A Subproblem-dependent Heuristic in MOEA/D for the Deployment and Power Assignment Problem in Wireless Sensor Networks

Andreas Konstantinidis, Qingfu Zhang and Kun Yang
The School of Computer Science and Electronic Engineering
University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, UK
{akonst, qzhang, kunyang}@essex.ac.uk

***Abstract*— In this paper, we propose a Subproblem-dependent Heuristic (SH) for MOEA/D to deal with the Deployment and Power Assignment Problem (DPAP) in Wireless Sensor Networks (WSNs). The goal of the DPAP is to assign locations and transmit power levels to sensor nodes for maximizing the network coverage and lifetime objectives. In our method, the DPAP is decomposed into a number of scalar subproblems. The subproblems are optimized in parallel, by using neighborhood information and problem-specific knowledge. The proposed SH probabilistically alternates between two DPAP-specific strategies based on the subproblems objective preferences. Simulation results have shown that MOEA/D performs better than NSGA-II in several WSN instances.**

I. Introduction

Most of the research in Wireless Sensor Network (WSN) topology design focuses on deciding optimal (a) locations (deployment [1]) and (b) transmit power levels (power assignment [2]) of the sensor nodes to be deployed in an area of interest. Several approaches have been proposed for the deployment and power assignment problems in WSNs, with major goal on maximizing (i) the coverage [3], or (ii) the lifetime [4] objective, respectively. However, few attempts have been made for simultaneously tackling both (a) and (b) decision variables, considering both (i) and (ii) objectives [5]. Even though, the latter approaches optimize the objectives individually, or by combining them into a single objective, or constraining one and optimizing the other, which often results on ignoring and losing "better" solutions. The coverage and lifetime of WSNs are conflicting objectives and warrant a trade-off. Hence, we have recently proposed the Deployment and Power Assignment Problem (DPAP) [6] in WSNs, as a Multiobjective Optimization Problem (MOP).

There are many methods for dealing with MOPs in the literature [7], with the Multi-Objective Evolutionary Algorithms (MOEAs) being a promising approach. General MOEAs, usually tackle a MOP as a "black box" [8], i.e. without any problem specific knowledge. This might be a drawback for MOEAs when dealing with real life problems (such as DPAP) having undesirable effects, e.g. force the evolutionary process into unnecessary searches, negatively affecting their performance. Thus, the incorporation of problem specific knowledge in MOEAs [9], to direct the search into promising areas of the search space, can be proven beneficially [10]. However, designing problem specific heuristics for a MOP as a whole is difficult. The Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [11] overcomes this difficulty by decomposing the MOP into many scalar subproblems and optimizing them simultaneously, by using neighborhood information and single-objective techniques. The difficulty, however, on adding knowledge on a decompositional MOEA is that the subproblems have different objective preferences, require different treatment and have to be solved simultaneously, in a single run. Therefore, the problem-specific heuristics should adapt to the requirements and objective preference of each subproblem dynamically, during the evolution.

In [6], we have briefly introduced our problem specific MOEA/D approach. This work concentrates on the improvement part of MOEA/D and proposes two improvement strategies, each focuses on one objective of DPAP. A Subproblem-dependent Heuristic (SH) is then adopted, to probabilistically apply the two improvement strategies to the solution of each subproblem and to strategically direct the search into promising areas of the search space. The goal is to improve the MOEA/D's performance in terms of diversity and quality of the PF for the DPAP. In Section II, we briefly introduce the MO-DPAP. Section III analyzes the problem by classifying the non-dominated solutions based on their objective preferences. The MOEA/D is briefly introduced in Section IV, followed by the proposed SH in Section V. The results of Section VI show an increase on the performance of MOEA/D and its superiority against the widely used Non-dominated Sorting Genetic Algorithm-II (NSGA-II [12]). The paper ends with some concluding remarks.

II. Problem Definition

A. System Model

Consider a 2-D static WSN formed by: a rectangular sensing area A, N homogeneous sensors and a static sink H with unlimited energy, placed at the center of A. The sensors are responsible to monitor and periodically report an event of interest to H. Each sensor i communicates directly or via multiple hops through nearby sensors with H, through the path loss communication model as in [13]. In this model, the transmit power level that should be assigned to a sensor i to reach a sensor j is $P_i = \beta \times d_{ij}^{\alpha}$, where $\alpha \in [2, 6]$ is the path loss exponent and $\beta = 1$ is the transmission quality parameter. The energy loss due to channel transmission is d_{ij}^{α}, d_{ij} is the Euclidean distance between sensors i, j and $R_c^i = d_{ij}$ is i's communication range. The calculated power assignments are considered static for the whole network's lifetime. The residual energy of sensor i, at time t, is calculated as follows:

$$E_i(t) = E_i(t-1) - [(r_i(t) + 1) \times P_i \times amp] \quad (1)$$

where $r_i(t) + 1$ is the total traffic load that sensor i forwards towards H at t ($r_i(t)$ is the traffic load that i relays and "$+1$" is the data packet generated by i to forward its own data information) and amp is the power amplifier's energy consumption. We assume that, the sensor nodes communicate through long transmission distances and therefore the transmit power consumption is a major factor on their total energy consumption [13]. Therefore, the energy consumed by the transceiver electronics, as well as, for reception and generation of data are considered negligible and ignored.

For sensing purposes and simplicity, we assume that A is composed by rectangular grids of identical dimensions centered at (x', y') and we adopted a "binary" sensing model [3]. Namely, a grid at (x', y') is covered, denoted by $g(x', y') = 1$, if it falls within a sensor's sensing range R_s, otherwise $g(x', y') = 0$. We consider unit-size grids, which are several times smaller than R_s, for a more accurate placement [3].

B. Problem formulation

The DPAP can be formulated as a MOP,

Given:

- A: 2-D plane of area size $x \times y$.
- N: number of sensors to be deployed in A.
- E: initial power supply, the same for all sensors.
- R_s: sensing range, the same for all sensors.

Decision variables:

- L_j : the location of sensor j.
- P_j : the transmission power level of sensor j.

Objectives: Maximize coverage $Cv(X)$ and lifetime $L(X)$ of network design X:

The network coverage $Cv(X)$ is defined as the percentage of the covered grids over the total grids of A and is evaluated as follows:

$$Cv(X) = [\sum_{x'=0}^{x} \sum_{y'=0}^{y} g(x', y')]/(x \times y) \quad (2)$$

where, $x \times y$ is the total grids of A and
$g(x', y') = \begin{cases} 1 & \text{if } \exists j \in \{1, ..., N\},\ d_{(x_j, y_j),(x', y')} \le R_s \\ 0 & \text{otherwise} \end{cases}$
is the monitoring status of the grid centered at (x', y').

The network lifetime is defined as the duration from the deployment of the network to the cycle t a sensor j depletes its energy supply, E. The lifetime objective $L(X)$ is evaluated as follows:

Algorithm: Lifetime Evaluation

Step 0: Set $t := 1$; $E_j(0) := E$, $\forall j \in \{1, ..., N\}$;

Step 1: **For** all sensors j at each time interval t **do**

Step 1.1:Update $E_j(t)$ according to Eq. 1;

Step 1.2:Assign each incoming link of sensor j a weight equal to $E_j(t)$;

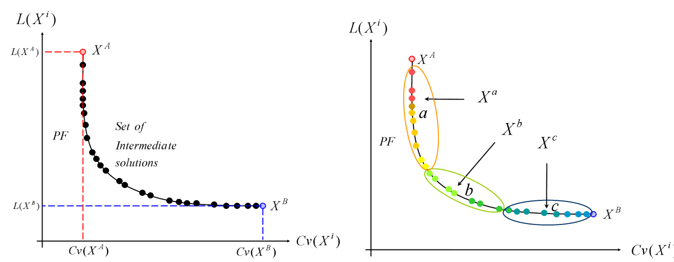
Step 1.3:Calculate the shortest path from j to H;

Step 2: **If** $\exists\ j \in \{1, ..., N\}$ such that $E_j(t) = 0$ **then** stop and set:

$$L(X) := t; \quad (3)$$

Else $t = t + 1$, go to step 1;

The same algorithm can be easily modified to consider different energy models and routing algorithms (e.g. geographical-based [14] routing algorithms).



(a) The PF of DPAP

(b) Various pareto optimal solutions of the PF

Fig. 1. Classifying the optimal network designs in DPAP

III. Problem Analysis

In multi-objective DPAP, there is not a unique network design, which can satisfy all objectives at the same time. Each design provides a preference to a particular objective. The Pareto optimal network designs that are close in the objective space should have similarities with each other in the decision space. In this work, we have identified two extreme optimal network designs X^A and X^B, which are dedicated on one objective each. Thereinafter, the intermediate solutions can be designed based on some network concepts and their position in the objective space (i.e. objective preference). The extreme Pareto optimal solutions and the set of intermediate solutions (Figure 1(a)) can be characterized as follows:

- **Solution X^A:** Dedicated on increasing the network's lifetime performance fully or highly ignoring the network's coverage quality.
- **Solution X^B:** Dedicated on increasing the network's coverage quality, fully or highly ignoring the network's lifetime performance.
- **Set of intermediate solutions:** A set of solutions providing the trade offs between the network lifetime and coverage objectives.

The optimal solution X^A provides the maximum lifetime among all the solutions in the PF,

$$L(X^A) = \frac{E}{d_{min}^{\alpha} \times amp},$$

where d_{min} is a controllable parameter, indicating the minimum distance allowed between a sensor node and H. Thus, a dense deployment of all sensor nodes around H with minimum transmission distances $R_c^i = d_{min}$ and direct communications with H (resulting on $r_i(t) = 0$) is desirable. Note that, $L(X^A)$ is used for normalizing the lifetime objective for the rest of this work, with the value 1 representing the best possible fitness. Moreover, since X^A is dedicated on $L(X^A)$, the $C(X^A)$ should be the minimum among all Pareto optimal solutions in the PF. It is desirable, however, to achieve the highest possible $Cv(X^A)$, which can be equal to

$$Cv(X^A) = A'/(x \times y),$$

where $A' \approx (2 \times (R_s + d_{min}))^2$.

The optimal solution X^B provides the maximum coverage among all the solutions in the PF. $Cv(X^B)$ highly depends on N. In this paper, we assume a spread like deployment, hence, let $N \leq \frac{(x \times y)}{(2R_s)^2}$ be small. Therefore, the sensor nodes should be deployed regularly, with a fixed distance $2R_s$ between each other and H, avoiding any sensing range overlaps. The maximum coverage can be calculated as follows:

$$Cv(X^B) = \frac{N \times \pi R_s^2}{(x \times y)}.$$

Similarly to X^A, achieving a network design X^B with the highest $L(X^B)$ is always desirable,

$$L(X^B) = \frac{E}{k \times (N/4) \times (2R_s)^\alpha \times amp},$$

where $k \times (N/4) \times (2R_s)^\alpha \times amp$ is the energy consumption of each sensor i that is directly connected to H at each t, and $N/4 \times k$ is a fixed minimum number of packets of size k (i.e. the traffic load) that should be burden by each sensor node i, assuming a regular, symmetrical deployment.

The goal of DPAP, however, is to provide the interested users with a diverse set of network design choices, giving the trade offs between the extreme optimal network designs X^A and X^B. However, the procedure of designing the intermediate topologies is complicated, since there is not a scalar method that can design all of them, in a single run. In the following, we introduce some general concepts for designing non-dominated solutions in different areas (e.g. a,b, and c in Figure 1(b)) of the intermediate set of solutions (Figure 1(a)):

- **Solution X^a:** favors a high network lifetime. Hence, the focus is to provide dense network designs by placing the sensor nodes with near to minimum transmission distances close to H. This, however, leads to high sensing range overlaps and poor coverage.
- **Solution X^c:** favors a high network coverage. Therefore, the focus is to provide spread network designs by placing the sensor nodes with high transmission ranges and low sensing range overlaps between (a) the sensor nodes and (b) the sensor nodes and the area's boundaries. This, however, leads to a high energy consumption of each sensor node at each t, which results to a poor lifetime.

Note that, it is expected that the interrelation of solutions X^a and X^c with the foresaid network concepts, "fades" as they get closer to the center of the PF. Thereinafter, a combination of these concepts could provide a balance on the DPAP's objectives as follows:

- **Solution X^b:** The sensor nodes are connected in such a way that their transmission power decrease/increase, and the sensing range overlaps increase/decrease, as they get closer to H, according to a slight preference on the lifetime or coverage objective, respectively.

Note that, when $N > \frac{(x \times y)}{(2R_s)^2}$ is high, the sensor nodes can be deployed more densely, to provide a $Cv(X^B) = 1$ by allowing some sensing range overlaps, e.g. with a fixed $\frac{2R_s}{\sqrt{2}}$ distance between each other and the area's boundaries. There are also scalar techniques that provide a higher $L(X^B)$ by utilizing a higher N, such as Chen's et al. approach [5].

IV. Brief introduction on MOEA/D

The MOP can be decomposed into m subproblems by adopting any technique for aggregating functions [11], e.g. the Weighted Sum Approach used here. Let λ^i be a weight that its associated subproblem i can be defined as:

$$max\ g^i(X^i|\lambda^i) = \lambda^i L(X^i) + (1 - \lambda^i) Cv(X^i)$$

Initially, the Internal Population, IP, which stores the best solutions found for each subproblem during the search, is randomly initialized. The genetic operators (i.e. selection, crossover and mutation) are then invoked on IP for offspring reproduction, X^i, for each subproblem i, where $i = 1$ to m. Moreover, problem specific heuristics are applied to improve each X^i and obtain Y^i. The update phase of MOEA/D is processed in three steps. (1) Update IP, $IP/\{X^i\}$ and $IP \cup \{Y^i\}$ if $g_i(Y^i|\lambda^i) > g^i(X^i|\lambda^i)$, otherwise X^i remains in IP. (2) Update the neighborhood of Y^i, i.e. the solutions of the T closest subproblems of i in terms of their weights $\{\lambda^1, \cdots, \lambda^m\}$ are updated. If $g^j(Y^i|\lambda^j) > g^j(X^j|\lambda^j)$, then $IP/\{X^j\}$ and $IP \cup \{Y^i\}$, otherwise X^j remains in IP, where $j \in \{1, ..., T\}$. (2) Update the External Population (EP), which stores all the non-dominated solutions found so far during the search. $EP = EP \cup \{Y^i\}$ if Y^i is not dominated by any solution $X^j \in EP$ and $EP = EP/\{X^j\}$, for all X^j dominated by Y^i. The search stops after a pre-defined number of generations, gen_{max}.

One of the main advantages of MOEA/D is that, appropriate scalar strategies can be adapted specifically to each subproblem i. Traditionally, it is hard to design an operator and/or heuristic to benefit all subproblems, since they have different objective preferences and they have to be solved simultaneously, in a single run. In order to overcome this difficulty, we have developed problem specific operators [6] and heuristics rising by each subproblem i's preference (weight coefficient λ^i) and adapted to its requirements. The λ^i parameter is used as a guide to the operators and heuristics for adjusting the degree of coverage and lifetime, and therefore designing different preference WSNs. MOEA/D proceeds as follows:

Input: • network parameters (A, N, E, R_s);
- m : population size and number of subproblems;
- T: neighborhood size;
- uniform spread of weights $\lambda^1, ..., \lambda^m$;
- the maximum number of generations, gen_{max};

Output: • the external population, EP.

Step 0-Setup: Set $EP := \emptyset$; $gen := 0$; $IP := \emptyset$;

Step 1-Decomposition: Initialize m subproblems, i.e. max $g^i(Z^i|\lambda^i)$, for $i = 1, ..., m$.

Step 2-Initialization: Randomly generate an initial internal population $IP = \{Z^1, \cdots, Z^m\}$;

Step 3: **For** each subproblem $i = 1$ to m **do**

Step 3.1-Genetic Operators: Generate a new solution X^i by using selection, crossover and mutation operators.

Step 3.2-Improvement: Apply a problem specific repair/improvement heuristic on X^i to produce Y^i.

Step 3.3-Update Populations: Update IP, EP and the T closest neighbors of subproblem i with Y^i.

Step 4-Stopping criterion: If stopping criterion is satisfied, i.e. $gen = gen_{max}$, then stop and output EP, otherwise $gen = gen + 1$, go to **Step 3**.

We refer interested readers to [11] for details. In this paper, the focus is on the improvement Subproblem-dependent Heuristic (SH) for incorporating problem specific knowledge to MOEA/D and producing near optimal network designs for the DPAP in WSNs.

V. The Subproblem-dependent Heuristic (SH)

The SH is composed by two simple single-objective strategies. Each strategy is based on a network concept related to an objective of DPAP and provides different treatment to the solutions of each subproblem i. Particularly $ImpL(X^i)$ benefits $L(X^i)$ and $ImpCv(X^i)$ benefits $Cv(X^i)$. The λ^i coefficient of a particular i shows a preference on one of the two objectives (except in the case where $\lambda^i = 0.5$). Therefore, we can probabilistically adapt a problem specific strategy to selectively improve a solution X^i. This is achieved by uniformly randomly generating a number $rand \in [0, 1]$, comparing it with the λ^i weight coefficient of each subproblem i and applying an improvement strategy accordingly.

Algorithm 1 The Subproblem-dependent Heuristic (SH)

Input: X^i, λ^i
Output: Y^i
Step 1: Run the energy-efficient GRP on solution X^i.
Step 2: Generate a uniform random number $rand \in [0, 1]$.
Step 2.1 If $rand < \lambda^i$ then

$$Y^i \leftarrow ImpL(X^i)$$

Step 2.2 Else

$$Y^i \leftarrow ImpCv(X^i)$$

The proposed SH, illustrated in Figure 2, works as in Algorithm 1. Note that, a new $rand$ is obtained in each generation for each subproblem, hence, the intermediate subproblems, which prefer a balance between the two objectives (e.g. $\lambda^i = 0.5$) have a high probability to be tackled by both improvement strategies, in different generations, and to design balanced topologies, such as solution X^b. Besides, subproblems with high or low λ coefficient still have some probability to be improved by $ImpCv(X^i)$ and $ImpL(X^i)$, respectively. In the following we analyze the two improvement strategies separately.

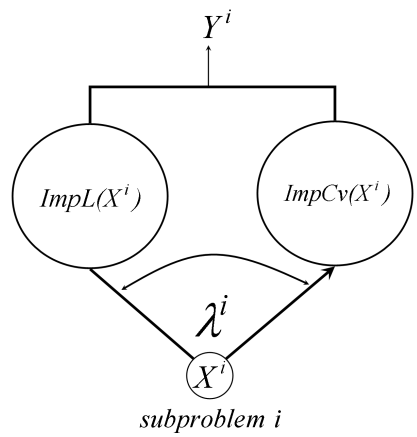


Fig. 2. The main concept of the Subproblem-dependent Heuristic (SH)

Improve lifetime: Following the analysis of section III, the $ImpL(X^i)$ improvement strategy is introduced. The goal of $ImpL(X^i)$ is to densely deploy the sensor nodes around H and decrease their transmit power levels as they get closer to H. This method mainly favors the solutions of areas a and b.

A sensor node j at location L_j is moving towards its one-hop neighbor h a distance $shift$, which depends on:

- the current energy consumption of sensor j, i.e. $(r_j(t) + 1) \times P_j \times amp$.
- the energy consumption of sensor k at location L_k, which considers sensor j as its one-hop neighbor node, i.e. $(r_k(t) + 1) \times P_k \times amp$

such that sensors j and k deplete their energy supply approximately at the same time.

Let $\overline{r_j}$ and $\overline{r_k}$ be the average traffic load of sensors j and k during the network's lifetime, respectively, and $a = 2$. Firstly, the required distance d'_{jh} between sensors j and h, such that sensors j and k deplete their energy supply approximately at the same time, is calculated as follows:

$$d'_{jh} = \sqrt{\frac{d_{kj}^2 \times amp \times \overline{r_k}(t)}{amp \times \overline{r_j}(t)}} \tag{4}$$

Consequently, the shift that sensor j should move towards h is equal to:

$$shift = d_{jh} - d'_{jh} \tag{5}$$

Thereinafter, the new location of sensor j is:

$$L'_j = L_j + shift \times (L_h - L_j)/d_{jh} \tag{6}$$

The transmit power levels of sensor j and k are then updated as follows:

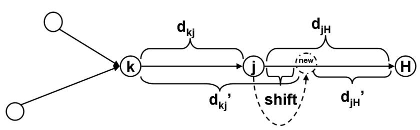
$$\begin{aligned} P'_j &= \sqrt{(x'_j - x_h)^2 + (y'_j - y_h)^2} \\ P'_k &= \sqrt{(x'_j - x_k)^2 + (y'_j - y_k)^2} \end{aligned} \tag{7}$$

where should satisfy $P'_j \times amp \times \overline{r_j} = P'_k \times amp \times \overline{r_k}$ and $d'_{jh} < d_{jh}$ such that $P'_j < P_j$.

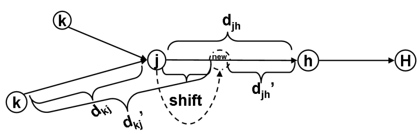
The $ImpL(X^i)$ strategy works as in Algorithm 2. The procedure of $ImpL(X^i)$, as well as, some special cases are illustrated in Figure 3. Note that, when j is directly connected to H, then $h = H$ and when j is at the end of the network, then the shift is fixed and equals to $d = R_c^h$.

Algorithm 2 $ImpL(X^i)$

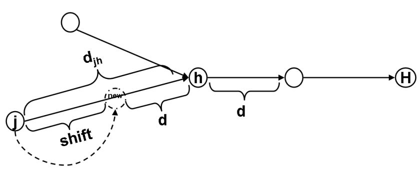
Input: Solution X^i;
Output: Improved solution Y^i;
For $j = 1$ to N **do**
- **Step 1:** Calculate the distance d'_{jh} using Eq. 4;
- **Step 2:** Subtract d'_{jh} from the initial distance d_{jh} between j and h to calculate the $shift$ towards sensor h, as in Eq. 5;
- **Step 3:** Calculate and set the new location L'_j to sensor j using Eq. 6;
- **Step 4:** Update the transmit power level of j and k using Eq. 7;



(a) Case 1 - Sensor connected to sink



(b) Case 2 - Sensor connected to another sensor



(c) Case 3 - Sensor at the network's boundaries

Fig. 3. An example of $ImpL(X^i)$

Improve coverage: The improvement strategy that benefits the coverage objective (i.e. $ImpCv(X^i)$) follows the analysis of Section III as well. This particular strategy, however, mainly favors solutions such as X^b and X^c of the objective space, since $ImpCv(X^i)$ decreases (a) the sensing range overlap between the sensor nodes by increasing the distance between them and (d) the sensing range overlaps between the sensor nodes and the area's boundaries, as required by the concepts introduced in Section III for X^b, X^c.

In $ImpL(X^i)$, a sensor node k at location (L_k) is shifted backwards from its one-hop neighbor node j a distance $shift$, to decrease the sensing range overlap between them. The sensing range overlap between sensors k, j denoted as $A_o(k,j)$, is equal to:

$$A_o(k,j) = R_s^2(q - sin(q))$$

where $q = 2 \times acos(d_{kj}/2R_s)$. Hence, by increasing d_{kj} the $A_o(k,j)$ between k and j decreases. Note that, for $d_{kj} = 2 \times R_s$ the $A_o(k,j) = 0$.

However, $ImpCv(X^i)$ may force all subproblems i with low λ^i to converge into a single solution, i.e. X^B giving $Cv(X^B)$. This is undesirable, since we need the objectives trade offs, i.e the solutions between the extreme solutions X^A, X^B. Hence, the new position of sensor k should be calculated in such a way that the sensing range overlap between k and j is decreased and the current network lifetime is maintained. Let $\overline{r_k}$, $\overline{r_j}$ be the average relay data information of sensors k and j respectively and $a = 2$.

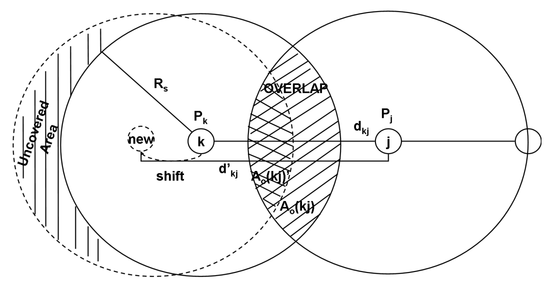


Fig. 4. An example of the first part of $ImpCv(X^i)$

Firstly, we calculate the required distance, d'_{kj}, such that sensors k and j deplete their energy supply approximately at the same time:

$$d'_{kj} = \sqrt{\frac{P_k \times amp \times \overline{r_k}}{amp \times \overline{r_j}}} \tag{8}$$

Then, we calculate the shift that sensor k moves backward from j:

$$shift = d'_{kj} - d_{kj} \tag{9}$$

Consequently, the new location of sensor k is:

$$L'_k = (L_k \times d'_{k,j} - shift \times L_j)/(d'_{k,j} - shift) \tag{10}$$

The new transmission power level of sensor k is:

$$P'_k = \sqrt{(x'_k - x_j)^2 + (y'_k - y_j)^2} \tag{11}$$

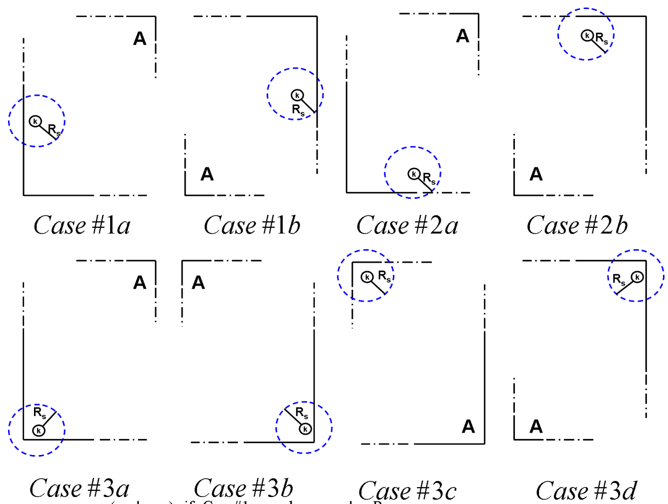
The final location and power assignment of sensor k should satisfy $P_j \times amp \times \overline{r_j} = P'_k \times amp \times \overline{r_i}$ and $d'_{kj} > d_{kj}$ such that $A_o(kj)' < A_o(kj)$.

- When sensor k has many one-hop neighbor nodes, then j is the one with the smallest d_{kj} and consequently the largest $A_o(kj)$.

This first part of $ImpCv(X^i)$ is illustrated in Figure 4.

Thereinafter, $ImpCv(X^i)$ decreases the sensing range overlaps between the sensor nodes and the area's boundaries. Assuming that the area is a rectangle, there are three different cases where a sensor violates the area's boundaries:

Case #1: Violation on x-axis: (a) left or (b) right bound.
Case #2: Violation on y-axis: (a) lower or (b)upper bound.
Case #3: Violation on both axes: (a) lower/left, (b) lower/right, (c) upper/left, (d) upper/right.



$$L_k' = \begin{cases} (x_k', y_k) \text{ if Case\#1a, where } x' = R_s \\ (x_k', y_k) \text{ if Case\textbackslash\#1b, where } x_{k'} = x - R_s \\ (x_k, y_k') \text{ if Case\#2a, where } y' = R_s \\ (x_k, y_k') \text{ if Case\#2b, where } y' = y - R_s \\ (x_k', y_k') \text{ if Case\#3a, such that } d_{((0,0),(x_k', y_k'))} = \sqrt{(0 - x_k')^2 + (0 - y_k')^2} \\ (x_k', y_k') \text{ if Case\#3b, such that } d_{((x,0),(x_k', y_k'))} = \sqrt{(x - x_k')^2 + (0 - y_k')^2} \\ (x_k', y_k') \text{ if Case\#3c, such that } d_{((0,y),(x_k', y_k'))} = \sqrt{(0 - x_k')^2 + (y - y_k')^2} \\ (x_k', y_k') \text{ if Case\#3d, such that } d_{((x,y),(x_k', y_k'))} = \sqrt{(x - x_k')^2 + (y - y_k')^2} \end{cases}$$

Fig. 5. The cases of the boundaries violations and the relocation of each case in the second part of $ImpCv(X^i)$

If a sensor k in a location $L_k = (x_k, y_k)$ violates any of the Cases #1,#2,#3 is redeployed to L'_k as in Figure 5. Thereinafter, sensor k is assigned a P^i_k such that it reaches its closest neighbor. $ImpCv(X^i)$ works as in Algorithm 3.

Algorithm 3 $ImpCv(X^i)$

Input: Solution X^i;
Output: Improved solution Y^i;
For $k = 1$ to N do
Step 1: If j is k's next-hop neighbor and $j \neq H$ then
- **Step 1.1:**Calculate the distance d'_{kj} by using Eq. 8;
- **Step 1.2:**Calculate the backward $shift$ by using Eq. 9;
- **Step 1.3:**Use Eq. 10 to find L'_k of sensor k and update P^i_k as in Eq. 11;

Step 2: If k violates any of the Cases #1,#2,#3 then
- **Step 2.1:**Set k into a new location L'_k as in Figure 5;
- **Step 2.2:**Update P^i_k, such that it reaches its closest one-hop neighbor.

VI. Simulation results and discussion

The goal of our simulation studies is: 1) to demonstrate the effectiveness of our problem specific improvement strategies (i.e. ImpL and ImpCv) and to empirically show how the proposed SH takes advantage of the foresaid strategies and adapts to the subproblems requirements, 2) to show that the SH increases the performance of the conventional MOEA/D and 3) to test the strength of the improved MOEA/D against the NSGA-II in various network instances, giving the objectives trade offs and a variety of network design choices.

Table I shows various network instances in a 2^2 factorial design [15]. Network Instances (NIn) 1,2 and 3,4 are of different As and same density (i.e. N/A) and NIn 1,3 and 2,4 are with different densities in the same A. Additionally, in all simulations we have used the following network parameter settings: $a = 2$, $R_s = 10m$, $E = 5J$, $amp = 100pJ/bit/m^2$ [14], $maxR_c = 20m$, $d_{min} = 10m$; and the following MOEA settings: $m = 120$, $c_{rate} = 1$, $m_{rate} = 0.1$, $s_t = 10$ and $gen_{max} = 250$. MOEA/D also considers a $T = 2$.

TABLE I
Network Instances

Network Instances	A (m^2)	Density (N/A)
NIn1	10000	.0013 (N=13)
NIn2	40000	.0013 (N=52)
NIn3	10000	.005 (N=50)
NIn4	40000	.005 (N=200)

Figure 6 illustrates the results of MOEA/D adopting each improvement strategy separately and simultaneously through the SH. As can be seen, when the density of the network is low, the MOEA/D w/ImpCv provides a more diverse set of non-dominated solutions. In addition, when the network size is small, MOEA/D w/ImpCv obtains better results than MOEA/D w/ImpL for solutions of areas b, c and provides a better approximation towards the extreme solution X^B. On the other hand, MOEA/D w/ImpL outperforms MOEA/D w/ImpCv in areas a, b and obtains a better approximation towards the extreme solution X^A. Hence, it is difficult to say which heuristic is the best with respect all NIns. MOEA/D-SH obtaining solutions from both strategies PFs, provides a balanced diverse set of high quality solutions in all areas of the objective space and the best approximation towards both extreme solutions X^A and X^B in all NIns.

Thereinafter, Figure 7 shows the contribution of the subproblem-dependent heuristic. MOEA/D-SH increases the performance of MOEA/D, especially on the subproblems that require solutions around X^b, giving a similar or better approximation towards the extreme solutions X^A and X^B, in all network instances. For the networks with low density (i.e. NIn1 and NIn3), MOEA/D-SH provides an average coverage increase of 0.5% for the solutions in areas a and c with the same lifetime quality, and a simultaneous average increase of 1.5% lifetime and 1% coverage for those in area b. For the network instances with high density, MOEA/D-SH obtains non-dominated solutions of about 1.5% more average lifetime

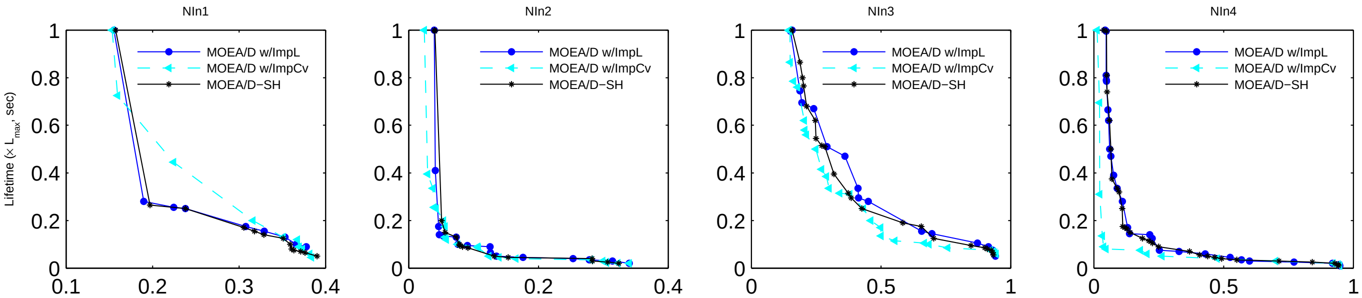


Fig. 6. MOEA/D with ImpL, ImpCv and SH in NI1

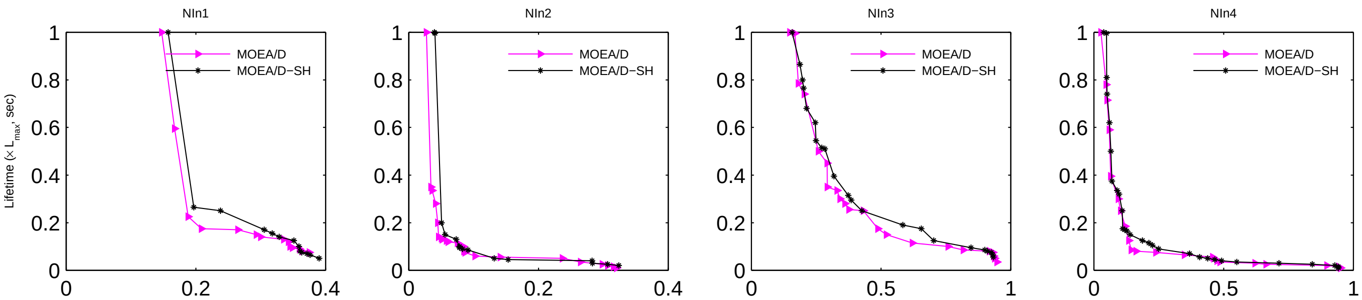


Fig. 7. MOEA/D with and without the SH in NI1

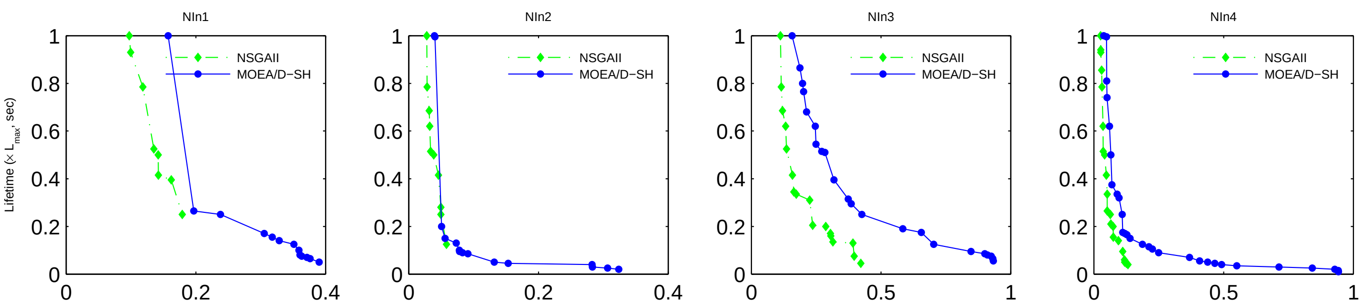


Fig. 8. MOEA/D vs. NSGAII in NI1-4

and the same coverage for area a, and about 1% more coverage and the same lifetime for area c. Moreover, it provides a simultaneous increase of 1.5% lifetime and 2% coverage, in average, for the solutions in area b, with respect those obtained by MOEA/D. Moreover, MOEA/D-SH obtains a better approximation towards X^A and a similar approximation towards X^B in all NIns.

For comparing the MOEAs, we have adopted various performance metrics that are usually employed for comparing sets of solutions obtained by different algorithms. The metrics are the $C(A, B)$ metric [12], which measures the solutions in an algorithm A's PF dominated by the solutions in an algorithm B's PF (i.e. the smaller $C(A, B)$ is, the better A is), the $\Delta(A)$ metric [12], which shows the diversity of the PF obtained by an algorithm A, i.e. the spread of the network design choices along the PF. $\Delta(A) = 0$ is the maximum, which means that the solutions are evenly spread along A's PF. A straightforward comparison metric between two sets of non-dominated solutions is the number of Non-Dominated Solutions NDS(A), i.e. the volume of network design choices, since in a real life problem (such as DPAP) is very difficult to obtain many NDS. Hence, a high NDS is desirable for increasing the decision maker's choices. Thus, the combination of the number of NDS with the C-metric and Δ-metric should be an adequate set of metrics to judge if an algorithm has obtained a large, diverse set of high quality solutions.

Figure 8 and Table II illustrates the superiority of the proposed MOEA/D method against NSGA-II. MOEA/D outperforms NSGA-II in all network instances in terms of quality of solution in the PF, number of NDS and in terms of diversity in dense network topologies. In network topologies with low density, NSGA-II provides a more uniform spread of solutions. Even though, the width of the PF covered by MOEA/D is more than the one obtained by NSGA-II, since NSGA-II lacks

TABLE II

MOEA/D VS. NSGAII NI1-4, AS8

Δ-metric	Δ(MOEA/D)	Δ(NSGAII)
NIn1:	0.9867	**0.8410**
NIn2:	0.9869	**0.8375**
NIn3:	**0.7271**	0.7877
NIn4:	**0.7262**	0.8219
Average:	0.8567	**0.8220**
NDS-metric:	NDS(MOEA/D)	NDS(NSGAII)
NIn1:	**13**	8
NIn2:	**15**	10
NIn3:	**22**	17
NIn4:	**29**	21
Average:	**19.7500**	14.0000
C-metric:	C(MOEA/D,NSGAII)	C(NSGAII,MOEAD)
NIn1	**0.0000**	1.0000
NIn2	**0.0000**	0.7000
NIn3	**0.0000**	1.0000
NIn4	**0.0000**	1.0000
Average:	**0.0000**	0.9250

TABLE III

ANALYTICAL SOLUTIONS X^A AND X^B AND THEIR APPROXIMATION BY THE EXTREME SOLUTIONS X^1 AND X^m OBTAINED BY MOEA/D AND NSGAII

NIn	Method	Lifetime $X^A \backslash X^1$	Coverage $X^A \backslash X^1$	Lifetime $X^B \backslash X^m$	Coverage $X^B \backslash X^m$
1	Analytical	1	0.16	0.00003	0.408
	MOEA/D	1	0.1574	0.05	0.3903
	NSGAII	1	0.097	0.25	0.1793
2	Analytical	1	0.04	7.69e^{-5}	0.3926
	MOEA/D	1	0.039	0.02	0.3239
	NSGAII	1	0.0272	0.125	0.05
3	Analytical	1	0.16	8e^{-5}	1
	MOEA/D	1	0.1572	0.055	0.9332
	NSGAII	1	0.1118	0.045	0.4221
4	Analytical	1	0.04	2e^{-5}	1
	MOEA/D	1	0.0377	0.01	0.9427
	NSGAII	1	0.0262	0.04	0.131

on obtaining solutions around X^c and obtains few solutions around X^b. MOEA/D dominates 92.50% of the solutions obtained by NSGA-II in average of all network instances, giving six additional non-dominated network design choices, in average. NSGA-II obtains a slightly better average diversity than MOEA/D, which sacrifices some of the diversity in the PF for the sake of better quality.

Table III summarizes the objective values of the optimal network designs X^A and X^B, which are analytically measured according to section III, and their approximation, i.e. the objective values of the extreme solutions X^1 and X^m obtained by each MOEA, for each network instance. The results show that MOEA/D approximates the optimal network designs more efficiently than NSGA-II. Another conclusion that can be empirically drawn is that, MOEA/D is not sensitive on the WSN's area size or density, giving similar results in each case. MOEA/D obtains a $L(X^1) = 100\%$, $Cv(X^1) = 15.74\%$, $L(X^1) = 100\%$, $Cv(X^1) = 15.72\%$ and $L(X^1) = 100\%$, $Cv(X^1) = 3.9\%$, $L(X^1) = 100\%$, $Cv(X^1) = 3.77\%$ for the same $10000m^2$ and $40000m^2$ area sizes, respectively, with different densities. Moreover, MOEA/D provides a $L(X^m) = 5\%$, $Cv(X^m) = 39.03\%$, $L(X^m) = 2\%$, $Cv(X^m) = 32.39\%$ and $L(X^m) = 5.5\%$, $Cv(X^m) = 93.32\%$, $L(X^m) = 1\%$, $Cv(X^m) = 94.27\%$ for the same 0.0013 and 0.005 densities, respectively, in different area sizes.

VII. CONCLUSIONS

In this work, a Subproblem-dependent Heuristic (SH) is proposed and successfully applied to MOEA/D for tackling the multi-objective DPAP in WSNs. Initially, the DPAP is decomposed and analyzed based on the subproblems objective preferences. Then, the SH is introduced, i.e. a probabilistic mixture of two DPAP-specific improvement strategies, each favoring one objective. Finally, simulation results have shown that the hybridization of the proposed SH with MOEA/D obtains better results than NSGA-II. A Generalized Subproblem-dependent Heuristic (GSH) is currently under investigation to deal with a constrained multiobjective DPAP in WSNs.

REFERENCES

[1] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Elsevier Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2007.

[2] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Computing Surveys*, vol. 37, no. 2, pp. 164–194, 2005.

[3] X. Xu and S. Sahni, "Approximation algorithms for sensor deployment," *IEEE Transactions on Computers*, vol. 56, no. 12, 2007.

[4] A. Konstantinidis, K. Yang, H.-H. Chen, and Q. Zhang, "Energy aware topology control for wireless sensor networks using memetic algorithms," *Elsevier Computer Communications*, vol. 30, no. 14-15, pp. 2753–2764, 2007.

[5] Y. Chen, C.-N. Chuah, and Q. Zhao, "Network configuration for optimal utilization efficiency of wireless sensor networks," *Elsevier Ad Hoc Networks*, vol. 6, pp. 92–107, 2008.

[6] A. Konstantinidis, K. Yang, and Q. Zhang, "An evolutionary algorithm to a multi-objective deployment and power assignment problem in wireless sensor networks," in *IEEE Global Communications Conference, GlobeCom08*, 2008.

[7] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley and Sons, 2002.

[8] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, April 2003.

[9] R. L. Becerra, L. V. Santana-Quintero, and C. A. C. Coello, "Knowledge incorporation in multi-objective evolutionary algorithms," in *Multi-Objective Evolutionary Algorithms for Knowledge Discovery from Databases*, 2008, pp. 23–46.

[10] N. Weicker, G. Szabo, K. Weicker, and P. Widmayer, "Evolutionary multiobjective optimization for base station transmitter placement with frequency assignment," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 189–203, April 2003.

[11] Q. Zhang and H. Li, "MOEA/D: A multi-objective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[13] X. Liu and P. Mohaparta, "On the deployment of wireless data backhaul networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 4, pp. 1426–1435, 2007.

[14] T. Melodia, D. Pompili, and I. F. Akyildiz, "On the interdependence of distributed topology control and geographical routing in ad hoc and sensor networks," *IEEE Journal on Selected Areas In Communications*, vol. 23, no. 3, pp. 520–532, 2005.

[15] D. C. Montgomery, *Design and Analysis of Experiments*, W. Anderson, Ed. John Wiley & Sons, 2001.