



HAL
open science

Optimizing Deployment Cost in Camera-Based Wireless Sensor Networks

Mehdi Rouan Serik, Mejdj Kaddour

► **To cite this version:**

Mehdi Rouan Serik, Mejdj Kaddour. Optimizing Deployment Cost in Camera-Based Wireless Sensor Networks. 5th International Conference on Computer Science and Its Applications (CIIA), May 2015, Saida, Algeria. pp.454-464, 10.1007/978-3-319-19578-0_37. hal-01789949

HAL Id: hal-01789949

<https://inria.hal.science/hal-01789949>

Submitted on 11 May 2018

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.



Distributed under a Creative Commons Attribution| 4.0 International License

Optimizing Deployment Cost in Camera-Based Wireless Sensor Networks

Mehdi Rouan Serik and Mejdi Kaddour

LITIO Laboratory, University of Oran 1, BP 1524, El-M'Naouer, 31000 Oran, Algeria
{rouan.mehdi,kaddour.mejdi}@univ-oran.dz

Abstract. We discuss in this paper a deployment optimization problem in camera-based wireless sensor networks. In particular, we propose a mathematical model to solve the problem of minimizing the number of cameras required to cover a set of targets with a given level of quality. Since solving this kind of problems with exact methods is computationally expensive, we rather rely on an adapted version of *Binary Particle Swarm Optimization* (BPSO). Our preliminary results are motivating since we obtain near-optimal solutions in few iterations of the algorithm. We discuss also the relevance of hybrid meta-heuristics and parallel algorithms in this context.

Keywords: Camera-based wireless sensor networks, minimum cost deployment, coverage quality, binary particle swarm optimization.

1 Introduction

Wireless Sensor Networks (WSN) are particular ad-hoc networks defined as a set of cooperating nodes disseminated in a given geographic area in order to collect its data about some phenomenon autonomously. Specifically, Camera-based wireless sensor networks (WSN) form an emerging research area with many promising applications. Potential applications include remote video surveillance, monitoring and assisting elderly and health patients, and habitat monitoring.

We study in this paper the cost deployment of camera-based WSNs, where the main concern is to determine the optimal minimum number of cameras, along with their positions and their orientations to track a given set of targets with a prescribed level of quality. This problem can be solved with various exact mathematical programming tools such as Branch and Bound. However, as the problem size increases, solving such problems using these exact methods becomes computationally intractable. In fact, Ai and Abouzeid [1] have demonstrated that this problem is NP-hard. A traditional way to deal with such difficult problems is to rely upon meta-heuristic methods. In particular, we adapt the generic procedure of *Particle Swarm Optimization* (PSO) to solve this deployment problem. We also introduce a quality coverage parameter which serves to ensure that each target in the solution is covered with a sufficient level of quality. Indeed, as the targeting objects move away from the sensing camera,

the level of details falls off. Our preliminary results show that the proposed method behaves well both in computational performances and solution quality.

The rest of paper is organized as follows. Section 2 reviews the relevant literature on this topic. Section 3 formulates the problem and describes the proposed coverage model. The detailed mathematical model is given in Section 4, while the proposed PSO-based algorithm is described in Section 5. Section 6 assesses and discusses the obtained results. Finally, The last section concludes the paper and suggests some future research directions.

2 Related Works

Existing literature in the field distinguishes typically two important kinds of deployment problems: target coverage and area coverage. We focus here on the first one. One version of the problem consists on covering a maximum number of targets with a minimum number of sensors. In [1], authors proposed an exact integer linear program (ILP) and a Centralized Greedy Algorithm (CGA) for the maximum coverage with minimum sensors (MCMS) problem. Then, they provided a Distributed Greedy Algorithm (DGA) solution. They showed that DGA does better than the two other methods (ILP and CGA) by incorporating a measure of the sensors' residual energy into DGA. Aziz *et. al.* [2] proposed a new algorithm to optimize sensor coverage using PSO and Voronoi diagrams. PSO is used to find the optimal deployment of sensors providing the best coverage, while Voronoi diagram is used to evaluate the fitness of the solution. They showed that the proposed algorithm achieves a good coverage with a better time efficiency than existing approaches.

Authors in [12] improved the field of view (FOV) coverage of a camera network. They considered randomly scattered cameras in a wide area, where each camera may adjust only its orientation and not its localisation. They also implemented a PSO algorithm and efficiently found an optimal orientation for each camera. They considered also region of interest in the search space (ROI) and occlusions. In [6], authors considered the area coverage problem in a 2D/3D-grid space. The solution process was based on Particle Swarm Optimization Inspired Probability (PSO-IP). For comparison purposes, they also implemented alternative methods based on Tabu Search, genetic algorithms and simulated annealing. Results showed that the proposed PSO-IP overcomes the three other methods especially with large instances. Unlike these two works, we deal with targets coverage not area coverage. We give also a mathematical model to the coverage problem. Note that our solution approach is partly inspired by this last work. But as far as we know, no existing work has formulated a problem similar to ours, in particular by considering a continuous measure of sensing quality.

3 Problem Definition

We assume that a set of N camera sensors $\{S_i : i = 1, \dots, n\}$ are deployed on the euclidean space A . For each camera S_i , we are given its Cartesian coordinates (X_{S_i}, Y_{S_i}) and its orientation φ in A . The field of vision of each camera is modeled as in [8] by isosceles triangle as depicted in Fig. 1, where α represents the aperture of the camera and W_d its working distance.

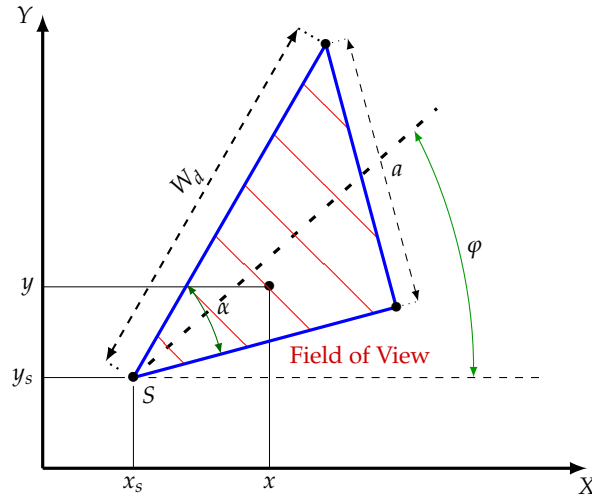


Fig. 1: Camera's field of view.

A target located on the coordinates (x, y) is assumed to be covered by a given camera with coordinates (x_s, y_s) if the three following constraints are satisfied:

$$\cos(\varphi) \cdot (x - x_s) + \sin(\varphi) \cdot (y - y_s) \leq W_d \quad (1a)$$

$$\begin{aligned} -\sin(\varphi) \cdot (x - x_s) + \cos(\varphi) \cdot (y - y_s) &\leq \\ \frac{a}{2W_d} \cdot (\cos(\varphi) \cdot (x - x_s) + \sin(\varphi) \cdot (y - y_s)) &\end{aligned} \quad (1b)$$

$$\begin{aligned} -\sin(\varphi) \cdot (x - x_s) + \cos(\varphi) \cdot (y - y_s) &\geq \\ -\frac{a}{2W_d} \cdot (\cos(\varphi) \cdot (x - x_s) + \sin(\varphi) \cdot (y - y_s)) &\end{aligned} \quad (1c)$$

3.1 Coverage Model

Coverage models determine first if a given target can be covered or not by some sensor, but can also measure a corresponding quality parameter [11]. This is ac-

complished by calculating the geometric relation between sensors and targets. In most cases, it consists in calculating the euclidean distance and angles, but some research works assume also that the sensing quality of a sensor is reduced with the increase of the distance away from the sensor [5], [10]. In our case, we make a similar assumption by adopting a directional disk model where the coverage parameter for a given sensor/target pair is represented by a non negative real number calculated as follows:

$$f(d(s, z)) = \frac{C}{d^\alpha(s, z)} \quad (2)$$

where $d(s, z)$ is the distance separating camera s from target z , α is the exponent attenuation and C is a constant. In particular, we assume that quality decreases quadratically as a function of the distance ($\alpha = 2$).

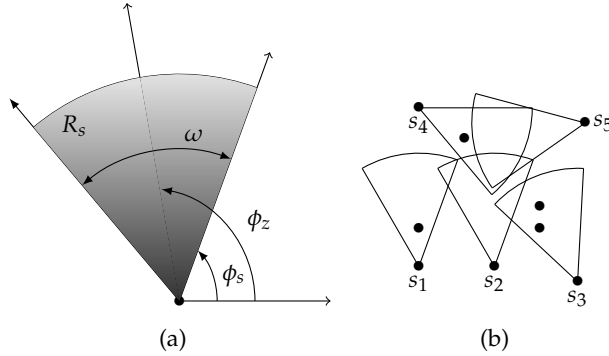


Fig. 2: Illustrations of : (a) Directional Model ; (b) Three active cameras covering 4 targets.

4 Mathematical Model

The adopted model assumes a known number of targets, in a given space, where the objective is to cover these targets optimally. Table 1 define the formal notations used in our model.

let b_{ij} and O_{ik} binary variables defined as follows:

$$b_{ij} = \begin{cases} 1 & \text{if camera at position } i \text{ covers target } j \\ 0 & \text{otherwise} \end{cases}$$

$$O_{ik} = \begin{cases} 1 & \text{if camera at position } i \text{ has active angle } k \\ 0 & \text{otherwise} \end{cases}$$

Table 1: Problem's formal notations.

| Variables | Designation |
|------------------|--|
| n | number of possible camera locations |
| m | number of possible angles |
| t | number of targets |
| δ | quality parameter |
| d_{ij} | distance between camera i and target j |
| (x_i, y_i) | Cartesian coordinate of the camera i |
| (x_j^t, y_j^t) | Cartesian coordinates of the target j |
| φ_k | angle k |

The objective of our optimization problem is defined as follows:

$$\min \sum_{i=1}^n \sum_{k=1}^m O_{ik} \quad (3)$$

This objective implies the minimization of the number of deployed cameras, provided that the following constraints are satisfied.

$$\sum_{k=1}^m O_{ik} \leq 1, \quad \forall i = 1, \dots, n. \quad (4)$$

The above constraint ensures that at most one angle is active per each camera. $\sum_{k=1}^m O_{ik} = 0$, corresponds to the situation where no camera is deployed at location i .

$$\sum_{i=1}^n b_{ij} \geq 1, \quad \forall j = 1, \dots, t. \quad (5)$$

This constraints ensures that each target is covered by at least one camera.

Now, the minimum coverage quality for each target is satisfied through the following constraint:

$$\sum_{i=1}^n b_{ij} \frac{C}{d_{ij}^\alpha} \geq \delta, \quad \forall j = 1, \dots, t. \quad (6)$$

where C is a constant and α is the attenuation exponent.

Given some locations of a camera i oriented with angle k , the following three constraints enforce, as described above in (1a),(1b) and (1c), that some target j is covered properly if $b_{ij} = 1$.

$$\sum_{k=1}^m O_{ik} \left[\cos(\varphi_k) \cdot (x_j^t - x_i) + \sin(\varphi_k) \cdot (y_j^t - y_i) \right] \leq d + L_1(1 - b_{ij}),$$

$$\forall i = 1, \dots, n, \quad \forall j = 1, \dots, t. \quad (7)$$

$$\begin{aligned} \sum_{k=1}^m O_{ik} \left[- \left(\sin(\varphi_k) + \frac{a}{2d} \cos(\varphi_k) \right) (x_j^t - x_i) + \left(\cos(\varphi_k) - \frac{a}{2d} \sin(\varphi_k) \right) (y_j^t - y_i) \right] \\ \leq L_2(1 - b_{ij}) \quad \forall i = 1, \dots, n, \quad \forall j = 1, \dots, t. \end{aligned} \quad (8)$$

$$\begin{aligned} \sum_{k=1}^m O_{ik} \left[\left(\sin(\varphi_k) - \frac{a}{2d} \cos(\varphi_k) \right) (x_j^t - x_i) - \left(\cos(\varphi_k) - \frac{a}{2d} \sin(\varphi_k) \right) (y_j^t - y_i) \right] \\ \leq L_3(1 - b_{ij}) \quad \forall i = 1, \dots, n, \quad \forall j = 1, \dots, t. \end{aligned} \quad (9)$$

where L_1, L_2, L_3 are large constants, which are introduced to make these constraints meaningless when $b_{ij} = 1$.

The above objective with the introduced constraints define a mixed-integer problem which is hard to solve in general. In particular, it is not easy to move from one solution to another when applying some meta-heuristic algorithm for example. Hence, we relax this model by moving the coverage constraints (5)-(6) into the objective, as follows.

$$\min \left\{ \sum_{i=1}^n \sum_{j=1}^m O_{ij} - \sum_{i=1}^n \sum_{j=1}^t b_{ij} \left(1 + \frac{C}{d_{ij}^x} \right) \right\} \quad (10)$$

Subject to (4),(7)-(9).

5 Particle Swarm Optimization

Particle swarm optimization (PSO) is a meta-heuristic method invented by *Russel Eberhart* (Electrical Engineer) and *James Kennedy* (socio-psychologist) in 1995. This algorithm, inspired by social behaviour, has been introduced as an optimization tool dealing with real numbers initially and with integers lately [7]. It is mostly inspired from the manner in which a flock of birds moves with various individuals leading the flock during the travel at different periods of time. The PSO algorithm consists of a group of individuals named particles. Each particle $p < \text{Swarm_size}$ is a potential solution to an optimization problem, having its own position in the space search. After each iteration, it moves in function of one of its components:

- Actual velocity V ;
- Best solution L_b ;
- Actual position X ;
- Best neighbourhood solution G_{best} ;

The movement of each particle obeys to the equations:

$$V_{k+1} = \omega \cdot V_k + C_1 r_1 (P_b - X_k) + C_2 r_2 (P_g - X_k) \quad (11)$$

$$X_{k+1} = X_k + V_{k+1} \quad (12)$$

where $X = (x_{ik}), V = (v_{ik}), i = 1, 2, \dots, n, k = 1, 2, \dots, m$, denote the distance/angle and the velocity vectors, respectively. ω is the inertia weight, r_1 and r_2 are two random numbers uniformly chosen in $[0, 1]$, and C_1, C_2 are constant values. Finally k is the iteration index.

A given solution for the deployment problem is a number of cameras with corresponding positions and active angles (see Fig. 3a). We adopt a binary representation of the position vector X . Besides, since PSOs require movements, we define two different types of moves:

1. Rotation: selecting a different camera for an active camera.
2. Displacement: moving one camera from position i to position i' .

These movements are guided by the velocity parameter. Thus, we redefine (11) as follows:

$$V = (v_{ik}, \forall k = 1, 2, \dots, m) = \begin{cases} 1 & \text{if } a_{lea} > \frac{1}{1+(C_{best}-L_b)} \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

Where a_{lea} is a random number from $[0, 1]$, then Eq. (12) become:

$$X_{k+1} = X_k \oplus V_{k+1} \quad (14)$$

A camera rotation is defined through the logical operator "exclusive OR", i.e., to invert a zero bit to one in the X vector, while a camera displacement is defined through a binary shift, and this will be a position swap. As a stopping criteria, we define a maximum number of iterations. Algorithm 1 gives an overview of our implemented PSO.

For comparison purposes, we implemented a standard *Simulated Annealing Algorithm* (SA) [9]. The basic idea behind comes from the principles of statistical mechanics whereby the annealing process requires heating and then slowly cooling a substance to obtain a strong crystalline structure. At each iteration, a random neighbour is generated. Movements that improve the cost function are always accepted. Otherwise, the neighbour is selected with a given probability that depends on the current temperature and the amount of degradation ΔE of the objective function. This probability is calculated as follows:

$$P(\Delta E, T) = e^{\frac{-\Delta E}{T}}$$

where ΔE represents the fitness difference. This algorithm is used in large-scale optimization problems in wireless sensors networks as in [6], [4].

Algorithm 1 Proposed PSO algorithm

```

1: for all ( $p < \text{Swarm\_size}$ ) do
2:    $\text{Random\_init}(p)$ ;
3:    $L_b(p) = \text{Fitness}(p)$ ;
4: end for
5:  $G_{best} = \min_p \{L_b(p)\}$ ;
6: repeat
7:    $\text{alea} = \text{rand}()$ ;
8:   for all ( $p < \text{Swarm\_size}$ ) do
9:     Calculate  $V(p)$  using (13);
10:     $\text{move}(p)$ ;
11:    if  $\text{Fitness}(p) < L_b(p)$  then
12:       $L_b(p) = \text{Fitness}(p)$ ;
13:    end if
14:  end for
15:   $G_{best} = \min_p \{L_b(p)\}$ ;
16:  Update positions with (14);
17: until Stop criteria satisfied
  
```

6 Experiments and Results

We discuss in this section various experiments related to our approach. First, the most important parameter to be defined in order to implement a PSO algorithm is the solution coding or representation. Each solution is encoded in binary where the vector X represents camera positions and angles (Fig 3a) and (Fig 3b). The experiments were executed on a computer with Intel © Core™ i3-2350 M CPU @ 2.30 GHz CPU and 4.0 GB of RAM.

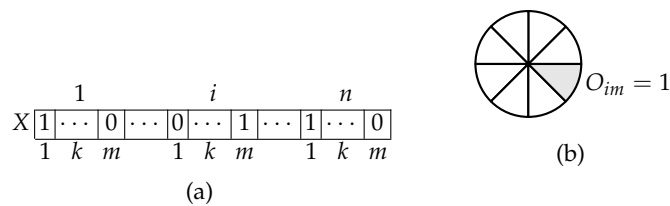


Fig. 3: PSO solution: (a) X Position vector; (b) Active angle.

Table 2a gives parameters used to implement the method. Swarm_size and Steps represent the number of particles (potential solutions) involved in PSO and iteration's number, respectively. C_1 and C_2 represent learning factors of the algorithm, most often set to 2.0 [3]. W_d represents the working distance of a camera (extrinsic parameter), X_{\max} and Y_{\max} are the grid dimensions. Q_{\max} and Q_{\min}

Table 2: PSO and SA parameters and executions.

| (a) | | (b) | | | | | |
|--------------------|---------|------|--------|-------|-------|-------|----------|
| Parameters | Values | Exec | Fit. | n_s | n_t | Steps | time (s) |
| Swarm_size | 30 | 1 | -7.018 | 2 | 9 | 200 | 0.706 |
| Steps | 300 | 2 | -7.050 | 3 | 10 | 167 | 0.566 |
| C_1, C_2 | 1.4 | 3 | -8.057 | 2 | 10 | 85 | 0.109 |
| n | 10 | 4 | -7.033 | 2 | 9 | 52 | 0.041 |
| m | 8 | 5 | -7.048 | 3 | 10 | 1 | 0.003 |
| t | 10 | 6 | -7.036 | 2 | 9 | 269 | 0.78 |
| L_1, L_2, L_3 | Max_DBL | 7 | -7.032 | 2 | 9 | 61 | 0.052 |
| C | 10.0 | 8 | -7.035 | 2 | 9 | 170 | 0.12 |
| Q_{max} | 0.4 | 9 | -7.058 | 3 | 10 | 131 | 0.509 |
| Q_{min} | 0.05 | 10 | -7.058 | 3 | 10 | 61 | 0.05 |
| W_d | 10 | | | | | | |
| X_{max}, Y_{max} | 100 | | | | | | |
| T | 1000.0 | | | | | | |

are, respectively, the maximum and the minimum quality coverage of a camera. Here we require a certain level of quality in target covering. C is a constant which specifies how the quality coverage decreases when moving away from the sensor.

In Fig 4, we give an initial solution then a final one found by PSO algorithm. The figures show that we can easily find a good solution consisting of two cameras covering nine targets. As shown in Tab 2b, finding such a solution does not require more than a hundred of iterations in a very brief time for a small number of targets (less than a half of a second). We observe also the high success rate over all executions. As shown in (Fig 4d) we clearly obtain better average fitness by the proposed PSO than the one obtained by the SA for fifty executions of the two methods.

Finally, note that compared to *Y. Morsly et. al.* approach in [6] and *Yi-Chun Xu et al.* in [12], we consider a coverage quality parameter and deal with targets instead of area coverage. Here we must determine camera's positions and orientation, while coverage quality can be specified by the user. Even if we assume mobile targets, the proposed method can easily re-adapt the solution to the new scenario; this is guaranteed by the camera's movement (displacement and rotation).

7 Conclusion

We have proposed a mathematical mixed-integer model for the deployment of camera-based wireless sensor networks. Solving such models with exact is only feasible with small problem instances, beyond which, the use of approximative method such as meta-heuristics is required. In literature, only a few papers

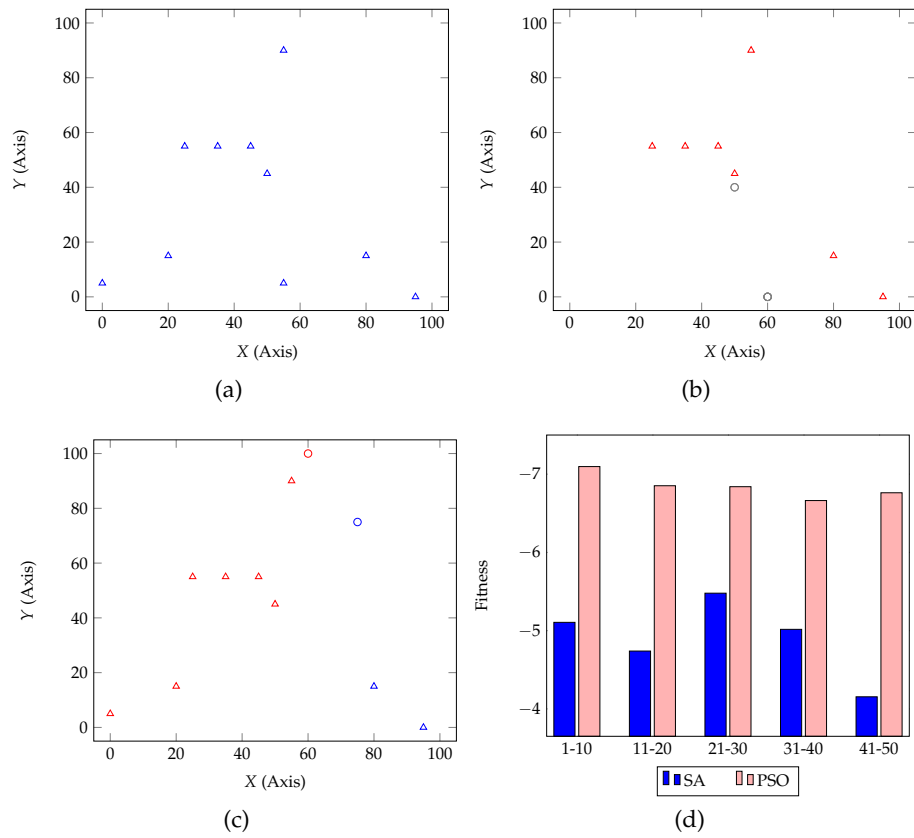


Fig. 4: Illustrations of: (a) Initial target positions; (b) Random initial solution; (c) Final solution found by PSO; (d) PSO vs SA.

treat the target-based coverage model along with a coverage quality parameter. Hence, we have introduced an additional problem parameter to account for coverage quality. We have adapted a PSO algorithm to minimize the number of active cameras used to cover targets with certain quality. The proposed method gives easily a significant improvement of the initial random solution in few steps. Compared to SA, our PSO returns better solutions. We envisage later to calculate exact solutions by a ILP solver and then compare the results with ours. Moreover, we are working to introduce connectivity and energy-efficiency criteria to our problem. Finally, we are considering to use parallel versions of PSO and evaluate their performance over multiple platforms.

References

1. Ai, J., Abouzeid, A.A.: Coverage by directional sensors in randomly deployed wireless sensor networks. *Journal of Combinatorial Optimization* 11(1), 21–41 (2006)
2. Aziz, N.A.B.A., Mohemmed, A.W., Alias, M.Y.: A wireless sensor network coverage optimization algorithm based on particle swarm optimization and voronoi diagram. In: *Networking, Sensing and Control, 2009. ICNSC'09. International Conference on*. pp. 602–607. IEEE (2009)
3. Gorse, D.: Binary particle swarm optimisation with improved scaling behaviour. In: *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (2013)
4. Kannan, A.A., Mao, G., Vucetic, B.: Simulated annealing based wireless sensor network localization with flip ambiguity mitigation. In: *Vehicular Technology Conference, 2006. VTC 2006-Spring*. IEEE 63rd. vol. 2, pp. 1022–1026. IEEE (2006)
5. Megeerian, S., Koushanfar, F., Qu, G., Veltri, G., Potkonjak, M.: Exposure in wireless sensor networks: theory and practical solutions. *Wireless Networks* 8(5), 443–454 (2002)
6. Morsly, Y., Aouf, N., Djouadi, M.S., Richardson, M.: Particle swarm optimization inspired probability algorithm for optimal camera network placement. *Sensors Journal, IEEE* 12(5), 1402–1412 (2012)
7. Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization. *Swarm intelligence* 1(1), 33–57 (2007)
8. Trucco, E., Umasuthan, M., Wallace, A.M., Roberto, V.: Model-based planning of optimal sensor placements for inspection. *Robotics and Automation, IEEE Transactions on* 13(2), 182–194 (1997)
9. Van Laarhoven, P.J., Aarts, E.H.: *Simulated annealing*. Springer (1987)
10. Veltri, G., Huang, Q., Qu, G., Potkonjak, M.: Minimal and maximal exposure path algorithms for wireless embedded sensor networks. In: *Proceedings of the 1st international conference on Embedded networked sensor systems*. pp. 40–50. ACM (2003)
11. Wang, B.: Coverage problems in sensor networks: A survey. *ACM Computing Surveys (CSUR)* 43(4), 32 (2011)
12. Xu, Y.C., Lei, B., Hendriks, E.A.: Camera network coverage improving by particle swarm optimization. *Journal on Image and Video Processing* 2011, 3 (2011)