

## 6.1. Experimental methodology

Comparing different multi-objective sets is not a trivial issue, since the tools from the mono-objective domain (mean and standard deviation) cannot be extended. Therefore, two specific approaches followed in this work are described here: the hypervolume indicator, HV (Zitzler and Thiele (1999)) (Section 6.1.1) and the attainment surfaces (Knowles (2005)) (section 6.1.2). When sets of scalar values are compared, a statistical analysis is performed (Section 6.1.3).

### 6.1.1. Hypervolume indicator

HV calculates the (hyper)volume (in the objective space of solutions) covered by members of a nondominated set of solutions  $Q$  for problems where all objectives are to be optimized in the same direction (either minimized or maximized). Since the problem is a minimization one, the minimization version of HV is described. For each solution  $i \in Q$ , a hypercube  $v_i$  is constructed with the solution  $i$  and a reference point  $W$  as the diagonal corners. This reference point is common to all the hypercubes, and is generated here as a vector containing the worst objective function values per objective found in the global pool of non-dominated solutions of each problem instance. Only those points that dominate the reference point are computed for the HV. Thereafter, the union of all hypercubes is computed and its hypervolume ( $HV$ ) is calculated as  $HV = volume \left( \bigcup_{i=1}^{|Q|} v_i \right)$ .

Finally, a normalization procedure is performed that translates the hypervolume values to the range  $[0.0, 1.0]$ . For this, the set of globally non-dominated solutions found among all executions is produced, this set is called the reference Pareto front. Let  $f_{max} = [f_1^{max}, f_2^{max}, \dots, f_k^{max}]$  and  $f_{min} = [f_1^{min}, f_2^{min}, \dots, f_k^{min}]$  be the vectors of maximum and minimum values for the  $k$  objectives in the reference Pareto front. Then every nondominated solution  $f = [f_1, f_2, \dots, f_k]$  is normalized, assuming a minimization problem, as follows:  $f_i^{norm} = \frac{f_i - f_i^{min}}{f_i^{max} - f_i^{min}}$  if  $f_i^{min} \leq f_i \leq f_i^{max}$ ,  $i = 1, \dots, k$ . Higher values of the hypervolume metrics are desirable.

### 6.1.2. Attainment function and surface

From the point of view of a decision maker, knowing the HV value gives little information, because it indicates nothing about the shape of the front. However, there is a need of knowing the general shape of the front, and thus of a way of representing the expected non-dominated front. For this, the concept of empirical attainment function is used (EAF, see (Knowles *et al.* (2006))). In short, the EAF is a function  $\alpha$  from the objective space  $\mathbb{R}^n$  to the interval  $[0, 1]$  that estimates for each  $n$ -dimensional point in the objective space the probability of being dominated by the Pareto front from a single run of the multi-objective algorithm. Given  $r$  approximate Pareto fronts obtained in that same number of different runs, the EAF is defined as:

$$\alpha(z) = \frac{1}{r} \sum_{i=1}^r I(A^i \preceq \{z\}) \quad (10)$$

where  $A^i$  is the  $i$ -th approximate Pareto optimal set and  $I$  is an indicator function that takes value 1 when  $A^i$  dominates solution  $z$ , and 0 otherwise. From the attainment function it is possible to define the concept of  $k\%$ -attainment surface (Knowles (2005)): the level curve with  $\alpha$  value  $k/100$ . Informally, the 50%-attainment surface in the multi-objective domain is analogous to the median in the mono-objective one.

The attainment surface provides the decision maker with a tool for quick evaluation

Table 1.: Results of different PACO configurations: HV. Median and IQR

Threshold	Prob.	NSGAII	SPEA2	PAES	MOCeII
–	0.00	0.548 <sub>0.067</sub>	0.518 <sub>0.066</sub>	0.535 <sub>0.083</sub>	0.506 <sub>0.074</sub>
5	0.01	0.536 <sub>0.029</sub>	0.500 <sub>0.050</sub>	0.529 <sub>0.073</sub>	0.500 <sub>0.045</sub>
	0.50	0.558 <sub>0.081</sub>	0.498 <sub>0.068</sub>	0.548 <sub>0.063</sub>	0.507 <sub>0.078</sub>
	1.00	0.543 <sub>0.053</sub>	0.519 <sub>0.046</sub>	0.537 <sub>0.058</sub>	0.510 <sub>0.073</sub>
15	0.01	0.560 <sub>0.056</sub>	0.522 <sub>0.074</sub>	0.525 <sub>0.034</sub>	0.505 <sub>0.048</sub>
	0.50	0.561 <sub>0.071</sub>	0.533 <sub>0.063</sub>	0.561 <sub>0.092</sub>	0.518 <sub>0.073</sub>
	1.00	0.556 <sub>0.051</sub>	0.516 <sub>0.043</sub>	0.587 <sub>0.073</sub>	0.517 <sub>0.065</sub>
30	0.01	0.548 <sub>0.038</sub>	0.540 <sub>0.046</sub>	0.518 <sub>0.060</sub>	0.501 <sub>0.063</sub>
	0.50	0.574 <sub>0.049</sub>	0.536 <sub>0.048</sub>	0.590 <sub>0.054</sub>	0.544 <sub>0.066</sub>
	1.00	0.567 <sub>0.052</sub>	0.557 <sub>0.053</sub>	0.591 <sub>0.086</sub>	0.553 <sub>0.047</sub>

of the variability of an algorithm. When the number of objectives of the MOP does not surpass three, the attainment surfaces can be represented graphically and constitute a helpful visual tool.

### 6.1.3. Statistical analysis

Since EAs are stochastic algorithms their results have to be given statistical significance. The following statistical procedure is used. First, 30 independent runs for test case (an algorithm with a given configuration used on a problem instance) are performed. The HV indicator and the attainment surfaces are then computed. In the case of HV, the following statistical analysis are carried out (Sheskin (2007)). First a Kolmogorov-Smirnov test is performed in order to check whether the samples are distributed according to a normal distribution or not. If so, an ANOVA I test is performed; otherwise a Kruskal-Wallis test is performed. Since more than two algorithms are involved in the study, a post-hoc testing phase which allows for multiple comparison of samples (multicompare) is been performed. Because of room constraints, the full details of the statistical analysis are not displayed in the paper. However, their results will be properly discussed when needed.

## 6.2. PACO general operating characteristics

The first set of experiments serves to outline the main features of the proposed optimization operator, PACO, as well as for tuning its internal parameters. For this, the different configurations of PACO are tested on the four algorithms with a standard configuration: NSGA-II, SPEA2, and MOCeII use a population size equal to 100, the genetic operators are SBX crossover and polynomial mutation with rates  $p_c = 0.9$  and  $p_m = 1.0/L$  (where  $L$  is the maximum number of sensors), respectively. Each run stops after completing 1,000,000 solution evaluations.

Within PACO, the threshold parameter takes the values 5, 15, and 30, while the application probability takes the values: 1%, 50% and 100%. Additionally, the results without PACO (application probability 0.0) are obtained as a test case to assess the operator's performance. The results are displayed in Table 1, under the format of median and interquartile range (IQR) of the HV indicator; for each algorithm, the best and second best configurations found are highlighted using dark and light grey background, respectively.

For all the algorithms but NSGA-II, the best configuration is found to be the one with the most intensive use of PACO: *threshold* = 30, and *probability* = 1.00. NSGA-II, however, reaches the best HV value with a very similar setting (*threshold* = 30, and *probability* = 0.50). What remains clear is that, in general, the higher the value of the threshold parameter, the higher the value of the HV. Similarly, the higher the

Table 2.: Influence of the initial conditions on PACO: HV. Median and IQR

X	NSGAI1		SPEA2		PAES		MOCcell	
	no PACO	PACO	no PACO	PACO	no PACO	PACO	no PACO	PACO
50%	0.548 <sub>0.067</sub>	0.567 <sub>0.052</sub>	0.518 <sub>0.066</sub>	0.557 <sub>0.053</sub>	0.535 <sub>0.083</sub>	0.591 <sub>0.086</sub>	0.506 <sub>0.074</sub>	0.553 <sub>0.047</sub>
75%	0.552 <sub>0.055</sub>	0.571 <sub>0.045</sub>	0.507 <sub>0.047</sub>	0.559 <sub>0.052</sub>	0.550 <sub>0.065</sub>	0.601 <sub>0.081</sub>	0.498 <sub>0.056</sub>	0.532 <sub>0.058</sub>
100%	0.543 <sub>0.044</sub>	0.582 <sub>0.051</sub>	0.497 <sub>0.027</sub>	0.542 <sub>0.045</sub>	0.536 <sub>0.082</sub>	0.607 <sub>0.055</sub>	0.501 <sub>0.045</sub>	0.556 <sub>0.041</sub>

application probability, the higher the HV (especially when the threshold parameter is high). As for the benefits of using PACO, the results are encouraging: for  $threshold \geq 15$  and  $probability \geq 50\%$ , most of the configurations with PACO (all but three) have outperformed the one without, in each of the four algorithms.

The next step is to test the sensibility of PACO towards the node density in the WSN. This issue is dealt with by modifying the initial node deployment probability  $X$ . Besides the predefined probability of  $X = 50\%$  for the problem the values  $X = 75\%$  and  $X = 100\%$  are also tested. The results of this experiment (HV median and IQR) are shown in Table 2. Again the results show clearly the benefits of using PACO: in the twelve scenarios consisting of combining algorithm and starting node density, the configuration using PACO outperforms the one without it. Furthermore the results obtained with PACO-equipped algorithms are fairly stable over the range of initial node densities, while the configurations without PACO always experience some performance degradations (SPEA2 specially), therefore demonstrating the robustness of the operator for varying node densities.

As a result, the ideal configuration of PACO is set to  $threshold = 30$ ,  $probability = 1.00$  for the remainder of this article; similarly, the initial node density is set to  $X = 50\%$ .

### 6.3. Performance of PACO with different genetic operators

The set of experiments in this section will test the effect of using PACO with different algorithms, genetic operators and parametric configurations, in order to assess the general robustness and performance of the operator. The genetic operators used are two crossover operators (SBX and RGX), and two mutation operators (random and polynomial mutation), previously described in sections 5.4.1 and 5.4.2, respectively. The parametric configurations are the different combinations of crossover probability with values  $p_c = 0.0$ ,  $p_c = 0.1$ ,  $p_c = 0.5$  and  $p_c = 0.9$ , and mutation probability such that on average 1 node, 5 nodes and 10 nodes are modified (for convenience referred to as  $p_m = 1.0$ ,  $p_m = 5.0$  and  $p_m = 10.0$ , respectively). Table 3 displays all the results obtained in this study; in this table, algorithms and parametric configurations are sorted by rows, while genetic operators and application/not of PACO, by columns. Again, the best results obtained (per line) are highlighted with grey background color.

The HV values displayed in Table 3 vary from 0.0 to 0.721 (they are normalized to unity). The configurations integrating PACO produce higher HV than the same configurations without PACO in 113 of 132 test configurations, that is improved efficiency with 85.61% probability. But some of these test configurations correspond to poor performing configurations, and their results are not extremely meaningful; if the comparison is restricted to the upper half (the best performing half) of the test configurations, then PACO yields improved performance with 98.48% probability. Hence PACO is a robust technique, and its performance improves for high performing configurations.

Regarding the mutation operator, it is clear that random mutation does not bring high performances and is largely outperformed by polynomial mutation: for the 132 test

Table 3.: Performance of PACO with different genetic operators: median and IQR of the HV indicator.

Crossover operator Mutation operator			SBX				RGX			
Algorithm	$p_m$	$p_c$	Polynomial		Random		Polynomial		Random	
			no PACO	PACO	no PACO	PACO	no PACO	PACO	no PACO	PACO
NSGAI	1.0	0.0	0.591 <sub>0.031</sub>	0.621 <sub>0.053</sub>	0.442 <sub>0.040</sub>	0.529 <sub>0.047</sub>	0.591 <sub>0.031</sub>	0.621 <sub>0.053</sub>	0.442 <sub>0.040</sub>	0.529 <sub>0.047</sub>
		0.1	0.592 <sub>0.041</sub>	0.631 <sub>0.027</sub>	0.459 <sub>0.052</sub>	0.543 <sub>0.040</sub>	0.651 <sub>0.047</sub>	0.686 <sub>0.060</sub>	0.508 <sub>0.083</sub>	0.619 <sub>0.066</sub>
		0.5	0.592 <sub>0.050</sub>	0.618 <sub>0.034</sub>	0.457 <sub>0.054</sub>	0.530 <sub>0.069</sub>	0.624 <sub>0.057</sub>	0.688 <sub>0.070</sub>	0.472 <sub>0.087</sub>	0.578 <sub>0.057</sub>
		0.9	0.548 <sub>0.067</sub>	0.569 <sub>0.049</sub>	0.434 <sub>0.032</sub>	0.515 <sub>0.065</sub>	0.615 <sub>0.044</sub>	0.660 <sub>0.054</sub>	0.475 <sub>0.061</sub>	0.574 <sub>0.052</sub>
	5.0	0.0	0.469 <sub>0.027</sub>	0.482 <sub>0.059</sub>	0.420 <sub>0.037</sub>	0.432 <sub>0.038</sub>	0.469 <sub>0.027</sub>	0.482 <sub>0.059</sub>	0.420 <sub>0.037</sub>	0.432 <sub>0.038</sub>
		0.1	0.466 <sub>0.043</sub>	0.486 <sub>0.034</sub>	0.408 <sub>0.046</sub>	0.412 <sub>0.037</sub>	0.483 <sub>0.044</sub>	0.499 <sub>0.035</sub>	0.435 <sub>0.046</sub>	0.453 <sub>0.042</sub>
		0.5	0.416 <sub>0.050</sub>	0.431 <sub>0.041</sub>	0.354 <sub>0.043</sub>	0.370 <sub>0.034</sub>	0.524 <sub>0.041</sub>	0.541 <sub>0.032</sub>	0.489 <sub>0.048</sub>	0.501 <sub>0.037</sub>
		0.9	0.295 <sub>0.066</sub>	0.264 <sub>0.067</sub>	0.238 <sub>0.058</sub>	0.231 <sub>0.072</sub>	0.529 <sub>0.065</sub>	0.558 <sub>0.033</sub>	0.476 <sub>0.067</sub>	0.524 <sub>0.069</sub>
	10.0	0.0	0.109 <sub>0.031</sub>	0.126 <sub>0.036</sub>	0.066 <sub>0.023</sub>	0.091 <sub>0.034</sub>	0.109 <sub>0.031</sub>	0.126 <sub>0.036</sub>	0.066 <sub>0.023</sub>	0.091 <sub>0.034</sub>
		0.1	0.093 <sub>0.016</sub>	0.110 <sub>0.035</sub>	0.058 <sub>0.030</sub>	0.070 <sub>0.029</sub>	0.118 <sub>0.030</sub>	0.123 <sub>0.030</sub>	0.074 <sub>0.031</sub>	0.078 <sub>0.027</sub>
		0.5	0.056 <sub>0.029</sub>	0.057 <sub>0.025</sub>	0.024 <sub>0.020</sub>	0.027 <sub>0.022</sub>	0.131 <sub>0.036</sub>	0.148 <sub>0.033</sub>	0.104 <sub>0.047</sub>	0.102 <sub>0.051</sub>
		0.9	0.000 <sub>0.007</sub>	0.000 <sub>0.000</sub>	0.000 <sub>0.000</sub>	0.000 <sub>0.000</sub>	0.154 <sub>0.039</sub>	0.166 <sub>0.040</sub>	0.130 <sub>0.044</sub>	0.126 <sub>0.034</sub>
SPEA2	1.0	0.0	0.521 <sub>0.057</sub>	0.587 <sub>0.037</sub>	0.402 <sub>0.044</sub>	0.484 <sub>0.064</sub>	0.521 <sub>0.057</sub>	0.587 <sub>0.037</sub>	0.402 <sub>0.044</sub>	0.484 <sub>0.064</sub>
		0.1	0.527 <sub>0.051</sub>	0.592 <sub>0.034</sub>	0.425 <sub>0.068</sub>	0.493 <sub>0.055</sub>	0.560 <sub>0.044</sub>	0.613 <sub>0.086</sub>	0.462 <sub>0.054</sub>	0.541 <sub>0.072</sub>
		0.5	0.536 <sub>0.053</sub>	0.575 <sub>0.064</sub>	0.417 <sub>0.067</sub>	0.502 <sub>0.074</sub>	0.583 <sub>0.044</sub>	0.628 <sub>0.061</sub>	0.439 <sub>0.041</sub>	0.540 <sub>0.054</sub>
		0.9	0.518 <sub>0.066</sub>	0.542 <sub>0.047</sub>	0.418 <sub>0.058</sub>	0.456 <sub>0.056</sub>	0.566 <sub>0.050</sub>	0.623 <sub>0.070</sub>	0.444 <sub>0.071</sub>	0.505 <sub>0.088</sub>
	5.0	0.0	0.446 <sub>0.036</sub>	0.478 <sub>0.029</sub>	0.400 <sub>0.052</sub>	0.408 <sub>0.055</sub>	0.446 <sub>0.036</sub>	0.478 <sub>0.029</sub>	0.400 <sub>0.052</sub>	0.408 <sub>0.055</sub>
		0.1	0.439 <sub>0.045</sub>	0.472 <sub>0.036</sub>	0.395 <sub>0.030</sub>	0.409 <sub>0.035</sub>	0.468 <sub>0.042</sub>	0.497 <sub>0.041</sub>	0.424 <sub>0.054</sub>	0.437 <sub>0.055</sub>
		0.5	0.405 <sub>0.038</sub>	0.417 <sub>0.050</sub>	0.364 <sub>0.050</sub>	0.379 <sub>0.049</sub>	0.499 <sub>0.048</sub>	0.518 <sub>0.076</sub>	0.465 <sub>0.045</sub>	0.482 <sub>0.056</sub>
		0.9	0.318 <sub>0.069</sub>	0.270 <sub>0.110</sub>	0.279 <sub>0.071</sub>	0.243 <sub>0.072</sub>	0.512 <sub>0.054</sub>	0.538 <sub>0.044</sub>	0.464 <sub>0.052</sub>	0.502 <sub>0.053</sub>
	10.0	0.0	0.112 <sub>0.023</sub>	0.129 <sub>0.021</sub>	0.082 <sub>0.029</sub>	0.091 <sub>0.021</sub>	0.112 <sub>0.023</sub>	0.129 <sub>0.021</sub>	0.082 <sub>0.029</sub>	0.091 <sub>0.021</sub>
		0.1	0.103 <sub>0.033</sub>	0.123 <sub>0.025</sub>	0.068 <sub>0.026</sub>	0.078 <sub>0.022</sub>	0.126 <sub>0.042</sub>	0.144 <sub>0.024</sub>	0.084 <sub>0.023</sub>	0.105 <sub>0.031</sub>
		0.5	0.059 <sub>0.024</sub>	0.062 <sub>0.023</sub>	0.039 <sub>0.023</sub>	0.032 <sub>0.022</sub>	0.145 <sub>0.027</sub>	0.156 <sub>0.031</sub>	0.105 <sub>0.018</sub>	0.125 <sub>0.029</sub>
		0.9	0.005 <sub>0.011</sub>	0.000 <sub>0.005</sub>	0.000 <sub>0.001</sub>	0.000 <sub>0.000</sub>	0.177 <sub>0.049</sub>	0.174 <sub>0.041</sub>	0.145 <sub>0.051</sub>	0.151 <sub>0.036</sub>
PAES	1.0	N/A	0.535 <sub>0.083</sub>	0.632 <sub>0.083</sub>	0.398 <sub>0.110</sub>	0.500 <sub>0.052</sub>	0.535 <sub>0.083</sub>	0.632 <sub>0.083</sub>	0.398 <sub>0.110</sub>	0.500 <sub>0.052</sub>
	5.0	N/A	0.496 <sub>0.058</sub>	0.545 <sub>0.084</sub>	0.394 <sub>0.095</sub>	0.455 <sub>0.077</sub>	0.496 <sub>0.058</sub>	0.545 <sub>0.084</sub>	0.394 <sub>0.095</sub>	0.455 <sub>0.077</sub>
	10.0	N/A	0.186 <sub>0.051</sub>	0.211 <sub>0.062</sub>	0.158 <sub>0.065</sub>	0.162 <sub>0.062</sub>	0.186 <sub>0.051</sub>	0.211 <sub>0.062</sub>	0.158 <sub>0.065</sub>	0.162 <sub>0.062</sub>
MOCcell	1.0	0.0	0.598 <sub>0.044</sub>	0.657 <sub>0.033</sub>	0.448 <sub>0.055</sub>	0.539 <sub>0.063</sub>	0.598 <sub>0.044</sub>	0.657 <sub>0.033</sub>	0.448 <sub>0.055</sub>	0.539 <sub>0.063</sub>
		0.1	0.595 <sub>0.034</sub>	0.649 <sub>0.033</sub>	0.465 <sub>0.058</sub>	0.531 <sub>0.053</sub>	0.687 <sub>0.054</sub>	0.717 <sub>0.044</sub>	0.557 <sub>0.066</sub>	0.630 <sub>0.039</sub>
		0.5	0.555 <sub>0.059</sub>	0.605 <sub>0.041</sub>	0.435 <sub>0.071</sub>	0.517 <sub>0.037</sub>	0.685 <sub>0.045</sub>	0.721 <sub>0.060</sub>	0.521 <sub>0.064</sub>	0.625 <sub>0.049</sub>
		0.9	0.506 <sub>0.074</sub>	0.549 <sub>0.050</sub>	0.391 <sub>0.079</sub>	0.472 <sub>0.062</sub>	0.658 <sub>0.055</sub>	0.710 <sub>0.046</sub>	0.513 <sub>0.070</sub>	0.619 <sub>0.090</sub>
	5.0	0.0	0.397 <sub>0.044</sub>	0.412 <sub>0.055</sub>	0.343 <sub>0.044</sub>	0.363 <sub>0.044</sub>	0.397 <sub>0.044</sub>	0.412 <sub>0.055</sub>	0.343 <sub>0.044</sub>	0.363 <sub>0.044</sub>
		0.1	0.375 <sub>0.041</sub>	0.406 <sub>0.068</sub>	0.332 <sub>0.036</sub>	0.358 <sub>0.055</sub>	0.398 <sub>0.050</sub>	0.430 <sub>0.048</sub>	0.357 <sub>0.037</sub>	0.369 <sub>0.067</sub>
		0.5	0.317 <sub>0.080</sub>	0.337 <sub>0.070</sub>	0.282 <sub>0.027</sub>	0.292 <sub>0.054</sub>	0.469 <sub>0.048</sub>	0.501 <sub>0.049</sub>	0.451 <sub>0.042</sub>	0.446 <sub>0.059</sub>
		0.9	0.334 <sub>0.051</sub>	0.329 <sub>0.066</sub>	0.309 <sub>0.084</sub>	0.296 <sub>0.089</sub>	0.521 <sub>0.046</sub>	0.522 <sub>0.067</sub>	0.475 <sub>0.051</sub>	0.512 <sub>0.054</sub>
	10.0	0.0	0.045 <sub>0.016</sub>	0.062 <sub>0.026</sub>	0.023 <sub>0.018</sub>	0.040 <sub>0.018</sub>	0.045 <sub>0.016</sub>	0.062 <sub>0.026</sub>	0.023 <sub>0.018</sub>	0.040 <sub>0.018</sub>
		0.1	0.040 <sub>0.022</sub>	0.055 <sub>0.016</sub>	0.020 <sub>0.021</sub>	0.032 <sub>0.014</sub>	0.052 <sub>0.020</sub>	0.062 <sub>0.024</sub>	0.025 <sub>0.024</sub>	0.042 <sub>0.027</sub>
		0.5	0.015 <sub>0.014</sub>	0.018 <sub>0.030</sub>	0.003 <sub>0.005</sub>	0.003 <sub>0.010</sub>	0.082 <sub>0.042</sub>	0.096 <sub>0.030</sub>	0.040 <sub>0.025</sub>	0.059 <sub>0.019</sub>
		0.9	0.004 <sub>0.013</sub>	0.001 <sub>0.005</sub>	0.000 <sub>0.000</sub>	0.000 <sub>0.000</sub>	0.115 <sub>0.045</sub>	0.112 <sub>0.036</sub>	0.090 <sub>0.030</sub>	0.098 <sub>0.025</sub>

1  
2  
3  
4  
5 configurations, the HV obtained with polynomial mutation was never lower than the one  
6 obtained with random mutation (97.73% improved efficiency). For the crossover operator,  
7 it is RGX that produces the best results: in the 108 test configurations (excluding the ones  
8 with PAES and the ones having  $p_c = 0$  since there is no crossover involved there), RGX  
9 always obtained higher HV than SBX (again 100% improved efficiency). Furthermore,  
10 for the three algorithms including crossover (NSGA-II, SPEA2 and MOCcell), the best  
11 configuration with crossover outperforms 100% of the time the one without crossover.

12 In the parametric configuration, the dominant factor seems to be the mutation proba-  
13 bility, with the highest HV values obtained for  $p_m = 1.0$ . For the crossover, the rate does  
14 not have such a big influence, but the best results are generally obtained with  $p_c = 0.5$ .  
15 The statistical analysis results prove that the algorithmic configurations with SBX, ran-  
16 dom mutation, and PACO/noPACO are statistically similar, and they are statistically  
17 worse than the rest. At the same time, the configuration the uses RGX, polynomial  
18 mutation and PACO is statistically better than the rest.

19 Finally, for the comparison of the algorithms, the results are less clear. MOCcell obtains  
20 the highest HV values (the 10 best performing configurations obtain their highest HV  
21 values if MOCcell is used), but is quite sensitive to the operator configuration and is  
22 outperformed by NSGA-II in the big picture (of the 84 test configurations, NSGA-II  
23 outperforms MOCcell in 57, hence in 67.86%). SPEA2 and PAES produce lower HV  
24 values. In the statistical tests, the best configuration of MOCcell is the one that most often  
25 outperforms any other configuration; as a matter of fact, for any combination of mutation,  
26 crossover and PACO operator, MOCcell with  $p_m = 1.0$  and  $p_c = 0.5$  systematically obtains  
27 the highest number of wins against other algorithms and/or configurations, or is at least  
28 tied for highest number of wins.

29 The 50%-attainment surfaces obtained by the best configuration of each algorithm  
30 both with PACO and without PACO are plotted in Figure 3. For all the four algorithms  
31 the attainment surfaces when PACO is used completely dominate the ones where PACO  
32 is not used. For NSGA-II the region where the number of nodes is below 70 is clearly  
33 improved with PACO, where the one where the number of nodes is beyond 70 differences  
34 are tighter. Finally, the same behavior emerges for MOCcell: the attainment surface of  
35 PACO when the number of nodes is below 80 clearly dominates the one without PACO  
36 and, beyond 80 nodes, the surfaces get closer each other. From the problem's perspective,  
37 this means that for a given number of nodes (i.e., for a fixed cost), the algorithmic  
38 configuration with PACO achieves full terrain coverage with lower energy consumption  
39 than the configuration without PACO, these differences being more apparent as the  
40 number of nodes is reduced.

41 Finally, the 50%-attainment surfaces obtained by the best configuration of each algo-  
42 rithm with PACO are shown in Figure 4. NSGA-II and MOCcell dominate each other  
43 below and beyond configurations with 70 nodes, respectively, and both noticeably dom-  
44 inate SPEA2 and PAES.

45 As a result of the experiments in this section, the following conclusions are drawn:

- 46 • PACO offers a robust enhancement to the performances of the optimization algorithms.
- 47 • The polynomial mutation and RGX crossover are best suited for the WSNL problem.
- 48 • NSGA-II and MOCcell outperform SPEA2 and PAES.

#### 49 6.4. Scalability study

50 The experiment in this section explore the robustness of PACO when solving instances  
51 of higher complexity. The basic problem instance is thus extended into the following:  
52  
53  
54  
55  
56  
57  
58  
59  
60

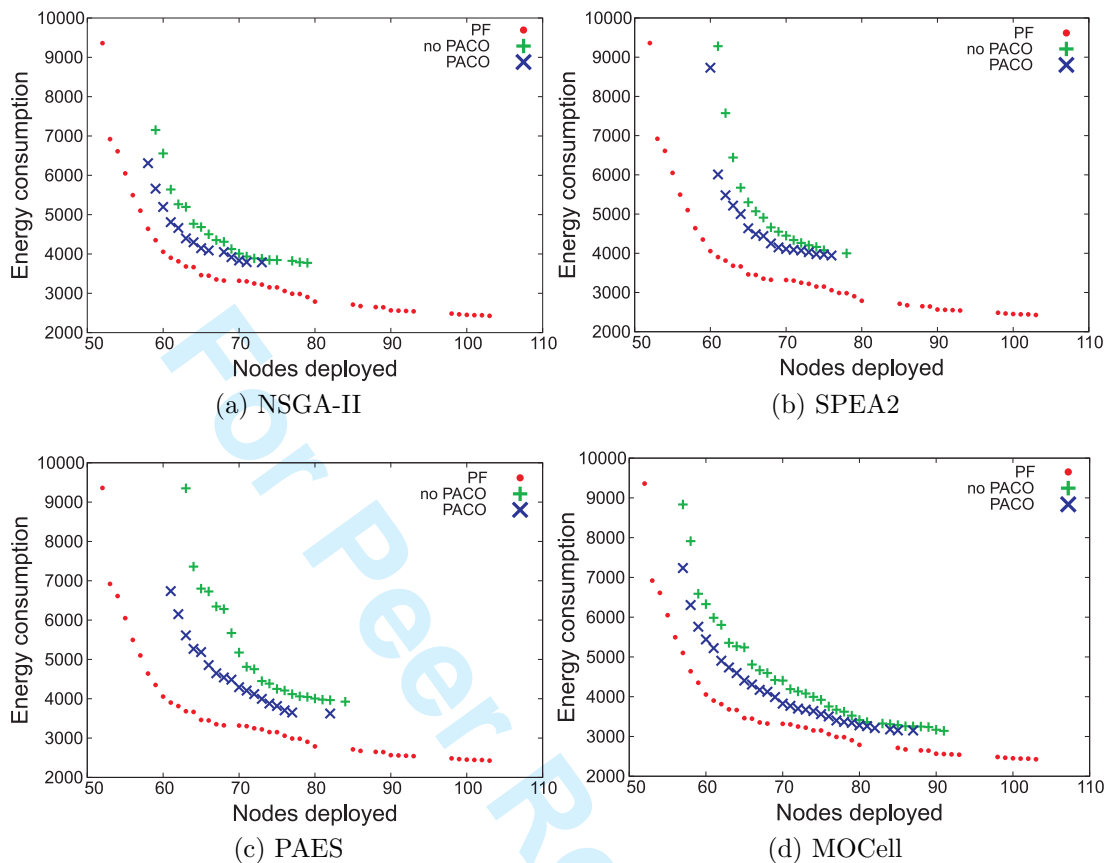


Figure 3.: 50%-attainment surfaces of the optimization algorithms with and without PACO. The global non-dominated fronts are represented for comparison, labeled as 'PF'.

- Square terrain:  $500 \times 500 m^2$ , maximum number of nodes: 1000
- Square terrain:  $750 \times 750 m^2$ , maximum number of nodes: 2000

While the rest of features are left unchanged. These two instances shall be named “instance 500” and “instance 750” for brevity. The best configuration found in the previous section is maintained in this study: polynomial mutation with  $p_m = 1.0$ , RGX crossover with  $p_c = 0.5$ .

Table 4 shows the results for the algorithms both with and without PACO. The first remark is that PACO produces greater gains in performance for larger instances: in the instance 250, the HV value increases by less than 18%, in the instance 500 by at least 24%, and in the instance 750 by more than 292%. Regarding the algorithms, the population-based techniques (NSGA-II, SPEA2 and MOCeLL) suffer HV degradation when the instance grows, MOCeLL still outperforms the other two. PAES, on the other side, seems to be the least affected; in the instance 750, PAES is the best performing technique. This can be explained because the number of function evaluations has been kept the same for all the instance sizes, hence PAES, which converges faster than the population-based techniques, obtains higher HV values.

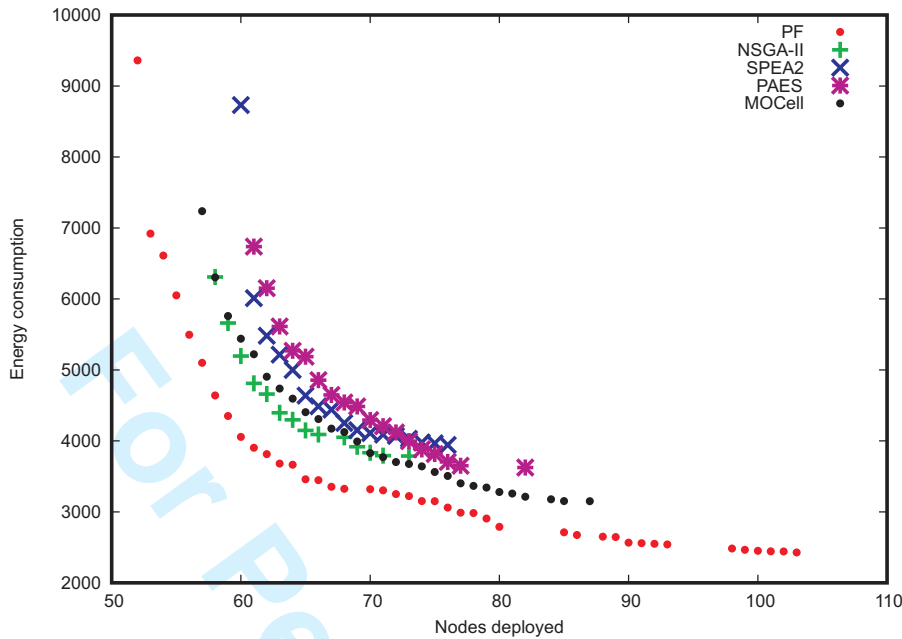


Figure 4.: 50%-attainment surface comparison of the optimization algorithms using PACO.

Table 4.: Scalability properties of the different algorithmic instances (HV, Median and IQR)

Instance	NSGAII		SPEA2		PAES		MOCcell	
	no PACO	PACO	no PACO	PACO	no PACO	PACO	no PACO	PACO
250	0.624 <sub>0.057</sub>	0.688 <sub>0.070</sub>	0.583 <sub>0.044</sub>	0.628 <sub>0.061</sub>	0.535 <sub>0.083</sub>	0.632 <sub>0.083</sub>	0.687 <sub>0.054</sub>	0.717 <sub>0.044</sub>
500	0.470 <sub>0.078</sub>	0.682 <sub>0.080</sub>	0.340 <sub>0.109</sub>	0.588 <sub>0.063</sub>	0.460 <sub>0.084</sub>	0.636 <sub>0.077</sub>	0.574 <sub>0.087</sub>	0.712 <sub>0.093</sub>
750	0.000 <sub>0.000</sub>	0.184 <sub>0.108</sub>	0.000 <sub>0.000</sub>	0.083 <sub>0.092</sub>	0.103 <sub>0.102</sub>	0.404 <sub>0.103</sub>	0.000 <sub>0.003</sub>	0.252 <sub>0.127</sub>

### 6.5. Solutions obtained

The quality estimation parameters are useful tools to compare different techniques or decide the best configuration for a given algorithm, but give poor insight on what solutions can actually be expected from the algorithm. Therefore, this section will briefly discuss the solutions produced by the optimization techniques. Figure 5 shows the solutions at the two opposite extremes of the non-dominated front generated by PACO: The solution with the minimum number of nodes (Figure 5(a)) and the one with minimum energy consumed (Figure 5(b)). As expected from the nature of the problem, solution 5(b) required a higher number of nodes, in order to build more spokes (seven instead of three) that share the traffic load.

## 7. Conclusions and future work

The Proximity Avoidance Coverage-preserving Operator (PACO) for local improvement of the Wireless Sensor Network layout problem (WSNL) is presented in this paper. The WSNL problem consists in deciding the number and geographical locations of the nodes



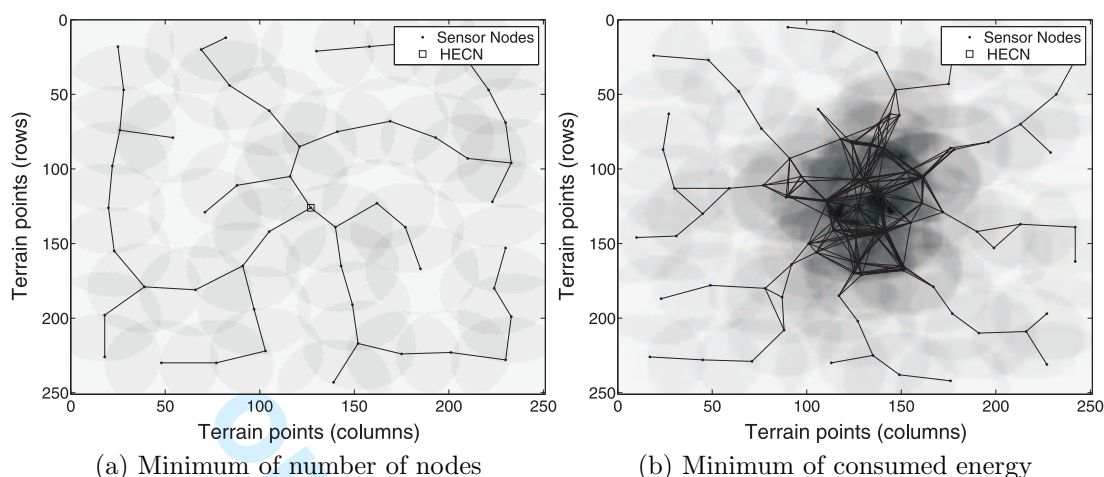


Figure 5.: Best performing solutions produced by MOCcell using PACO for the basic instance:  $250 \times 250 m^2$

of a sensor network to guarantee field coverage by the sensors, while minimizing the cost (number of nodes) and maximizing the lifetime (minimize the energy consumption). Given the twofold opposed optimization objectives, a multi-objective approach is adopted. The main idea behind PACO is to detect local inefficiencies in the network due to redundancy by close neighbors, and look for an equivalent configuration that fixes the inefficiency by replacing the two close nodes by a single node. The equivalent configuration, which must maintain the network connectivity and coverage levels, reduces the number of nodes, and possibly the energy consumption (since less information packets are generated).

In the experimental tests, PACO has demonstrated to produce a significant enhancement of the quality of the fronts produced (measured with hypervolume and attainment surfaces) in four different multi-objective techniques. The proposed operator has displayed a robust behavior on the one hand, and an improved performance when the basic algorithm (without PACO) obtains high HV values on the other. Furthermore, the benefits of the operator show good scaling properties towards the size of the problem instance: the difference between PACO-equipped algorithms and basic optimization algorithms increase as the problem instance grows in size.

Further research will focus in testing PACO on new and more complete contexts. In this sense, the consideration of the network's robustness against node failures, or the use of more complex, irregular terrain models to simulate specific environments are immediate steps.

## Acknowledgments

This work has been partially funded by the "Consejería de Innovación, Ciencia y Empresa", Junta de Andalucía under contract P07-TIC-03044, the DIRICOM project (<http://diricom.lc.uma.es>), and the Spanish Ministry of Science and Innovation and FEDER under contract TIN2008-06491-C04-01, the M\* project

(<http://mstar.lcc.uma.es>). Guillermo Molina acknowledges support from grant AP2005-0914 from the Spanish government. Francisco Luna acknowledges support from the grant BES-2006-13075 funded by the Spanish government.

## References

- Aitsaadi, N., *et al.*, 2008. A Tabu Search Approach for Differentiated Sensor Network Deployment. *In: Proceedings of the 5th IEEE Consumer Communications and Networking Conference, 2008. CCNC 2008.*, jan.. IEEE Computer Society Press, 163–167.
- Akyildiz, I., *et al.*, 2002. A Survey on Sensor Networks. *IEEE Communications Magazine*.
- Bai, X., *et al.*, 2006. Deploying wireless sensors to achieve both coverage and connectivity. *In: MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, Florence, Italy New York, NY, USA: ACM, 131–142.
- Biagioni, E.S. and Sasaki, G., 2003. Wireless Sensor Placement For Reliable and Efficient Data Collection. *In: Proceedings of the Hawaii International Conference on Systems Sciences, HICSS 03*, Vol. 5, Jan. Los Alamitos, CA, USA: IEEE Computer Society, p. 127b.
- Blum, C. and Roli, A., 2003. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*, 35 (3), 268–308.
- Brass, P., 2007. Bounds on coverage and target detection capabilities for models of networks of mobile sensors. *ACM Trans. Sen. Netw.*, 3 (2), 9.
- Cardei, M. and Wu, J., 2006. Energy-efficient coverage problems in wireless ad-hoc sensor networks.. *Computer Communications*, 29 (4), 413–420.
- Cevher, V. and M.Kaplan, L., 2009. Acoustic sensor network design for position estimation. *ACM Trans. Sen. Netw.*, 5 (3), 1–28.
- Chen, Y., Chuah, C.N., and Zhao, Q., 17-20 Oct. 2005. Sensor placement for maximizing lifetime per unit cost in wireless sensor networks. *Military Communications Conference, 2005. MILCOM 2005. IEEE*, 1097–1102 Vol. 2.
- Cheng, M.X., Ruan, L., and Wu, W., 2005. Achieving minimum coverage breach under bandwidth constraints in wireless sensor networks. *In: INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, Vol. 4, march., 2638 – 2645.
- Coello Coello, C.A., Lamont, G.B., and Van Veldhuizen, D.A., 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Second New York: Springer ISBN 978-0-387-33254-3.
- Culler, D., Estrin, D., and Srivastava, M., 2004. Overview of Sensor Networks. *IEEE Computer*, 37 (8), 41–49.
- Dargie, W. and Poellabauer, C., 2010. *Fundamentals of Wireless Sensor Networks: Theory and Practice*. Wiley.
- Deb, K., 2001. *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons.
- Deb, K. and Agrawal, R., 1995. Simulated Binary Crossover for Continuous Search Space. *Complex Systems*, 9, 115–148.
- Deb, K., *et al.*, Apr 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6 (2), 182–197.
- Dhillon, S. and Chakrabarty, K., 2003. Sensor placement for effective coverage and surveillance in distributed sensor networks. *In: Proceedings of the IEEE Wireless Communications and Networking Conference* Los Alamitos, CA, USA: IEEE Computer

- Society, 1609–1614.
- Durillo, J., Nebro, A., and Alba, E., 2010. The jMetal Framework for Multi-Objective Optimization: Design and Architecture. *In: IEEE Congress on Evolutionary Computation, CEC'2010*, 4138 – 4325.
- Esseghir, M., Bouabdallah, N., and Pujolle, G., 2005. Sensor placement for maximizing wireless sensor network lifetime. *In: Vehicular Technology Conference, 2005. VTC-2005-Fall. 2005 IEEE 62nd*, Vol. 4, sept..., 2347 – 2351.
- Ferentinos, K.P. and Tsiligiridis, T.A., 2010. A memetic algorithm for optimal dynamic design of wireless sensor networks. *Comput. Commun.*, 33 (2), 250–258.
- Ganesan, D., Cristescu, R., and Beferull-Lozano, B., 2006. Power-efficient sensor placement and transmission structure for data gathering under distortion constraints. *ACM Trans. Sen. Netw.*, 2 (2), 155–181.
- Jain, E. and Liang, Q., 2005. Sensor placement and lifetime of wireless sensor networks: theory and performance analysis. *In: Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, Vol. 1, nov.-2 dec., 173–177.
- Jourdan, D.B. and Roy, N., 2008. Optimal sensor placement for agent localization. *ACM Trans. Sen. Netw.*, 4 (3), 1–40.
- Jourdan, D. and de Weck, O., 2004. Layout Optimization for a Wireless Sensor Network Using a Multi-Objective Genetic Algorithm. *In: Proceedings of the IEEE Semiannual Vehicular Technology Conference*, Vol. 5 Los Alamitos, CA, USA: IEEE Computer Society, 2466–2470.
- Kang, C.W. and Chen, J.H., 2009. Multi-objective evolutionary optimization of 3D differentiated sensor network deployment. *In: GECCO '09: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference*, Montreal, Québec, Canada New York, NY, USA: ACM, 2059–2064.
- Kar, K. and Banerjee, S., 2003. Node placement for connected coverage in sensor networks. *In: Proceedings of WiOpt*, Sophia Antipolis, France.
- Knowles, J., 2005. A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. *In: 5th International Conference on Intelligent Systems Design and Applications (ISDA '05)* Washington, DC, USA: IEEE Computer Society, 552 – 557.
- Knowles, J. and Corne, D., 2000. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8 (2), 149 – 172.
- Knowles, J., Thiele, L., and Zitzler, E., A tutorial on the Performance assessment of stochastic multiobjective optimizers. , 2006. , Technical report 214, Computer Engineering and Networks Laboratory, ETH Zurich.
- Konstantinidis, A., *et al.*, 2010. A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks. *Computer Networks*, 54 (6), 960 – 976 New Network Paradigms.
- Lazos, L. and Poovendran, R., 2006. Stochastic coverage in heterogeneous sensor networks. *ACM Trans. Sen. Netw.*, 2 (3), 325–358.
- Lazos, L., Poovendran, R., and Ritcey, J.A., 2009. Analytic evaluation of target detection in heterogeneous wireless sensor networks. *ACM Trans. Sen. Netw.*, 5 (2), 1–38.
- Lédeczi, A., *et al.*, 2005. Countersniper system for urban warfare. *ACM Trans. Sen. Netw.*, 1 (2), 153–177.
- Li, X.Y., Song, W.Z., and Wang, Y., 2006. Localized topology control for heterogeneous wireless sensor networks. *ACM Trans. Sen. Netw.*, 2 (1), 129–153.
- Manohar, P., Ram, S.S., and Manjunath, D., 2009. Path coverage by a sensor field: The nonhomogeneous case. *ACM Trans. Sen. Netw.*, 5 (2), 1–26.

- 1  
2  
3  
4  
5 Mladineo, N. and Knezic, S., 13-16 June 2000. Optimisation of forest fire sensor network  
6 using GIS technology. *Information Technology Interfaces, 2000. ITI 2000. Proceedings*  
7 *of the 22nd International Conference on*, 391–396.
- 8 Molina, G., 2010. Optimization techniques for wireless sensor networks. Thesis (PhD).  
9 University of Málaga, Departamento de Lenguajes y Ciencias de la Computación.
- 10 Nan, G. and Li, M., 2008. Evolutionary Based Approaches in Wireless Sensor Networks:  
11 A Survey. In: *Natural Computation, 2008. ICNC '08. Fourth International Conference*  
12 *on*, Vol. 5, oct., 217 –222.
- 13 Nebro, A.J., *et al.*, 2009. A Cellular Genetic Algorithm for Multiobjective Optimization.  
14 *International Journal of Intelligent Systems*, 24 (7), 723 – 725.
- 15 Paschalidis, I.C. and Guo, D., 2009. Robust and distributed stochastic localization in  
16 sensor networks: Theory and experimental results. *ACM Trans. Sen. Netw.*, 5 (4),  
17 1–22.
- 18 Pradhan, P., *et al.*, 2009. Energy Efficient Layout for a Wireless Sensor Network using  
19 Multi-Objective Particle Swarm Optimization. In: *Advance Computing Conference,*  
20 *2009. IACC 2009. IEEE International*, 6-7., 65 –70.
- 21 Sheskin, D.J., 2007. *Handbook of Parametric and Nonparametric Statistical Procedures.*  
22 Chapman & Hall/CRC; 4 edition.
- 23 Shrivastava, N., *et al.*, 2009. Target tracking with binary proximity sensors. *ACM Trans.*  
24 *Sen. Netw.*, 5 (4), 1–33.
- 25 Singh, S., Woo, M., and Raghavendra, C.S., 1998. Power-Aware Routing in Mobile Ad  
26 Hoc Networks. In: *MobiCom '98: Proceedings of the 4th annual ACM/IEEE interna-*  
27 *tional conference on Mobile computing and networking* Dallas, Texas: ACM, 181–190.
- 28 Thai, M.T., *et al.*, 2008. Coverage problems in wireless sensor networks: designs and  
29 analysis. *Int. J. Sen. Netw.*, 3 (3), 191–200.
- 30 Wettergren, T.A. and Costa, R., 2009. Optimal placement of distributed sensors against  
31 moving targets. *ACM Trans. Sen. Netw.*, 5 (3), 1–25.
- 32 Woehrle, M., *et al.*, 2010. Investigating Coverage and Connectivity Trade-offs in Wireless  
33 Sensor Networks: The Benefits of MOEAs. In: M. Ehrgott *et al.*, eds. *Multiple Criteria*  
34 *Decision Making for Sustainable Energy and Transportation Systems (MCDM 2008)*,  
35 Vol. 634 of *LNEMS* Heidelberg, Germany: Springer, 211–221.
- 36 Wu, Q., *et al.*, 2007. On efficient deployment of sensors on planar grid. *Comput. Com-*  
37 *mun.*, 30 (14-15), 2721–2734.
- 38 Xu, Y. and Yao, X., 2006. A GA approach to the optimal placement of sensors in wireless  
39 sensor networks with obstacles and preferences. In: *Consumer Communications and*  
40 *Networking Conference, 2006. CCNC 2006. 3rd IEEE*, Vol. 1, 8-10., 127 – 131.
- 41 Younis, M. and Akkaya, K., 2008. Strategies and techniques for node placement in wire-  
42 less sensor networks: A survey. *Ad Hoc Netw.*, 6 (4), 621–655.
- 43 Zhang, X. and Wicker, S.B., 2005. On the optimal distribution of sensors in a random  
44 field. *ACM Trans. Sen. Netw.*, 1 (2), 301–306.
- 45 Zheng, Y., Brady, D.J., and Agarwal, P.K., 2007. Localization using boundary sensors:  
46 An analysis based on graph theory. *ACM Trans. Sen. Netw.*, 3 (4), 21.
- 47 Zitzler, E. and Thiele, L., 1999. Multiobjective evolutionary algorithms: a comparative  
48 case study and the strength pareto approach. *IEEE Transactions on Evolutionary*  
49 *Computation*, 3 (4), 257–271.
- 50 Zitzler, E., Laumanns, M., and Thiele, L., 2002. SPEA2: Improving the Strength Pareto  
51 Evolutionary Algorithms. In: *EUROGEN 2001. Evolutionary Methods for Design, Op-*  
52 *timization and Control with Applications to Industrial Problems*, 95–100.
- 53  
54  
55  
56  
57  
58  
59  
60