Wireless Pers Commun (2015) 84:2187–2213
DOI 10.1007/s11277-015-2548-2

A Specific Heuristic Dedicated to a Coverage/Tracking Bi-objective Problem for Wireless Sensor Deployment

Matthieu Le Berre[1] · Maher Rebai[1] · Faicel Hnaien[2] · Hichem Snoussi[1]

Published online: 3 April 2015

Abstract In recent years, wireless sensor networks (WSN) have become very attractive for surveillance applications and particularly for target tracking. When a target has to be located by a WSN, accuracy is an important constraint. Most of the studies made in the WSNs problems deal with either coverage or tracking focus objectives. In this paper, we propose a modification of a previously studied bi-objective sensor placement problem taking into account both coverage and accuracy. The objectives are the minimization of the number of deployed sensors and the minimization of the tracking constraints violations, under the coverage constraints. The non sorting genetic algorithm and multi objective particle swarm optimization have been implemented to solve the problem. A specific heuristic (H3P) based on the mathematical decomposition of the problem has also been proposed. The performances of these algorithms are checked with integer programming results for small size instances, and they are compared on large size instances by multi-objective metrics. Results have shown that implemented metaheuristics provide less optimal solutions than the H3P for the small size instances. The comparison between the algorithms on large size instances set show that the H3P dominates the other implemented methods.

Keywords Multi-objective optimization · Wireless sensor networks · Deployment · Tracking · Coverage

✉ Matthieu Le Berre
matthieu.le_berre@utt.fr

Maher Rebai
maher.rebai@utt.fr

Faicel Hnaien
faicel.hnaien@utt.fr

Hichem Snoussi
snoussi@utt.fr

[1] ICD LM2S, UMR 6281, CNRS, University of Technology of Troyes, 10000 Troyes, France

[2] ICD LOSI, UMR 6281, CNRS, University of Technology of Troyes, 10000 Troyes, France

1 Introduction

With recent advances in miniaturization and communication technologies, wireless sensor networks (WSNs) have become attractive for civilian and military applications, such as health and environmental monitoring, or battlefield surveillance. One of the main interests of WSNs for these applications is cheaper installation than wired networks: no communication or energetic infrastructures are needed to allow a WSN to work. In fact, a WSN is composed of a large number of small communicating devices—or sensors such that each sensor has the ability to collect, process and send data across the network. In a WSN, the main aim is to cover an area to monitor interesting positions. For a large number of applications, at least one sensor is required to cover a position. In previous work, we proposed a new mathematical formulation taking into account both deployment cost and tracking accuracy objectives [8]. The mathematical formulation is defined as a bi-objective problem. The first objective is to minimize the number of deployed sensors to cover the entire area. This objective allows us to minimize the deployment cost of the network while maintaining the area coverage. The second objective is to focus on tracking applications, minimizing the non-accuracy. In this paper, we propose a modification of the mathematical model. We consider a binary representation of localization success to reduce the high number of variables and a minimum acceptable accuracy to reduce the high number of constraints. The aim is to offer a set of solutions representing the trade-off between these objectives. In this paper we define an integer linear programming model for this problem and we adapt the non sorting genetic algorithm (NSGA-II) [16] and multi-objective particle swarm optimization (MOPSO) [15] algorithms. We also propose a new specific heuristic based on the mathematical decomposition of the problem, the clustering of decision variables and the optimization of subproblems as Set Covering problems. The performances of these algorithms are compared with optimal solutions obtained by the integer linear model on small instances, and are compared each others on large instances.

Our contributions are as follows:

- the modification of our previous proposed bi-objective mathematical model, allowing to reduce both variables and constraints numbers, taking into account a binary definition of the localization success and a maximum localization radius.
- the development of a new heuristic, based on the mathematical decomposition of the problem.

The remainder of this paper is organized as follows: Sect. 2 summarizes related work. Section 3 introduces a mathematical formulation of the problem. Section 4 describes the dedicated heuristic developed for this problem. Experimental results are detailed in Sect. 5. Section 6 concludes the paper and presents some perspectives for future studies.

2 Related Work

2.1 Coverage Modeling

Coverage is often treated as a critical objective in WSNs. Several models have been considered in the literature [19]. The simplest one is binary coverage. Let $s = (x, y)$ be a sensor and $e = (x_e, y_e)$ be an event, as a target or intruder spawn, (x, y) and (x_e, y_e)

representing respectively the position of s and the position of e. The boolean function $cov_{bin}(e,s)$ expresses the coverage of the event e by the sensor s:

$$cov_{bin}(e,s) = \begin{cases} 1 & \text{if } (\mathrm{x}-\mathrm{x_e})^2 + (\mathrm{y}-\mathrm{y_e})^2 \leq \mathrm{R}^2 \\ 0 & \text{otherwise} \end{cases}$$

where R represents the sensing radius of the sensor s. In the continuous case, this coverage problem can be solved optimally in a 2D space in polynomial time, but it was proven to be NP-hard in 3D space [39]. The probabilistic model is a more realistic adaptation of the binary model. It takes into consideration the degradation of the signal according to the distance between the event and the sensor. Let $cov_{prob}(e,s)$ be the function expressing the coverage of the event e by the sensor s. Let R_a and R_b be both sensing radii, expressing the certitude of detection and the maximal detection range respectively:

$$cov_{prob}(e,s) = \begin{cases} 1 & \text{if } (\mathrm{x}-\mathrm{x_e})^2 + (\mathrm{y}-\mathrm{y_e})^2 \leq \mathrm{R_a^2} \\ 0 & \text{if } (\mathrm{x}-\mathrm{x_e})^2 + (\mathrm{y}-\mathrm{y_e})^2 \geq \mathrm{R_b^2} \\ e^{(-\lambda\alpha^\beta)} & \text{otherwise} \end{cases}$$

with $\alpha = (x-x_e)^2 + (y-y_e)^2 - R_a$. λ and β are technical parameters. This model is used in [23, 41]. The grid square coverage is a discretized adaptation of the binary model. The area to cover is represented as a grid, each cell representing a potential position of an event. The critical square grid coverage problem consists of finding the minimal number of sensors to cover all the cells. This problem has been proven NP-Complete [26]. In this study, we focus on the critical square grid coverage. A large number of studies have been done on this problematic [3, 19, 46]. In [4], the authors have defined a three dimensional sensor deployment problem as a classical Set Covering Problem. This model has been extended to K-coverage, ensuring each position is covered by at least K sensors [7, 18, 36, 43].

2.2 Set Covering Problem

The critical square grid coverage problem is a special case of a classical NP-Complete problem. The Set Covering Problem (SCP) is a classical NP-Complete problem, one of the 21 NP-Complete problems proposed by [25]. Let S the set of sensors and T the set of targets, each sensor of S covers one or more elements of T. The aim is to find the minimum subset of S covering T (Fig. 1).

In addition to be a general case of the coverage problem, the SCP formulation can be used to solve several subproblems in this paper. One of the main advantages is the large number of studies dedicated to the optimization of the SCP. First, the mathematical model has been proposed by [38]:

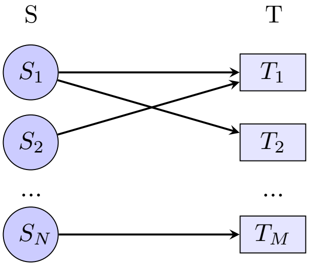

Fig. 1 Set covering problem

Table 1 Notations and their meaning

Notations	Meaning
P	The set of positions in the area
R_{cov}	The sensing radius of sensors
R_L^{max}	The maximum localization radius
$a_{i,j}$	The binary relation giving the coverage of $j \in P$ by $i \in P$

$$Minimize\ z = \sum_{i \in S} c_i X_i \tag{1}$$

$$S.t.:$$

$$\sum_{i \in S} a_{i,j} X_i \geq 1, \quad \forall j \in T \tag{2}$$

$$X_i \in \{0, 1\}, \quad \forall i \in S \tag{3}$$

where X_i is the decision binary variable representing the activation of the sensor S_i. The set of constraints (2) ensures the coverage of all the targets T. To this problem have been added properties to reduce the numbers of variables and constraints [9, 12, 17]. These properties have been used for the adaptation of several exact methods [5, 9]. The authors of [12] have proven the linear solvers are more efficient than the other exact methods for the optimization of this problem. A large number of approximation methods has been also developed for this problem, as greedy algorithms [1, 14, 20], heuristics based on mathematical relaxations [2, 10, 21, 31], GRASP [6, 35] and a large number of metaheuristics [11, 13, 37, 40]. The more efficient method to optimize unicost instances of SCP is the MetaRaps proposed by [29].

2.3 Multi-objective Optimization for WSNs

In this section, we present some works related to multi-objective optimization to solve WSNs problems. In [22, 23], the authors have proposed two works relating to sensors activation problem. First, they have treated a multi-objective problem considering the following two objectives: (i) the maximization of the coverage rate, using a probabilistic coverage model and (ii) the minimization of active sensors. The sensors are randomly deployed in the area. An energy-efficient coverage control algorithm (ECCA) based on NSGA-II algorithm has also been proposed to determine the subset of active sensors. The performances of ECCA have been compared with several algorithms and protocols such as PEAS [44] and OGDC [45], and the results show that ECCA provides better performances than other algorithms. The second work added to this model an energy consumption objective. The authors focus on the minimization of energy consumption considering the sensing radius of each sensor. The model's variables are based on the states (active or sleeping) and the sensing radius of sensors. The authors used an improved NSGA-II outperforming OGDC in several instances. In [32], the authors have developed the normal boundary intersection method (NBI) to solve a multi-objective problem considering: (1) the minimization of the probability of detection error and (2) the minimization of energy consumption. The decision variables are the thresholds of energy detection which determine the minimal signal intensity received by a sensor to send data to the user. The NBI method has been compared to NSGA-II algorithm. Results show that the NBI outperforms NSGA-II.

Other works have been dedicated to the sensor deployment problem, where the decision variables are the sensor positions. In [24], the authors used a multi-objective genetic algorithm (MOGA) to maximize both a binary model coverage and the network lifetime. Similar work has been done in [28], where the authors have defined the multi-objective Deployment and Power Assignment Problem (DPAP). The objectives are the maximization of the grid coverage and the maximization of the network lifetime, adjusting the transmission power levels of the sensors in the grid. The authors used a problem-specific multi-objective evolutionary algorithm based on Decomposition (MOEA/D) to solve this problem, and compared it to NSGA-II. The results showed that MOEA/D outperformed NSGA-II. A variation of this work has been proposed in [27], considering in particular a more accurate energy model and a dense deployment in small area. The authors have also proposed a MOEA/D hybridized with specific heuristic which outperforms the previous MOEA/D and NSGA-II algorithms. The integration of realistic area modeling has been treated in [30], where the authors have considered obstacles, various sensing radii and unreachable places. They have also developed the multi-objective optimization approach for sensor arrangement (MOASA), inspired by the SPEA-II algorithm to solve the following multi-objective problem considering the following objectives: (1) the maximization of the binary coverage, (2) the minimization of the overlap and (3) the minimization of the number of deployed sensors. The results showed the MOASA outperformed the SPEA-II for the resolution of this problem. In [42], the authors have developed a specific MOGA called forced-driven multi-objective genetic algorithm (FD-MOGA) to solve multi-objective 3D deployment problems where the objectives are the maximization of the binary coverage, the maximization of differential detection levels, using a probabilistic coverage model and the minimization of energy consumption. The FD-MOGA has outperformed a classic MOGA during the tests. Other improvements are possible, such as the consideration of user preferences. In [33], the authors used NSGA-II to optimize four objectives: (1) the maximization of the coverage, as a binary model (with various geometric figures to represent the sensing area of a sensor), (2) the minimization of the number of deployed sensors, (3) the maximization of the respect of the user preferences, based on the weighting assigned to the different types of sensors and (4) the minimization of the distance between the target and the sensors. In [7], the authors have adapted the K-coverage deployment model to tracking applications, taking into account the size of the subset of positions covered by the same subset of K sensors. The objectives are the minimization of the number of deployed sensors and the maximization of the number of K-covered positions, under accuracy constraints. NSGA-II has been implemented and two variations of this algorithm have been developed for this problem.

None of these works considers the accuracy for tracking applications as an objective. In this study we propose a modification of a previously proposed mathematical formulation taking into account two contradictory objectives: the minimization of the number of deployed sensors and the minimization of the localization conflicts sum, under total coverage constraints.

3 Problem Formulation

The problem takes place in an area containing positions of interest. The first step is the discretization of the area into a set of positions P. Each position has to be monitored and could contain a sensor. The assumptions are as follows:

- Sensors use a low energy technology as WiFi or ZigBee communication technologies. Using these technologies presents some advantages as the minimization of the impact on users health. The second main advantage is the dual purpose of the network: if WiFi can be used to localize targets, its main usage is the network connexion.
- Due to the used technology, it is not possible to determine the distance considering the signal power level. So the classical localization methods as triangulation are not applicable here. However, we consider it is possible to determine a maximum distance R_{cov}, allowing to use the binary coverage model (see Sect. 2).
- The maximum localization radius R_L^{max} represents the user inaccuracy tolerance, expressing the maximum tolerated size of the research area in case of event detection. This size should be defined considering the sensor network application: smaller is this size, greater is the localization accuracy. The research area is defined as a disk with a radius set to R_L^{max}. In other words, an event detected by the network should be localized in a disk with a radius less or equal to R_L^{max} corresponding to the acceptable minimum accuracy. Otherwise, a conflict has to be declared on the event position.

Notations and their meanings are as follows:

As in our previous works, the aim of this study is to find the optimal deployment taking into account the number of sensors and a tracking focus objective. The first objective is the minimization of the number of sensors, which allow us to reduce the deployment, purchase and maintenance costs. The second objective is to find the optimal configuration allowing us to localize a target. In previous works, we considered the minimization of the localization conflicts, which reduces the sizes of the search areas in the case of an event detection.

Figure 2 shows an example of a sensor deployment with $R_{cov} = 3$. The area is represented by a 11×6 (length by width) grid. The three dark positions represent the deployed sensors to cover and monitor the entire grid. Each disk represents the set of covered positions by a sensor.

Figure 3 shows the apparition of a target in the area. The target is represented by the square position. The target is detected by the first and second sensors, placed in positions (3,2) and (4,6) respectively. The target position belongs to the both sensor 1 and sensor 2 coverage disks.

Figure 4 represents the estimation of the target position by the sensors network. In the case of a binary modeling of the coverage, the only information to localize the target is the subset of sensors detecting it. In the example the target position is in conflict with 9 other positions. In our previous works [8], we proposed to minimize the number of these conflicts.

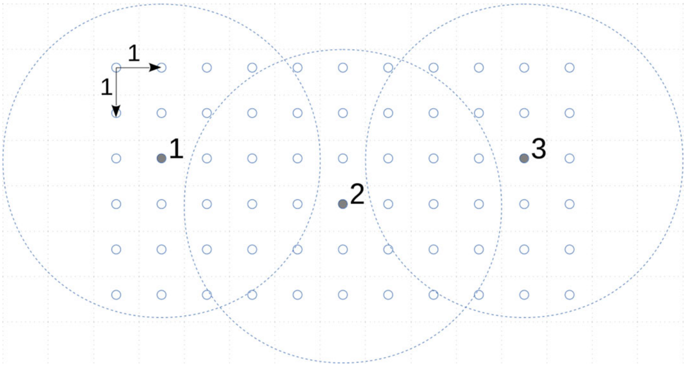

Fig. 2 An example of an area coverage

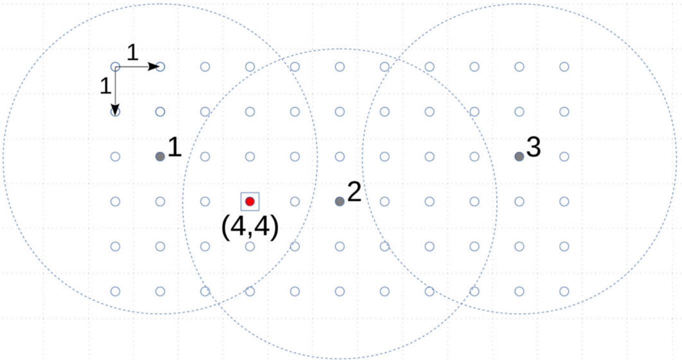

Fig. 3 Detection of a target in the area

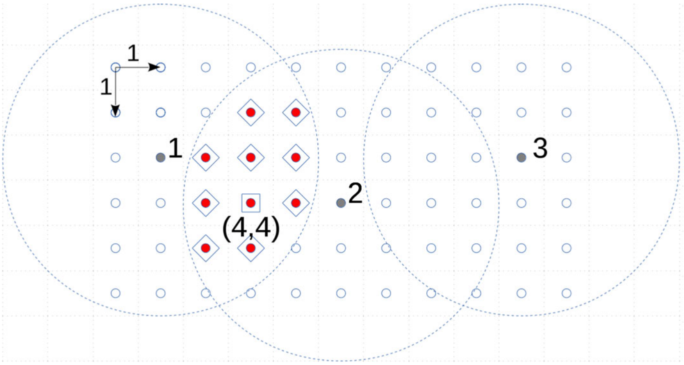

Fig. 4 Estimation of the target position

In this study, we consider a binary representation of the localization: at each position $j \in P$ is associated a binary variable representing the correct localization of an event appearing on position j. The aim is to minimize the number of positions impeding the correct localization of an event. We also add a minimum acceptable accuracy value R_L^{max}. Let j be a position of the grid. We define D_j the disk centered on j. The radius of D_j is equal to R_L^{max}. If an event appears on position j, D_j represents the corresponding acceptable localization area.

Figure 5 represents the previous example, with the minimum acceptable accuracy radius R_L^{max}. In this figure, the value of R_L^{max} is set to 1 and the hatched disk represents $D_{(4,4)}$. In this example, the localization fails since there are some positions in conflict with the position $(4,4)$, which are not belonging to the disk $D_{(4,4)}$.

Figure 6 shows a correct localization of the target on the position $(4,4)$. Two sensors have been added to the solution, and all the positions belonging to the same detection subset than the position $(4,4)$ belong to the disk $D_{(4,4)}$.

For each position $i \in P$, we assign a binary decision variable X_i corresponding to the existence of a deployed sensor on the position i. The integer linear model for the coverage problem is as follows:

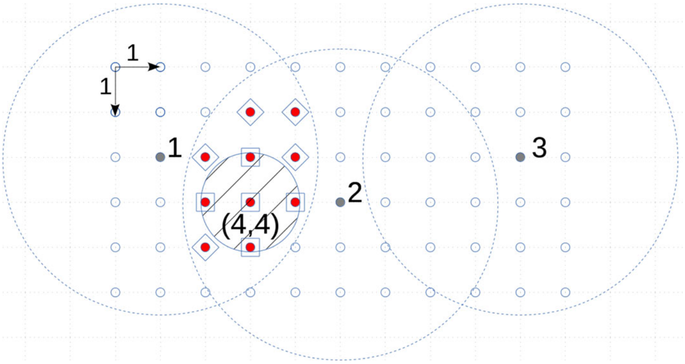

Fig. 5 Incorrect localization considering $R_L^{max} = 1$

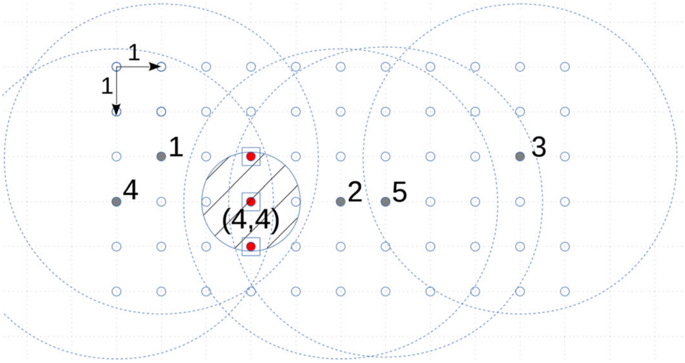

Fig. 6 Correct localization considering $R_L^{max} = 1$

$$Minimize\ z_1 = \sum_{i \in P} X_i \tag{4}$$

$$\begin{gathered} S.t: \\ \sum_{i \in P} a_{i,j} X_i \geq 1, \forall j \in P \end{gathered} \tag{5}$$

where Eq. (4) minimizes the number of deployed sensors. The set of equations (5) represents the coverage constraints. The second objective to be added in this study is the minimization of the non-accuracy. We aim to obtain a trade-off between deployment cost and quality of tracking. Let two positions j and k in P which are covered by the same set of sensors. If the distance between the two positions is greater than the maximum localization radius R_L^{max}, then a conflict has to be declared on the positions j and k. Let C the set of possible conflicts:

$$C \subseteq P^2, \quad \forall (j,k) \in C, j \neq k,$$
$$R_L^{max} < distance(j,k) \leq 2 * R_{cov}$$

The set C contains all couples of positions (j,k), provided the euclidian distance between j and k is greater than R_L^{max} and less or equal to $2 * R_{cov}$ (i.e. the positions which can be detected by the same sensor). In other words, it is acceptable that two positions are detected by the same detection subset only if the position k belongs to the disk D_j (i.e. j belongs to D_k).

Let Y_j be the binary variable associated to the position j. The variable Y_j is equal to 1 only if there is another position k such that the positions j and k are detected by the same sensors and k does not belong to D_j.

$$Y_j = \begin{cases} 1 & \text{if } \exists \mathrm{k} \in \mathrm{P}, \mathrm{k} \notin \mathrm{D_j}, \text{ as } \forall \mathrm{i} \in \mathrm{P}, \mathrm{a_{i,j} X_i} = \mathrm{a_{i,k} X_i} \\ 0 & \text{otherwise} \end{cases}$$

In fact, if the two positions j and k are covered by a same subset of sensors (i.e. $\sum_{i \in P} |a_{i,j} - a_{i,k}| X_i = 0$), therefore Y_j and Y_k will be equal to 1.

The bi-objective mathematical model is as follows:

$$Minimize\ z_1 = \frac{1}{|P|} \sum_{i \in P} X_i \tag{4}$$

$$Minimize\ z_2 = \frac{1}{|P|} \sum_{j \in P} Y_j \tag{6}$$

$$S.t:$$

$$\sum_{i \in P} a_{i,j} X_i \geq 1, \forall j \in P \tag{5}$$

$$\sum_{i \in P} |a_{i,j} - a_{i,k}| X_i + \frac{1}{2}(Y_j + Y_k) \geq 1, \forall (j,k) \in C \tag{7}$$

$$X_i \in \{0,1\}, \forall i \in P \tag{8}$$

$$Y_j \in \{0,1\}, \forall j \in P \tag{9}$$

The first objective z_1 (4) minimizes the number of deployed sensors divided by the grid size. The second objective z_2 (6) minimizes the number of positions entraving the correct localization of an event considering the minimum acceptable accuracy R_L^{max}. The first set of constraints (5) ensures the total grid coverage. The set of constraints (7) allows us to determine if two positions j and k belong to the same detection subset or not. The remaining constraints (8) and (9) are binary decision variables constraints. Note that all variables Y_j could be set as a real variable to allow linear solvers to use different methods.

The main advantages compared to our previous proposed model [8] are following: (1) the number of variables is reduced to $2 * |P|$ instead of a maximum value of $|P| + \frac{|P|*(|P|-1)}{2}$ and (2) the number of localization constraints is also reduced considering the minimum acceptable accuracy R_L^{max}.

4 Multi-objective Optimization

The two objective functions usually do not have the same optimal solution (they are contradictory). The multi-objective optimization allows the optimization of different objectives simultaneously. For multi-objective problems, the Pareto dominance is usually used. Considering a bi-objective minimization problem, where f_1 and f_2 are the objective functions, let u and v be two configurations. We have the following situations:

- u dominates v if $(f_1(u) < f_1(v)) \wedge (f_2(u) < f_2(v))$
- u is dominated by v if $(f_1(u) > f_1(v)) \wedge (f_2(u) > f_2(v))$
- u and v are not comparable if $(f_1(u) < f_1(v)) \wedge (f_2(u) > f_2(v))$ or $(f_1(u) > f_1(v)) \wedge (f_2(u) < f_2(v))$

The optimal Pareto front of a multi-objective problem is the set of the non-dominated solutions. A configuration u belongs to the optimal Pareto front if no other existing configuration dominates it. This approach allows to compare configurations considering a various number of objectives.

In addition to the NSGA-II [16] and MOPSO [15] algorithms, we propose a dedicated heuristic based on the mathematical decomposition of the problem into a set of SCP.

4.1 NSGA-II

NSGA-II [16] is a multi-objective genetic algorithm used to solve multi-objective problems. The basic process of this algorithm is as follows:

Algorithm 1 NSGA-II algorithm

```
Require: $Iter, Size_{Pop}, Size_{Sel}$
  $P_0 \leftarrow$ initial population
  for $i = 1$ to $Iter$ do
      $M_i \leftarrow$ reproductionSelection($P_{i-1}$)
      $O_i \leftarrow$ {crossover + mutation +reparation} ($M_i$)
      $P_i \leftarrow P_{i-1} + O_i$
      $P_i \leftarrow$ {Pareto dominance + Crowding distance} ($P_i$)
  end for
  return non-dominated configurations
```

The first step is the initialization of the population by $Size_{Pop}$ solutions inserted into P_0. At each iteration i, a selection for the reproduction is done among the configurations of P_{i-1}, giving the mating pool M_i. New solutions are created by crossover and mutation processes, giving the offspring O_i. A reparation operator is used to satisfy the coverage constraints. P_{i-1} and O_i are then pushed in P_i. NSGA-II sorts configurations in fronts, each front contains non-dominated solutions selected according to Pareto dominance and Crowing distance criteria.

4.2 MOPSO

MOPSO [15] is a multi-objective particle swarm algorithm. This algorithm is based on the evolution of both velocities and positions to update the particles in the parameters space. Let p a particle at the iteration t, the position and velocity update is done as follows:

$$\begin{aligned} v_{t+1}(p) = wv_t(p) + c_1(m_t(p) - x_t(p)) \\ + c_2(x_t(q) - x_t(p)) \end{aligned} \tag{10}$$

$$x_{t+1}(p) = v_{t+1}(p) + x_t(p) \tag{11}$$

where v_t, x_t, m_t and q represent the velocity, the position, the best position is the memory and the best close particle respectively. The parameter w represents the impact of the current velocity, allowing to diversify the space exploration. The parameters c_1 and c_2 represent the impact of the best position crossed by the particle and the impact of the neighborhood respectively. The main algorithm can be seen in Algorithm 2.

Algorithm 2 MOPSO algorithm

```
Require: $Iter, Size_{Swarm}, w, c_1, c_2$
  $Swarm \leftarrow$ initialization process
  $X* \leftarrow$ the non-dominated solutions in $Swarm$
  for $i = 1$ to $Iter$ do
    Update the velocity and the position of each particle in $Swarm$
    Reparation operator applied on $Swarm$
    Update $X*$
  end for
  return $X*$
```

To optimize binary problems, the authors of [34] have proposed several methods to adapt the exploration space process of BCE and PSO algorithms. In this paper, we use the first method proposed to adapt the MOPSO algorithm to our problem, the velocities representing the probability to set the variable values to 1.

4.3 Hybridization Process

The hybridization process aim to improve the current algorithms NSGA-II and MOSPO. At each solution X is assigned two values λ_1^X and λ_2^X corresponding to the weights of the objective functions. For each new solution, a problem specific operator is applied to ensure the satisfaction of the coverage constraints. This operator is similar to the greedy algorithm dedicated to the Set Covering problem [14]. The process is as follows: while there is at least one coverage constraint unsatisfied, a variable is selected considering a greedy score and is added to the current solution. Here we propose a new greedy score calculation, considering both cost and localization objectives.

$$\forall i \in P, score_i \leftarrow \lambda_1^X \sum_{j \in P} a_{i,j} + \lambda_2^X \sum_{(j,k) \in P^2, j \neq k} |a_{i,j} - a_{i,k}| \tag{12}$$

Considering the two values λ_1^X and λ_2^X, the solution X will promote the first or second objective.

The one-point crossover is also modified to take into account the λ coefficients. First, the couples of selected solutions are defined considering the λ coefficients matching. Let two selected solutions X and X' and the two new solutions Y and Y' generated by the crossover operator, the λ values calculation is as follows:

$$\begin{aligned} \lambda_1^Y &= \alpha\lambda_1^X + (1-\alpha)\lambda_1^{X'} \\ \lambda_2^Y &= \alpha\lambda_2^X + (1-\alpha)\lambda_2^{X'} \\ \lambda_1^{Y'} &= (1-\alpha)\lambda_1^X + \alpha\lambda_1^{X'} \\ \lambda_2^{Y'} &= (1-\alpha)\lambda_2^X + \alpha\lambda_2^{X'} \end{aligned}$$

where α is a real value between 0 and 1, considering the crossover point chose by the operator.

The λ coefficient are also used in MOPSO to guide the evolutionary process. The selection of the best neighbor and the best position in the memory is done taking into account the weighted objective sum, considering λ_1 and λ_2 the weights of z_1 and z_2 respectively.

4.4 Heuristic H3P

Each conflict variable Y_j is linked to several localization constraints. Each localization constraint can be decomposed into two parts: the first part is the sum of deployment variables allowing to avoid a conflict on the position j, and the second part is the declaration of the conflict (see Eq. 13).

$$\sum_{i \in P} |a_{i,j} - a_{i,k}| X_i + \frac{1}{2}(Y_j + Y_k) \geq 1, \quad \forall k \in P, k \notin D_j \tag{13}$$

To avoid a conflict on the position j, the left parts of the constraints have to be greater or equal to one, allowing to fix the value of Y_j to 0. Indeed, if Y_j is equal to zero, the Y_k variable has not enough importance in the equation to satisfy the constraint due to the $\frac{1}{2}$ weight. So both Y_j and Y_k variables can be removed from the constraint (see Eq. 14). In that case, minimizing the number of deployed sensors to avoid a conflict on position j can be formulated as a SCP called SCP_j.

$$\sum_{i \in P} |a_{i,j} - a_{i,k}| X_i \geq 1, \quad \forall k \in P, k \notin D_j \tag{14}$$

Let $Y = \{Y, .., Y_N\}$ be the conflict variables of the problem. If we affect a value to each conflict variable contained in Y, then minimizing the number of deployed sensors to cover each position and satisfy all conflicts considering the affectation is a SCP.

$$Minimize\, z_1(Y) = \frac{1}{|P|} \sum_{i \in P} X_i \tag{15}$$

$$\begin{aligned} &S.t. : \\ &\sum_{i \in P} a_{i,j} X_i \geq 1, \quad \forall j \in P \end{aligned} \tag{5}$$

$$\begin{aligned} &\forall Y_j \in Y \; as \; Y_j = 0, \\ &\sum_{i \in P} |a_{i,j} - a_{i,k}| X_i \geq 1, \quad \forall k \in P, k \notin D_j \end{aligned} \tag{16}$$

$$X_i \in \{0, 1\}, \quad \forall i \in P \tag{9}$$

The Eqs. (5, 9, 15, 16) define the subproblem $SCP(Y)$, where the variables Y_j as $Y_j = 0$ add localization constraints (16). The heuristic (H3P) is decomposed into three phases. At each iteration, the heuristic proceeds in a first phase to the clustering of the conflict variables Y_j considering a proximity metric. The second phase consists into the construction of a Pareto front considering the clustering done in the first phase. The third and last phase consists to disassembly of the last solution produced during the second phase. In the following subsections, we describe the process of each phase.

The main process of the heuristic is described in Fig. 7. At each phase of the heuristic, an archive operator is called to record non-dominated generated solutions. The process is done iteratively until the execution time is reached.

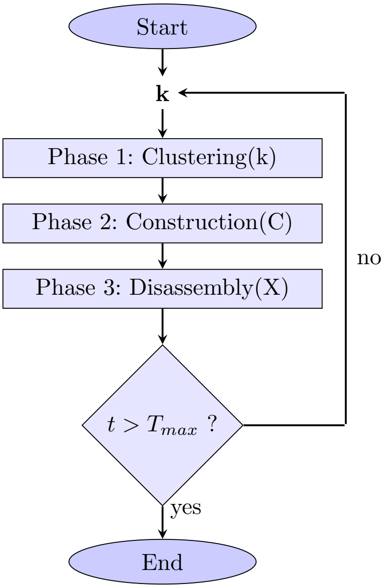

Fig. 7 *H*3*P* general process

4.4.1 Phase 1: Clustering

The aim of this part is to cluster the conflict variables. Let K be the number of clusters, K is determined randomly. Let $C = \{C_1, .., C_N\}$ the set of K clusters, each cluster contains a subset of Y. The conflict variables Y_j are inserted into clusters considering a metric allowing to compare them each others. This proximity metric is based on the occurrence numbers of all deployment variables in the satisfaction of the subproblem SCP_j. At each variable Y_j is associated an occurrence vector V_j as follows:

$$V_j = \left\{ V_j^1, ..., V_j^{|P|} \right\} \tag{17}$$

$$V_j^i = \sum_{k \in P, k \neq j} |a_{i,j} - a_{i,k}|, \forall i \in P \tag{18}$$

where V_j^i is the number of constraints linked to Y_j satisfied by the variable X_i. Let Y_j and Y_k two conflict variables, the proximity metric $M(Y_j, Y_k)$ is computed as an euclidian distance:

$$M(Y_j, Y_k) = \sum_{i \in P} (V_j^i - V_k^i)^2 \tag{19}$$

where V_j and V_k are the occurrence vectors associated to the variables Y_j and Y_k respectively. This metric is also used to compare a variable Y_j to a cluster C_h:

$$M(Y_j, C_h) = \sum_{i \in P} \left(V_j^i - \frac{1}{C_h} \sum_{Y_k \in |C_h|} V_k^i \right)^2 \tag{20}$$

The clustering is done by the K-means algorithm considering the K parameter and the proximity metric, to produce equally sized clusters. The clusters produced will be used in the second phase to construct a Pareto front.

4.4.2 Phase 2: Construction of the Pareto Front

The aim of this phase is to construct a Pareto front considering the clusters produced by the first phase. The construction is done by producing solutions, considering the coverage problem firstly, and adding localization constraints to considerate at each iteration of the phase. The main process of the second phase is described in Fig. 8. Let $SCP(Y)$ be the current subproblem to optimize, this problem is initialized as the coverage problem (i.e. $\forall Y_j \in Y, Y_j = 1$). In other words, no constraints as Eq. (11) are added to the subproblem. A solution X is generated, considering only the coverage constraints. The solution X is initialized as the solution X_{start}, X_{start} being the extreme solution belonging to the archive promoting the first objective. The optimization process is done by an extern optimization method dedicated to the unicost SCP (here we use the MetaRaps method [29]). At each creation or modification of the current solution X, an operator is called to update the archive containing the non-dominated solutions.

While the second objective value of the solution X is different to zero (i.e. there are positions annoying the localization), a cluster $C_h \in C$ is chose considering the minimum greedy score $GHS(C_h, X)$. This score is computed by the classical greedy algorithm [14], determining the number of variables to add to X to satisfy all the constraints linked to the variables Y_j belonging to C_h. All conflict variables Y_j belonging to C_h are added to the current subproblem (i.e. all Y_j are set to zero), and the solution X is optimized considering the new current subproblem.

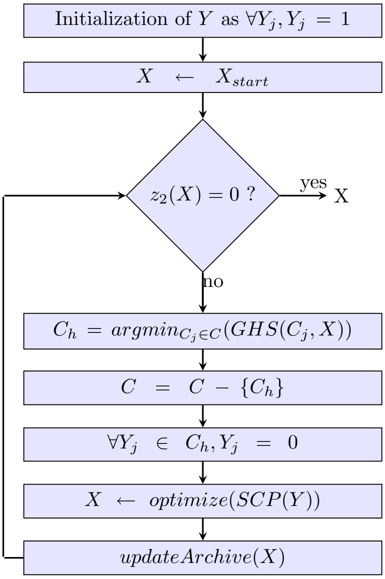
Initialization of Y as $\forall Y_j, Y_j = 1$
$X \leftarrow X_{start}$
$z_2(X) = 0$?
yes
X
no
$C_h = argmin_{C_j \in C}(GHS(C_j, X))$
$C = C - \{C_h\}$
$\forall Y_j \in C_h, Y_j = 0$
$X \leftarrow optimize(SCP(Y))$
$updateArchive(X)$

Fig. 8 *H3P* construction phase

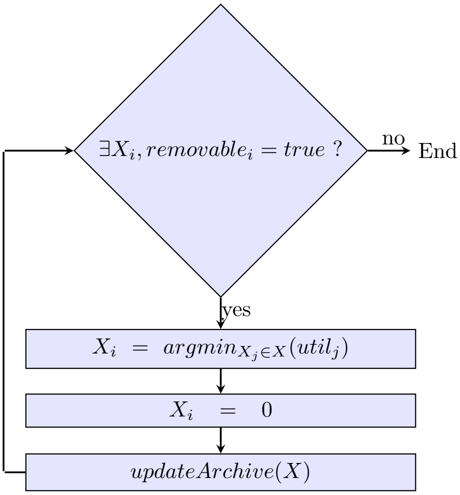
$\exists X_i, removable_i = true$?
no
End
yes
$X_i = argmin_{X_j \in X}(util_j)$
$X_i = 0$
$updateArchive(X)$

Fig. 9 *H3P* disassembly phase

4.4.3 Phase 3: Disassembly the Last Solution

The aim of this phase is to build another Pareto front considering the last solution X generated by the second phase. The objective is to remove variables of the solution while all coverage constraints are satisfied. A variable X_i can be removed from the current solution X only if it satisfies the following condition:

$$removable_i = \left(\forall j \in P, a_{i,j}X_i \neq \sum_{k \in P} a_{k,j}X_k\right) \tag{21}$$

So a variable X_i can be removed only if it does not affect the coverage constraints.

At each variable X_i is attributed an utility score $util_i$ corresponding to the number of localization constraints necessitating the variable X_i to be satisfied (see Eq. 22).

$$\begin{aligned} util_i = &|\{Y_j \in Y, Y_j = 0, \exists k \notin D_j \, as \\ &|a_{i,j} - a_{i,k}|X_i = \sum_{h \in P} |a_{h,j} - a_{h,k}|X_h\}| \end{aligned} \tag{22}$$

The variable with the minimum utility score is removed form the current solution (see Fig. 9).

5 Computational Experiments

5.1 Experimental Conditions

The algorithms have been implemented in C++ language and have been run on a Linux computer with a Core-I5 CPU. The three methods have been compared to the exact solutions for small size instances.

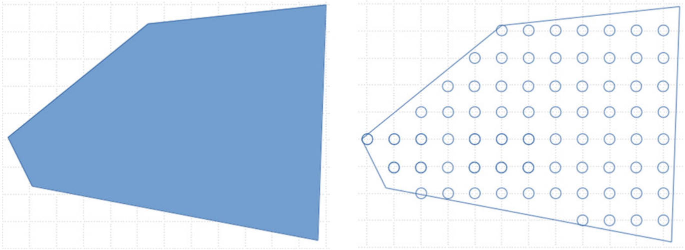

Fig. 10 Example of an area discretization

Let Z be the area to monitor, the set of positions P is defined by the discretization of Z. The positions spread is defined by the user: it can be homogeneous (a grid) or heterogeneous (with high density in interesting subareas and lower density in less interesting ones). The area geometry has to be considered for the discretization (see Fig. 10).

The instances computed by the algorithms are defined by a set of positions P (output of the discretization) and a coverage matrix a (considering P and R_{cov}) where $a_{i,j}$ represents the coverage relation between the positions i and j, with $(i,j) \in P^2$.

In this paper, we consider only homogeneous instances, defined as rectangular grids. The experiment consists of running the algorithms for 2 generated instances sets S and L. The first one contains all the small instances, allowing to compare the algorithms to the optimal Pareto fronts given by Gurobi. The instances set S contains 30 instances, the numbers of lines and columns varying between 6 and 10. The instances set L contains 27 generated instances. For these instances, the dimension of the grid varies within the range [10, 30] for the large size instances. The sensing radius is set to 3. Table 2 shows both instances sets.

The parameters of the NSGA-II and MOPSO are as follows: the population size (NSGA-II) and the swarm size (MOPSO) are set to 120 and 200 respectively. The crossover and mutation probabilities (NSGA-II) are set to 0.9 and $\frac{1}{N}$. The velocity parameters (MOPSO) w, c_1 and c_2 are set to 0.9, 2.5 and 2.5 respectively. The specific heuristic parameters are set as follows: the number of clusters K is randomly set between 2 and 20. The solution X_{start} is computed by the MetaRaps method, with a number of iterations set to 200. For all optimization processes in the second phase, the number of iterations of the MetaRaps is limited to 50. The number of columns to remove of the solutions is set to 70 %. For all the algorithms, the number of iterations has been replaced by a time limit set to 3 minutes.

5.2 Multi-objective Metrics

Among the large number of metrics (quality measures) proposed in the literature, in this paper we have chosen to use two metrics: the coverage of two Pareto fronts (noted C metric) proposed by [47] and the proportion of optimal solutions found by the algorithms (noted *Opt* metric).c The coverage of two Pareto fronts is a relative evaluation of two fronts. A front Q is evaluated by comparison to a front Q' by considering the proportion of solutions in Q' that are dominated by a solution belonging to Q. This measure is not symmetric and has to be computed in both directions due to the possible equivalent non-dominated solutions.

Table 2 Instances sets S and L

	H	W	R_L^{max}		H	W	R_L^{max}
Instance $S1$	6	6	1	Instance $S16$	7	9	2
Instance $S2$	6	6	2	Instance $S17$	7	10	1
Instance $S3$	6	7	1	Instance $S18$	7	10	2
Instance $S4$	6	7	2	Instance $S19$	8	8	1
Instance $S5$	6	8	1	Instance $S20$	8	8	2
Instance $S6$	6	8	2	Instance $S21$	8	9	1
Instance $S7$	6	9	1	Instance $S22$	8	9	2
Instance $S8$	6	9	2	Instance $S23$	8	10	1
Instance $S9$	6	10	1	Instance $S24$	8	10	2
Instance $S10$	6	10	2	Instance $S25$	9	9	1
Instance $S11$	7	7	1	Instance $S26$	9	9	2
Instance $S12$	7	7	2	Instance $S27$	9	10	1
Instance $S13$	7	8	1	Instance $S28$	9	10	2
Instance $S14$	7	8	2	Instance $S29$	10	10	1
Instance $S15$	7	9	1	Instance $S30$	10	10	2

	H	W	R_L^{max}		H	W	R_L^{max}
Instance $L1$	10	15	1	Instance $L15$	15	30	1
Instance $L2$	10	15	2	Instance $L16$	15	30	2
Instance $L3$	10	20	1	Instance $L17$	20	20	1
Instance $L4$	10	20	2	Instance $L18$	20	20	2
Instance $L5$	10	25	1	Instance $L19$	20	25	1
Instance $L6$	10	25	2	Instance $L20$	20	25	2
Instance $L7$	10	30	1	Instance $L21$	20	30	1
Instance $L8$	10	30	2	Instance $L22$	20	30	2
Instance $L9$	15	15	1	Instance $L23$	25	25	1
Instance $L10$	15	15	2	Instance $L24$	25	25	2
Instance $L11$	15	20	1	Instance $L25$	25	30	1
Instance $L12$	15	20	2	Instance $L26$	25	30	2
Instance $L13$	15	25	1	Instance $L27$	30	30	1
Instance $L14$	15	25	2				

$$\begin{gathered} Let\, Q\, and\, Q'\, be\, two\, sets\, of\, solutions, \\ C(Q,Q') = \frac{|\{q' \in Q'\, as\, \exists q \in Q,\ q \prec q'\}|}{|Q'|} \end{gathered} \tag{23}$$

In addition, we use the number of optimal solutions found by the algorithms in the small size instances experimentations. The metric is computed as follows:

$$\begin{gathered} Let\, Q\, be\, a\, set\, of\, solutions\, and\, Q^*\, the\, optimal \\ Pareto\, front, \\ Opt(Q) = \frac{|\{q \in Q \cap Q^*\}|}{|Q^*|} \end{gathered} \tag{24}$$

5.3 Small Size Instances Results

For each instance belonging to the instances set S, the optimal Pareto front has been computed by a linear optimization procedure using the linear solver Gurobi. The experiments results on the small size instances set S are presented in Table 3. The *Opt* metric results values vary between 0 and 1, 0 meaning none optimal solutions have been found and 1 meaning the algorithm has found the complete optimal Pareto front. In Table 3, minimum, average and maximum values of the *Opt* metric are reported for each algorithm and each instance, the algorithms running ten times each instance. The results show that the heuristic $H3P$ provides better solutions than the metaheuristics in average. Indeed the average values $Opt(H3P)$ are greater than the average values of $Opt(NSGAII)$ and $Opt(MOPSO)$ for all the instances of the set S. The heuristic seems to provide an important part of the optimal Pareto front for almost all the instances except the most difficult ones as $S21$ where the $Opt(H3P)$ average value is below 0.5 (i.e. the heuristic provides less than 50 % of the optimal Pareto front in average for this instance).

However, the $H3P$ results are always greater than *MOPSO* and *NSGAII* on this instances set, considering the *Opt* metric. The comparison between the two metaheuristics allows to see the *MOPSO* algorithm is better than the *NSGAII*, expect for the instances $S3$, $S11$ and $S24$. The *NSGAII* seems not able to find optimal solutions for the last instances $S29$ and $S30$, where the $Opt(H3P)$ average values for $S29$ and $S30$ are 0.51 and 0.5 respectively, and the $Opt(MOPSO)$ average values are 0.4 and 0.33 respectively.

5.4 Large Size Instances Results

The experiments results on the large size instances set L are presented in Tables 4, 5 and 6. Considering two methods M_1 and M_2, the C metric results values vary between 0 and 1, 0 meaning the method M_1 does not dominate the solutions provided by the method M_2, and 1 meaning the solutions provided by the algorithm M_1 dominate all the solutions provided by the method M_2. Tables 4, 5 and 6 present the dominance of *NSGAII*, *MOPSO* and $H3P$ respectively.

The dominance is gived by the $C(NSGAII, \_)$ values. The results presented in Table 4 show the *NSGAII* dominates weakly the other algorithms. The $C(NSGAII, \_)$ average values are equal to zero for almost all the instances, except for the instances $L1$, $L2$, $L4$, $L9$, $L13$, $L17$, $L23$ and $L25$ where the dominance of the *NSGAII* over the *MOPSO* is greater than the other instances results. Indeed, the $C(NSGAII, MOPSO)$ average values vary between 0.10 and 0.25 for these instances. If the dominance of the *NSGAII* over the heuristic $H3P$ is weak but not null for some instances, the $C(NSGAII, H3P)$ values are equal to zero for almost all the instances.

Table 5 presents the dominance of the *MOPSO* over the *NSGAII* and $H3P$ algorithms. The average values of $C(MOPSO, NSGAII)$ show that the *MOPSO* algorithm outperformed the *NSGAII* on the large instances set. Indeed, the C metric values are close to one for almost all the instances, meaning the solutions provided by the *MOPSO* algorithm dominates almost all solutions provided by the *NSGAII*. However the dominance of the *MOPSO* over the $H3P$ heuristic is clearly weak, the average values of $C(MOPSO, H3P)$ tend to decrease as the instance size grows.

Table 6 presents the dominance of the $H3P$ heuristic over the implemented metaheuristics. First, the $C(H3P, NSGAII)$ average values are close to one for all the instances, meaning the heuristic outperformed the implemented multi-objective genetic algorithm. Secondly, if the dominance of the $H3P$ over the *MOPSO* is not as great as the dominance

Table 3 *Opt* metric results on the small instances set *S*

	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
	*S*1			*S*2			*S*3		
Opt(NSGAII)	0.86	0.86	0.86	1.00	1.00	1.00	0.71	0.71	0.71
Opt(MOPSO)	0.86	0.86	0.86	1.00	1.00	1.00	0.43	0.43	0.43
Opt(H3P)	1.00	1.00	1.00	1.00	1.00	1.00	0.86	0.93	1.00
	*S*4			*S*5			*S*6		
Opt(NSGAII)	0.60	0.60	0.60	0.00	0.00	0.00	0.60	0.72	0.80
Opt(MOPSO)	1.00	1.00	1.00	0.71	0.71	0.71	1.00	1.00	1.00
Opt(H3P)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	*S*7			*S*8			*S*9		
Opt(NSGAII)	0.38	0.38	0.38	0.60	0.60	0.60	0.25	0.25	0.25
Opt(MOPSO)	0.62	0.62	0.62	1.00	1.00	1.00	0.50	0.56	0.75
Opt(H3P)	0.75	0.85	1.00	1.00	1.00	1.00	0.62	0.79	0.88
	*S*10			*S*11			*S*12		
Opt(NSGAII)	0.40	0.40	0.40	0.75	0.75	0.75	0.60	0.60	0.60
Opt(MOPSO)	1.00	1.00	1.00	0.50	0.50	0.50	1.00	1.00	1.00
Opt(H3P)	1.00	1.00	1.00	0.75	0.91	1.00	1.00	1.00	1.00
	*S*13			*S*14			*S*15		
Opt(NSGAII)	0.22	0.22	0.22	0.40	0.40	0.40	0.11	0.11	0.11
Opt(MOPSO)	0.33	0.33	0.33	0.60	0.60	0.60	0.33	0.33	0.33
Opt(H3P)	0.78	0.86	1.00	1.00	1.00	1.00	0.56	0.72	0.89
	*S*16			*S*17			*S*18		
Opt(NSGAII)	0.33	0.33	0.33	0.00	0.09	0.11	0.33	0.33	0.33
Opt(MOPSO)	0.67	0.67	0.67	0.33	0.39	0.44	0.50	0.50	0.50
Opt(H3P)	1.00	1.00	1.00	0.67	0.71	0.78	0.67	0.82	1.00
	*S*19			*S*20			*S*21		
Opt(NSGAII)	0.11	0.11	0.11	0.20	0.20	0.20	0.11	0.11	0.11
Opt(MOPSO)	0.22	0.23	0.33	0.80	0.80	0.80	0.22	0.22	0.22
Opt(H3P)	0.33	0.54	0.78	1.00	1.00	1.00	0.11	0.36	0.44
	*S*22			*S*23			*S*24		
Opt(NSGAII)	0.40	0.40	0.40	0.00	0.00	0.00	0.40	0.40	0.40
Opt(MOPSO)	0.60	0.60	0.60	0.38	0.38	0.38	0.20	0.20	0.20
Opt(H3P)	0.80	0.96	1.00	0.38	0.53	0.62	1.00	1.00	1.00
	*S*25			*S*26			*S*27		
Opt(NSGAII)	0.12	0.12	0.12	0.00	0.00	0.00	0.12	0.12	0.12
Opt(MOPSO)	0.25	0.25	0.25	0.60	0.60	0.60	0.50	0.50	0.50
Opt(H3P)	0.62	0.74	0.88	1.00	1.00	1.00	0.62	0.72	0.88
	*S*28			*S*29			*S*30		
Opt(NSGAII)	0.60	0.60	0.60	0.00	0.00	0.00	0.00	0.00	0.00
Opt(MOPSO)	0.60	0.60	0.60	0.40	0.40	0.40	0.33	0.33	0.33
Opt(H3P)	0.60	0.74	0.80	0.20	0.51	0.70	0.33	0.50	0.83

Table 4 $C(NSGAII, \_)$ metric results on the large instances set L

	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
	$L1$			$L2$			$L3$		
C(NSGAII,MOPSO)	0.27	0.28	0.29	0.11	0.12	0.12	0.00	0.00	0.00
C(NSGAII,H3P)	0.20	0.29	0.38	0.00	0.00	0.00	0.00	0.01	0.06
	$L4$			$L5$			$L6$		
C(NSGAII,MOPSO)	0.08	0.14	0.17	0.00	0.00	0.00	0.00	0.00	0.00
C(NSGAII,H3P)	0.00	0.18	0.27	0.00	0.07	0.13	0.00	0.06	0.13
	$L7$			$L8$			$L9$		
C(NSGAII,MOPSO)	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.10	0.10
C(NSGAII,H3P)	0.00	0.00	0.00	0.00	0.01	0.06	0.00	0.01	0.06
	$L10$			$L11$			$L12$		
C(NSGAII,MOPSO)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C(NSGAII,H3P)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	$L13$			$L14$			$L15$		
C(NSGAII,MOPSO)	0.19	0.21	0.23	0.00	0.00	0.00	0.03	0.03	0.03
C(NSGAII,H3P)	0.00	0.05	0.16	0.00	0.01	0.05	0.00	0.00	0.00
	$L16$			$L17$			$L18$		
C(NSGAII,MOPSO)	0.00	0.00	0.00	0.18	0.18	0.18	0.00	0.04	0.05
C(NSGAII,H3P)	0.00	0.00	0.00	0.03	0.05	0.11	0.00	0.02	0.14
	$L19$			$L20$			$L21$		
C(NSGAII,MOPSO)	0.05	0.07	0.10	0.00	0.00	0.00	0.00	0.01	0.02
C(NSGAII,H3P)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	$L22$			$L23$			$L24$		
C(NSGAII,MOPSO)	0.00	0.00	0.00	0.23	0.25	0.27	0.00	0.00	0.00
C(NSGAII,H3P)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	$L25$			$L26$			$L27$		
C(NSGAII,MOPSO)	0.13	0.16	0.17	0.00	0.00	0.00	0.03	0.05	0.06
C(NSGAII,H3P)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

over the *NSGAII* for the 8 first instances, the average values of $C(H3P, MOPSO)$ increase quickly as the instance size grows. For almost all the instances, the $H3P$ dominates an important part of the Pareto fronts provided by the metaheuristics. It seems the dominance of the heuristic over the other implemented methods increases as the instance size grows. However we notice an instability of the $C(H3P, MOPSO)$ average values considering the instance, especially for the instance $L28$, where the $C(H3P, MOPSO)$ and $C(MOPSO, H3P)$ average values are 0.63 and 0.28 respectively. In conclusion of the minimum, average and maximum values of the C metric, it seems the $H3P$ heuristic outperformed the implemented metaheuristics, except the *MOPSO* has provided some good quality solutions for a reduced number of instances.

Figure 11 presents the distributions of C metric on the L instances set. For each graph, the bars represent the proportionality of C metric results belonging to the following intervals: $[0, 0.1[$, $[0.1, 0.2[$, $[0.2, 0.3[$, $[0.3, 0.4[$, $[0.4, 0.5[$, $[0.5, 0.6[$, $[0.6, 0.7[$, $[0.7, 0.8[$, $[0.8, 0.9[$ and $[0.9, 1]$. The distributions of $C(NSGAII, \_)$, $C(MOPSO, \_)$, $C(H3P, \_)$, $C(\_, NSGAII)$, $C(\_, MOPSO)$ and $C(\_, H3P)$ are plotted. In

Table 5 $C(MOPSO, \_)$ metric results on the large instances set L

	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
	$L1$			$L2$			$L3$		
C(MOPSO,NSGAII)	0.73	0.74	0.75	0.75	0.82	0.88	0.95	0.96	1.00
C(MOPSO,H3P)	0.21	0.30	0.46	0.12	0.28	0.44	0.11	0.24	0.50
	$L4$			$L5$			$L6$		
C(MOPSO,NSGAII)	0.77	0.79	0.85	1.00	1.00	1.00	1.00	1.00	1.00
C(MOPSO,H3P)	0.00	0.34	0.58	0.13	0.23	0.36	0.07	0.32	0.57
	$L7$			$L8$			$L9$		
C(MOPSO,NSGAII)	1.00	1.00	1.00	1.00	1.00	1.00	0.91	0.91	0.91
C(MOPSO,H3P)	0.04	0.17	0.38	0.12	0.31	0.67	0.05	0.11	0.22
	$L10$			$L11$			$L12$		
C(MOPSO,NSGAII)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
C(MOPSO,H3P)	0.08	0.21	0.33	0.04	0.11	0.15	0.12	0.22	0.31
	$L13$			$L14$			$L15$		
C(MOPSO,NSGAII)	0.82	0.83	0.83	1.00	1.00	1.00	0.98	0.98	0.98
C(MOPSO,H3P)	0.00	0.07	0.14	0.05	0.14	0.23	0.03	0.06	0.13
	$L16$			$L17$			$L18$		
C(MOPSO,NSGAII)	1.00	1.00	1.00	0.78	0.81	0.81	0.96	0.96	0.96
C(MOPSO,H3P)	0.04	0.15	0.30	0.06	0.12	0.32	0.05	0.17	0.36
	$L19$			$L20$			$L21$		
C(MOPSO,NSGAII)	0.90	0.93	0.94	1.00	1.00	1.00	0.98	0.98	1.00
C(MOPSO,H3P)	0.00	0.07	0.14	0.00	0.17	0.38	0.00	0.09	0.21
	$L22$			$L23$			$L24$		
C(MOPSO,NSGAII)	1.00	1.00	1.00	0.70	0.74	0.76	1.00	1.00	1.00
C(MOPSO,H3P)	0.04	0.17	0.32	0.00	0.09	0.25	0.00	0.07	0.27
	$L25$			$L26$			$L27$		
C(MOPSO,NSGAII)	0.83	0.85	0.86	1.00	1.00	1.00	0.90	0.92	0.94
C(MOPSO,H3P)	0.00	0.10	0.27	0.03	0.11	0.18	0.00	0.13	0.25

blue we can see the dominance of a method over the other algorithms, and in red the method proportion to be dominated by the other algorithms.

The NSGA-II results show that at least 80 % of the dominance metric values belong to the interval $[0, 0.1[$ and almost all the $C(NSGAII, \_)$ values belong to the interval $[0, 0.3[$, the algorithm dominating very weakly the other methods. The $C(\_, NSGAII)$ distribution is consistent with these results, the NSGAII method is strongly dominated by the other algorithms. Indeed, most of the $C(\_, NSGAII)$ values belong to $[0.6, 1]$ and at least 80 % of these values belong to $[0.9, 1]$. So the NSGA-II is clearly outperfomed by both MOPSO and H3P algorithms.

The MOPSO results are presented in the $C(MOPSO, \_)$ and $C(\_, MOPSO)$ distribution graphs. We notice an important spread in both distributions. Indeed, an important part of the $C(MOPSO, \_)$ values belong to $[0.7, 1]$, meaning the MOPSO strongly dominates one or more of the other propose algorithms. On the other hand, a large part of these values also belong to $[0, 0.3]$, expressing a weak dominance of the MOPSO over one or more

Table 6 $C(H3P, \_)$ metric results on the large instances set L

	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
	*L*1			*L*2			*L*3		
C(H3P,NSGAII)	0.67	0.74	0.81	0.88	0.99	1.00	0.90	0.97	1.00
C(H3P,MOPSO)	0.50	0.62	0.73	0.38	0.60	0.78	0.42	0.66	0.84
	*L*4			*L*5			*L*6		
C(H3P,NSGAII)	0.77	0.84	1.00	0.89	0.93	0.96	0.86	0.91	1.00
C(H3P,MOPSO)	0.33	0.61	1.00	0.61	0.72	0.87	0.43	0.59	0.86
	*L*7			*L*8			*L*9		
C(H3P,NSGAII)	1.00	1.00	1.00	0.87	0.99	1.00	0.95	0.98	1.00
C(H3P,MOPSO)	0.54	0.73	0.96	0.18	0.64	0.88	0.76	0.84	0.90
	*L*10			*L*11			*L*12		
C(H3P,NSGAII)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
C(H3P,MOPSO)	0.62	0.78	0.92	0.73	0.82	0.92	0.65	0.78	0.88
	*L*13			*L*14			*L*15		
C(H3P,NSGAII)	0.87	0.95	1.00	0.96	1.00	1.00	1.00	1.00	1.00
C(H3P,MOPSO)	0.71	0.85	0.97	0.75	0.83	0.90	0.79	0.89	0.95
	*L*16			*L*17			*L*18		
C(H3P,NSGAII)	1.00	1.00	1.00	0.91	0.94	0.98	0.89	0.97	1.00
C(H3P,MOPSO)	0.70	0.80	0.87	0.56	0.77	0.88	0.50	0.76	0.91
	*L*19			*L*20			*L*21		
C(H3P,NSGAII)	0.94	0.98	1.00	1.00	1.00	1.00	0.98	1.00	1.00
C(H3P,MOPSO)	0.70	0.87	0.93	0.56	0.75	0.92	0.73	0.85	1.00
	*L*22			*L*23			*L*24		
C(H3P,NSGAII)	0.97	0.99	1.00	0.95	0.98	1.00	1.00	1.00	1.00
C(H3P,MOPSO)	0.55	0.72	0.86	0.77	0.89	0.96	0.55	0.87	0.97
	*L*25			*L*26			*L*27		
C(H3P,NSGAII)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
C(H3P,MOPSO)	0.63	0.80	0.94	0.69	0.79	0.91	0.73	0.82	0.97

algorithms. Due to the previous results, we can explain this by a strong dominance of the MOPSO on the NSGA-II and a weak dominance of the MOPSO on the H3P heuristic.

Finally, the H3P results available in the $C(H3P, \_)$ and $C(\_, H3P)$ distribution graphs validate the previous conclusion. Indeed, most of the $C(H3P, \_)$ belong to $[0.7, 1]$, expressing a strong dominance of the H3P heuristic over both MOPSO and NSGAII. Moreover, almost all $C(\_, H3P)$ values belong to the interval $[0, 0.3]$, meaning the H3P is weakly dominated by the other methods on the instances set L. So we can conclude that the heuristic is clearly better than the implemented metaheuristics for the optimization of this problem, considering the instances and the metrics used.

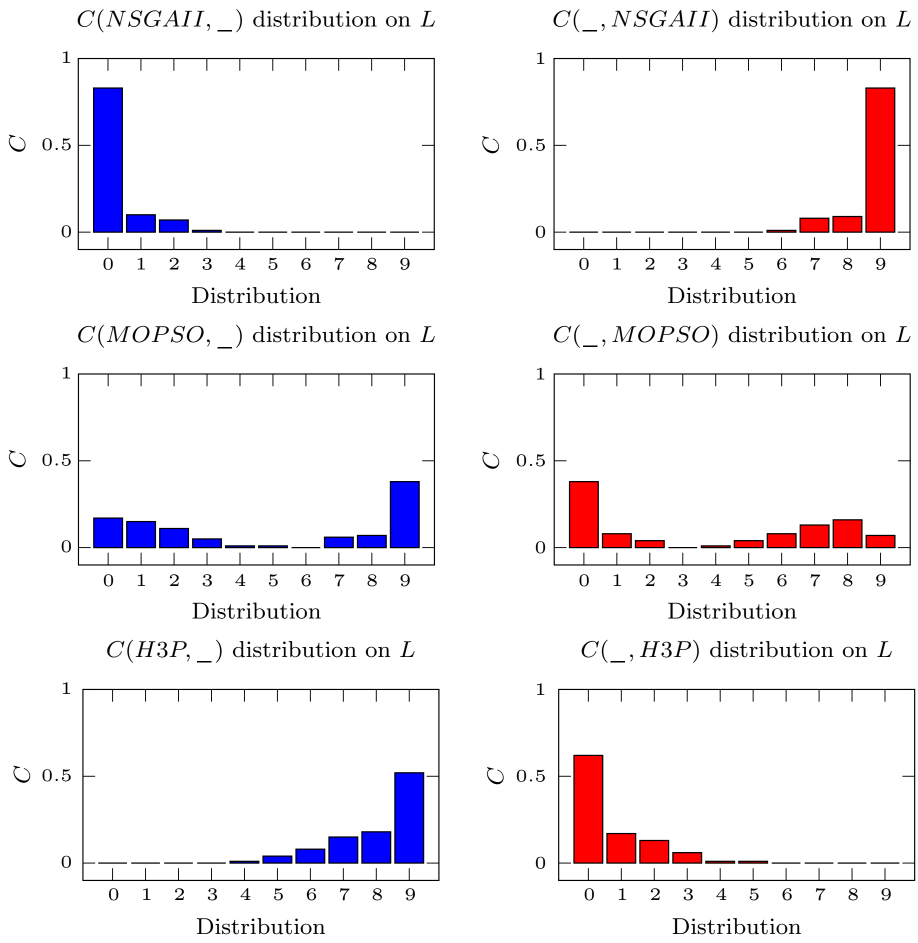

Fig. 11 C metric distribution on the large instances set L

6 Conclusion and Perspectives

In this paper, a multi-objective deployment problem in WSNs is studied. A modification of a previously studied model is proposed, allowing to reduce the high number of the model variables. The deployment of sensors has to ensure total coverage of sensing field for tracking applications. Therefore, we define two objectives to optimize: (1) the minimization of the number of deployed sensors and (2) the minimization of the non-accuracy. The main interest of this modeling is the introduction of accuracy for tracking applications in the problem. We assume that the WSN has to ensure a minimum precision to track targets efficiency. The main difficulty of this problem is the large number of constraints. The optimization is done with NSGA-II and an MOPSO multi-objective optimization algorithms. We also proposed a specific heuristic (H3P) based on the mathematical decomposition of the problem and the clustering of localization variables. We evaluate these algorithms with various size instances, and compare their results with multi-objective metrics. The H3P heuristic has provided more optimal solutions than the implemented metaheuristics for the small size instances. For the large size instances tests, the heuristic dominates a large part of the Pareto front provided by the metaheuristics for almost all the proposed instances.

Our perspectives for future works are following: first, we plain to propose new multi-objective models, taking into account new definitions of accuracy to reduce the numbers of variables and constraints, and also to consider user preferences. We also plain to propose new heuristics based on the models decomposition, considering the SCP subproblems optimization.

Acknowledgments This work is supported by the CapSec-SURECAP project funded by Région Champagne Ardenne and FEDER (Fonds Européen de Développement Régional).

References

1. Almiñana, M., & Pastor, J. T. (1994). Two new heuristics for the location set covering problem. *Top*, *2*, 315–328.
2. Almiñana, M., & Pastor, J. T. (1997). An adaptation of SH heuristic to the location set covering problem. *European Journal of Operational Research*, *100*, 586–593.
3. Altinel, I. K., Aras, N., Güney, E., & Ersoy, C. (2008). Binary integer programming formulation and heuristics for differentiated coverage in heterogeneous sensor networks. *Computer Networks*, *52*, 2419–2431.
4. Andersen, T., & Tirthapura, S. (2009). Wireless sensor deployment for 3d coverage with constraints. In *The 6th International Conference on Networked Sensing Systems*, pp. 78–81.
5. Balas, E., & Carrera, M. C. (1996). A dynamic subgradient-based branch-and-bound procedure for set covering. *Operations Research*, *44*, 875–890.
6. Bautista, J., & Pereira, J. (2007). A grasp algorithm to solve the unicost set covering problem. *Computers and Operations Research*, *34*, 3162–3173.
7. Le Berre, M., Hnaien, F., & Snoussi, H. (2013). A multi-objective modeling of k-coverage problem under accuracy constraint. In *The 5th international conference on modeling, simulation and applied optimization*, pp. 1–6.
8. Le Berre, M., Rebai, M., Hnaien, F., & Snoussi, H. (2014). A bi-objective model for wireless sensor deployment considering coverage and tracking applications. *Accepted for publication in the International Journal of Sensor Networks*. http://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijsnet.
9. Beslay, J. E. (1987). An algorithm for set covering problem. *European Journal of Operational Research*, *31*, 85–93.
10. Beslay, J. E. (1990). A lagrangian heuristic for the set covering problem. *Naval Research Logistics*, *37*, 151–164.
11. Beslay, J. E., & Chu, P. C. (1996). A genetic algorithm for the set covering problem. *European Journal of Operational Research*, *94*, 392–404.
12. Caprara, A., Toth, P., & Fischetti, M. (2000). Algorithms for the set covering problem. *Annals of Operations Research*, *98*, 353–371.
13. Caserta, M. (2007). Tabu search-based metaheuristic algorithm for large-scale set covering problems. In K. F. Doerner, et al. (Eds.), *Metaheuristics: Progress in complex systems optimization* (pp. 43–63). New York: Springer.
14. Chvatal, V. (1979). A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, *4*, 233–235.
15. Coello Coello, C. A., & Lechuga, M. S. (2002). Mopso : A proposal for multiple objective particle swarm. In Proceedings of the 2002 Congress on Evolutionary Computation, pp. 1051–1056.
16. Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm : Nsga-ii. *IEEE Transactions on Evolutionary Computation*, *6*, 182–197.
17. Fisher, M. L., & Kedia, P. (1990). Optimal solution of set covering/partitioning problems using dual heuristics. *Management Science*, *36*, 674–688.
18. Fusco, G., & Gupta, H. (2009). ϵ-net approach to sensor k-coverage. In *Proceedings of the 4th international conference on wireless algorithms, systems, and applications*, pp. 104–114.
19. Ghosh, A., & Das, S. K. (2008). Coverage and connectivity issues in wireless sensor networks: A survey. *Pervasive and Mobile Computing*, *4*, 303–334.
20. Grossman, T., & Wool, A. (1997). Computational experience with approximation algorithms for the set covering problem. *European Journal of Operational Research*, *101*, 81–92.

21. Haddadi, S. (1997). Simple lagrangian heuristic for the set covering problem. *European Journal of Operational Research*, *97*, 200–204.
22. Chen, J., Jia, J., Chang, G., Wen, Y., & Song, J. (2009). Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius. *Computer and Mathematics with Applications*, *57*, 1767–1775.
23. Jia, J., Chen, J., Chang, G., & Tan, Z. (2009). Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm. *Computer and Mathematics with Applications*, *57*, 1756–1766.
24. Jourdan, D.B., & de Weck, O.L. (2004). Layout optimization for a wireless sensor network using a multi-objective genetic algorithm. In: *2004 IEEE 59th vehicular technology conference, 2004. VTC 2004-Spring*. 5:2466–2470.
25. Karp, R.M. (1972). Reducibility amnong combinatorial problems. In *Complexity of computer computations*, pp. 85–103.
26. Ke, W. C., Liu, B. H., & Tsai, M. J. (2011). The critical-square-grid coverage problem in wireless sensor networks is np-complete. *Computer Networks*, *55*, 2209–2220.
27. Konstantinidis, A., & Yang, K. (2011). A multi-objective energy efficient dense deployment in wireless sensor networks using a hybrid problem specific moea/d. *Applied Soft Computing*, *11*, 4117–4134.
28. Konstantinidis, A., Yang, K., Zhang, Q., & Zeinalipour-Yazti, D. (2010). A multi-objective evolutionay algorithm for the deployment and power assignment problem in wireless sensor networks. *Computer Networks*, *54*, 960–976.
29. Lan, G., DePuy, G. W., & Whitehouse, G. E. (2007). An effective and simple heuristic for the set covering problem. *European Journal of Operational Research*, *176*, 1387–1403.
30. Lee, J. Y., Seok, J.-H., & Lee, J.-J. (2012). Multiobjective optimization approach for sensor arrangement in a complex indoor environment. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Applications and Reviews*, *42*, 174–186.
31. Lopes, F. B., & Lorena, L. A. (1994). Surrogate heirstic for set covering problems. *European Journal of Operational Research*, *79*, 138–150.
32. Masazade, E., Varshney, P. K., & Sendur, G. K. (2010). A multiobjective optimization approach to obtain decision thresholds for distributed detection in wireless sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, *40*, 444–457.
33. Oh, S.C., Tan, C.H., Kong, F.W., Tan, Y.S., Ng, K.H., Ng, G.W., & Tai, K. (2007). Multiobjective optimization of sensor network deployment by a genetic algorithm. In *The IEEE congress on evolutionary computation*, pp. 3917–3921.
34. Pampara, G., & Engelbrecht, A.P. (2011). Binary artificial bee colony optimization. In *The IEEE symposium on swarm intelligence*, pp. 1–8.
35. Pessoa, L. S., Resende, M. G. C., & Ribeiro, C. C. (2010). A hybrid lagrangean heuristic with grasp and path-relinking for set k-covering. *Technical report, AT&T Labs Research.*
36. Rebai, M., Khoukhi, I., Snoussi, H., & Hnaien, F. (2013). Linear models for the total coverage problem in wireless sensor networks. In: *The 5th international conference on modeling, simulation and applied optimization*, pp. 1–4.
37. Ren, Z.-G., Feng, Z.-R., Ke, L.-J., & Zhang, Z.-J. (2010). New ideas for applying ant colony optimization to the set covering problem. *Computers & Industrial Engineering*, *58*, 774–784.
38. Roth, R. (1969). Computer solutions to minimum cover problems. *Operations Research*, *17*, 455–465.
39. Rourke, J. O. (1987). *Art gallery theorems and algorithms*. Oxford: Oxford University Press.
40. Solar, M., Parada, V., & Urrutia, R. (2002). A parallel genetic algorithm to solve the set covering problem. *Computers & Operations Research*, *29*, 1221–1235.
41. Wang, X., & Wang, S. (2011). Hierarchical deployment optimization for wireless sensor networks. *IEEE Transactions on Mobile Computing*, *10*, 1028–1041.
42. Wei, L.-C., Kang, C.-W., & Chen, J.-H. (2009). A force-driven evolutionary approach for multi-objective 3d differentiated sensor network deployment. In *The IEEE 6th international conference on mobile adhoc and sensor systems*, pp. 983–988.
43. Wu, Y., Li, M., Cai, Z., & Zhu, E. (2008). A distributed algorithm to approximate node-weighted minimum α-connected (θ, k)-coverage in dense sensor networks. In *Proceedings of the 2nd annual international workshop on frontiers in algorithmics*, pp. 221–232.
44. Ye, F., Zhong, G., Cheng, J., Lu, S., & Zhang, L. (2003). Peas: A robust energy conserving protocol for long-lived sensor networks. In *Proceedings of the 23rd international conference on distributed computing systems*, pp. 28–37.
45. Zhang, H., & Hou, J. C. (2005). Maintaining sensing coverage and connectivity in large sensor networks. *Ad-hoc and Sensor Wireless Networks*, *1*, 89–124.

46. Zhu, C., Zheng, C., Shu, L., & Han, G. (2011). A survey on coverage and connectivity issues in wireless sensor networks. *Journal of Network and Computer Applications*, *35*, 619–632.
47. Zitzler, E., & Thiele, L. (1998). An evolutionary algorithm for multiobjective optimization: The strength pareto approach. In *Technical Report 43*, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich.

Matthieu Le Berre received the Master degree in computer science from the University of Angers in 2010. He obtained his Ph.D. degree in Operational Research at the University of Technology of Troyes, France (UTT) in 2014. Since 2014, he is a postdoctoral researcher at the UTT. His research is focused on multi-objective optimization dedicated to sensor deployment problems.

Maher Rebai received the Master degree in Operation Research and Management of Operations from the Faculty of Economic and Management of Sfax (Tunisia) in 2008. From 2008 to 2011 he obtained his Ph.D. at the UTT, in Optimization and Surety of systems. Since 2011, he is a postdoctoral researcher in Operation Research at the UTT. His research interests include combinatorial optimization, scheduling, wireless sensor networks optimization.

Faicel Hnaien graduated in industrial engineering from the National Engineering School of Tunis (Tunisia-ENIT) in 2004. He received his Master's degree at the Polytechnic of Grenoble (INPG) in industrial engineering in 2005. From 2005 to 2008 he obtained his Ph.D. at the Ecole des Mines de Saint Etienne, France, in industrial engineering. From 2008 to 2009 he was attached to a temporary teaching and research at the French Institute of Advanced Mechanics (IFMA) in France. Since 2009, he was associate professor at the University of Technology of Troyes. These studies focused on linear and nonlinear programming and the exact (B&B and B&Cut methods) and approximate methods based on heuristics and meta-heuristics.

Hichem Snoussi received the diploma degree in electrical engineering from the Ecole Supérieure d'Electricité (Supélec), Gif-sur-Yvette, France, in 2000. He also received the DEA degree and the Ph.D. in signal processing from the University of Paris-Sud, Orsay, France, in 2000 and 2003 respectively. Between 2003 and 2004, he was post-doctoral researcher at IRCCyN, Institut de Recherches en Communications et Cybernétiques de Nantes. He has spent short periods as visiting scientist at the Brain Science Institute, RIKEN, Japan and Olin Neuropsychiatry Research Center at the Institute of Living in USA. Between 2005 and 2009, he was associate professor at the University of Technology of Troyes, France. He has obtained the HDR degree from the University of Technology of Compiègne in 2009. Since 2010, he is Full Professor at the University of Technology of Troyes. His research interests include Bayesian techniques for source separation, information geometry, differential geometry, machine learning, robust statistics, with application to brain signal processing, astrophysics, and advanced collaborative signal/image processing techniques in wireless sensor/cameras networks. Since January 2011, he is leading the research group "Risk Management of Complex Systems and Networks" of the CNRS STMR UMR Laboratory. He is in charge of the regional research program S3 (System Security and Safety) of the CPER 2007–2013 and the CapSec plateform (wireless embedded sensors for security). He is leading many funded projects (ANR, Oséo, GIS, strategic UTT programs, Carnot, CPER). In 2009, he launched a new company Track&Catch on smart embedded cameras for security and surveillance, where he is the scientific director. He is author of more than 100 research papers in journal and international conferences.