An Evolutionary Algorithm to a Multi-Objective Deployment and Power Assignment Problem in Wireless Sensor Networks

Andreas Konstantinidis, Kun Yang and Qingfu Zhang
Department of Computing and Electronic Systems
University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, UK
{akonst, kunyang, qzhang}@essex.ac.uk

***Abstract*— Wireless Sensor Networks design requires high quality location assignment and energy efficient power assignment for maximizing the network coverage and lifetime. Classical deployment and power assignment approaches optimize these two objectives individually or by combining them together in a single objective or by constraining one and optimizing the other. In this article a multi-objective Deployment and Power Assignment Problem (DPAP) is formulated and a Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) is specialized. Following the MOEA/D's framework the above Multiobjective Optimization Problem (MOP) is decomposed into many scalar single objective problems. The sub-problems are solved simultaneously by using neighborhood information. Additionally, unique problem-specific, parameter-rising, genetic operators and local search heuristics were designed specifically for the DPAP. In addition, a new encoding scheme is designed to represent a WSN based on the DPAP's design variables. Simulation results show that MOEA/D provides a high quality set of alternative solutions without any prior knowledge on the objectives preference.**

I. Introduction

Wireless Sensor Networks (WSNs)[1] design has received significant attention in the recent years due to the resource constraint nature of sensor node devices. There are two important requirements to consider while designing efficient WSNs: i.e. the efficient sensor node locations (deployment [2]) and power levels (power assignment [3]) for maximizing the WSN's coverage and lifetime respectively.

On one hand, most of the power assignment approaches focus on dynamically assigning homogeneous [4] or non-homogeneous [5] power levels to all sensors in the network. There are also cases where the power assignment is performed centrally/offline before the network startup [6], in which the deployment of the sensors is considered given or performed randomly. On the other hand, there are deterministic deployment approaches which fix or ignore the power assignment of the sensors [7]. Few approaches tackle these two issues simultaneously for maximizing the lifetime and coverage objectives. These approaches (e.g.[8]), however, 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.

Maximizing coverage and lifetime are conflicting objectives and thus warrant a trade-off [9]. Thereby, the problem should be formulated as a Multiobjective Optimization Problem (MOP) [10]; since all objectives are considered equal and there is not a single solution to optimize them at the same time (the set of Pareto optimal or non-dominated solutions is usually called Pareto Front (PF) [10]). Therefore, classical methods are not applicable in such complex and non-linear problems and the adaptation of Multi-Objective Evolutionary Algorithms (MOEAs [10]), which poses desirable characteristics for this type of problems, can be proven beneficially.

In this paper, we investigate the multi-objective deterministic pre-Deployment and Power Assignment Problem (DPAP). DPAP is typical in applications which invoke a limited number of expensive sensors, where their operation is significantly affected by their position and communication [11]. In these cases, the random deployment and dynamic power assignment is not the only choice and the applications afford the use of a centralized or even an off-line algorithm to compute the locations and transmit power levels of the sensors. Specifically, we investigate the following problem: for a given surveillance sensing field, determine the locations and the transmit power levels of a fixed number of sensor nodes, such that the network lifetime and coverage are maximized simultaneously. [9] formulated a similar MOP focusing on the trade-offs of coverage and lifetime objectives using the popular Multi-Objective Genetic Algorithm (MOGA). This problem can be considered as a subclass of our DPAP, since the authors focused on deployment only, fixing the transmission range of all sensors and affiliating a simplified energy model. Our approach is a hybrid MOEA utilizing the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) framework [12] and Local Search (LS) [6].

The main contributions of this paper are three-fold: 1) We have formulated the deterministic deployment and power assignment problem (DPAP) in WSNs as a MOP. 2) We have designed a unique encoding representation of a WSN topology and problem-specific, parameter-rising, population initialization, genetic operators and local search heuristics (including a power assignment) for the DPAP. 3) We have successfully specialized a MOEA/D-LS approach to DPAP and obtained near-optimal results. Simulations were conducted to show the performance superiority of the proposed approach.

II. Problem Definition

A. System Model

Consider a 2-D static wireless sensor network formed by: a rectangular sensing area A, a number of homogeneous sensors N and a static sink H with unlimited energy, placed at the center of A. The sensors are responsible for a simple task, i.e. monitoring and periodically reporting an event of interest to H. Hence, each sensor $i = 1, ..., N$, must be able to communicate (directly or via multiple hops through nearby sensors) with H. We assume a perfect medium access control and adopt the simple but relevant path loss communication model as in [8]. In this model the transmit power level P_i that should be assigned to a sensor i to reach a neighbor 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 and j and $R_c^i = d_{ij}$ is i's transmission range. The calculated power/range assignments are considered static for the whole network's lifetime. The total energy consumption of each sensor node per cycle is calculated based on the following energy model [13]:

$$E_t(k, d) = E_{tx}(k, d) + E_{rx}(k) + E_{or}(k) \quad (1)$$

where $E_{tx}(k, d)$, is the energy consumption to transmit a k-bit data packet a distance d and $E_{rx}(k)$, $E_{or}(k)$ is the energy consumption to receive and originate a k-bit data packet respectively.

For sensing purposes and simplicity, we assume a grid area A, composed by rectangular grids of identical dimensions, centered at (x', y'), and a binary sensing model [7]. A grid centered at (x', y') is considered covered, denoted by $l_{x'y'} = 1$, if it falls within a sensor's sensing range circle R_s. Otherwise $l_{x'y'} = 0$. We consider unit-size grids, which are several times smaller than R_s, for a more accurate placement [7].

B. Problem formulation

The DPAP can be formulated as a MOP as follows:
Given:

- A: 2-D plane of area size $[0, x_{max}] \times [0, y_{max}]$
- N: number of sensors to be deployed in A
- E: initial power supply, for each sensor $i = 1, ..., N$
- R_s: sensing range, for each sensor $i = 1, ..., N$

The design variables vector (X) is composed by the following **design variables**:

- (x_i, y_i) : the location of sensor i.
- P^i : the transmission power level of sensor i which highly depends on R_c^i, the transmission range of sensor i.

Objectives: Maximize coverage f_{cr} and lifetime f_{lf}:

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

$$f_{cr}(X) = [\sum_{x'=0}^{x_{max}} \sum_{y'=0}^{y_{max}} l_{x'y'}]/c_{to} \quad (2)$$

where, $c_{to} = x_{max} \times y_{max}$ is the total grids of the area and $l_{x'y'} = \begin{cases} 1 & \text{if } \exists i \in \{1, ..., N\},\ d_{i,(x',y')} \le R_s \\ 0 & \text{otherwise} \end{cases}$ is the coverage status of grid (x', y').

The network lifetime is defined as the percentage of the duration from the deployment of the network to the cycle τ an i^{th} sensor depletes its energy supply E over the maximum possible network runtime τ_{max}. The network's lifetime is τ_{max} when all sensors forward only a k-bit packet over a unit distance. The lifetime objective $f_{lf}(X)$ is evaluated as follows:

```
Algorithm: Lifetime Evaluation
Step 0: Set $\tau := 1$; $\tau_{max} = E/E_t(k, d_{min})$; $E_r^i(0) := E$,
        $\forall i \in \{1, ..., N\}$;
Step 1: Calculate the shortest path from each $i \in \{1, ..., N\}$ towards $H$ with incoming cost link equal
        to $E_r^i(\tau) = E_r^i(\tau - 1) - E_t^i(k, d)$;
Step 2: If $\exists\ i \in \{1, ..., N\}$ such that $E_r^i(\tau) = 0$ then
            $f_{lf}(X) := \tau/\tau_{max};$    (3)
        Else $\tau = \tau + 1$, go to step 1;
```

where $E_t(k, d)$ is given by equation 1 and $E_r^i(\tau)$ is the residual energy of each sensor $i = 1, ..., N$ in sensing cycle τ.

III. Our Multi-objective Evolutionary Computation approach

This section presents the general framework of MOEA/D as well as the motivation and the design of our operators and heuristics. MOEA/D proceeds as follows:

```
Input: • network parameters(e.g. $A$, $N$, $E$, $R_s$)
       • $m$ : population size and number of subproblems;
       • $T$: size of a neighborhood;
       • uniform spread of weight vectors $\lambda_0, ..., \lambda_m$;
       • a stopping criterion $gen_{max}$, which corresponds to
         the maximum number of generations;
Output: • the external population, $EP = \{X^*\}$.
Step 0-Setup: Set $EP := \emptyset$; $gen := 0$; $IP := \emptyset$;
Step 1-Decomposition: Initialize $m$ subproblems,
    $g_i(X^i|\lambda_i)$ , $i = 1, ..., m$.
Step 2-Initialization: Generate a solution $X^i, i = 1, ..., m$,
    by a problem-specific method. Evaluate $F(X^i)$. Initialize $IP = (X^1, ..., X^m)$;
Step 3: For $i = 1, ..., m$ do
        Step 3.1-Genetic Operation: Generate a new solution $X^i$ by using genetic operators.
        Step 3.2-Local Search: Apply repair, improvement heuristics on $X^i$ to produce $X^i$'.
        Step 3.3-Update Populations:Update populations
            $IP$, $EP$ based on $X^i$' and update the $T$ closest neighbors of $X^i$'.
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.
```

However, much work had to be done for the general-purpose MOEA/D framework to be suitable for the multi-objective deterministic DPAP in WSNs. Firstly, a dedicated encoding representation is designed to image a WSN for the DPAP. Secondly, problem-specific population initialization and genetic operators required for giving high quality topologies to each subproblem. Finally, local search heuristics are designed to further improve and/or repair the existing topologies quality and energy efficiency. In the following, we explain the procedure of the above algorithm in more details.

A. Decomposition and the main idea of design

The Weighted Sum Approach [12] is used for decomposing the proposed MOP into m subproblems. Let $\lambda = (\lambda^1, \lambda^2, ..., \lambda^z)$ be a weight vector, where z is the number of objectives, $\lambda^j \geq 0$ and $\sum_{j=1}^{z} \lambda_j = 1$. In this work, $z = 2$ and $(\lambda_i, 1-\lambda_i)$ is the weight vector for each subproblem i. A scalar optimization problem is defined as $Max.\ g_i(X|\lambda_i) = \lambda_i f_{lf} + (1 - \lambda_i) f_{cr}$, for $i = 1, ..., m$. So, considering the preference (weight coefficient λ_i) of each i, we can adapt the appropriate scalar strategies to optimize it specifically. For example, the extreme subproblem 1 is dedicated on maximizing f_{lf} and f_{cr} is fully ignored. Hence, for $\lambda_1 = 1$, it is defined as $Max.\ g_1(X|\lambda_1) = f_{lf}$. While λ decreases the subproblems are more favoring f_{cr}. The other extreme subproblem m is therefore defined as $Max.\ g_m(X|\lambda_m) = f_{cr}$, for $\lambda_m = 0$. It has to be noted that this beneficial procedure cannot be utilized by any non-decompositional MOEA framework.

Traditionally, it is hard to design an operator and/or a heuristic to benefit all subproblems, since they have different objective preference and they have to be solved simultaneously, in a single run. In this work, we have developed problem specific operators and heuristics rising by the preference parameter λ and adapted to the requirements of each subproblem. The λ 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. So all of our methods, which follow, have the same focus: when λ is high, produce dense topologies, decrease the network's latency (maximum number of hops en route) and increase the load balancing (using both the relation between the transmission range and number of data to transmit and multiple routes) favoring f_{lf}. As λ decreases, spread the sensors far away from H, favoring f_{cr}, giving the opportunity to the solutions to have sensors around H to prevent disconnections.

B. Encoding Representation

In this paper, we present a new chromosome representation of size N. Each chromosome, X, is composed by multi-part genes, the design variables of the problem, e.g. gene i is composed by (x_i, y_i) and P_i. The chromosome representation, shown in Figure 1, is called *"dense-to-spread"* encoding because of the way the chromosome is sorted. The nearer a sensor is to H the sooner is added in the chromosome. This results in having all the sensors densely deployed around H at the beginning of each chromosome and the sensors which are spread away at the end. This encoding representation will facilitate the problem specific operators and heuristics discussed later.

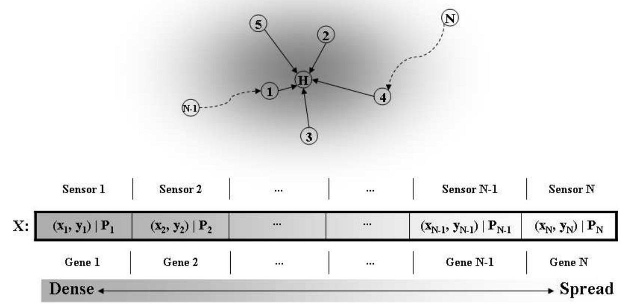


Fig. 1. Encoding Representation

C. Population Initialization

Population, IP_0, is an initial set of m solutions represented by their chromosome, produced in generation $gen = 0$. In this study, we designed a new problem specific population initialization which creates m solutions "seeded" in areas where optimal solutions are likely to be found specifically for each subproblem. We force the solutions of the subproblems which prefer high f_{lf}, i.e. high λ, densely deploy the sensors around H. Similarly, the solutions of subproblems with $\lambda \approx 1 - \lambda$ to be a mixture of sensors connected to H and spread around and finally, the solutions of subproblems more interested in f_{cr}, i.e. low λ, having most of the sensors spread around. IP_0, is then forwarded to the genetic operators.

D. Genetic Operators: Selection, Crossover, Mutation

Genetic operators [10] is the robust global searching tool of MOEAs. This section presents our problem-specific operators.

1) Selection: Selection operator [10] responsibly chooses high quality solutions from the current population to be included for reproduction in the next generation. In this paper, we have adopted a tournament selection-based operator [6], which is simple and fast. Our tournament selection has two major differences compared to conventional operators: (1) the solutions selected to compete in the tournament are the tm nearest neighbors of each subproblem i in IP, in terms of Euclidean distance of their λ value. (2) the neighbor solutions, e.g. j and z, are competing in i's tournament in terms of λ_i, ignoring their own λ_j and λ_z, their pareto domination and/or ranking [10]. The selected solutions, denoted as Pr_1^i and Pr_2^i parent solutions, are then forwarded to the crossover operator.

2) Crossover: Crossover is the operator which recombines Pr_1^i and Pr_2^i to produce one or more offspring, e.g. O_i. Our problem-specific crossover operator, namely *"window"-based crossover* works as follows:

For each subproblem i, the two parents are merged, $Pr_{1,2}^i = Pr_1^i \cup Pr_2^i$, and sorted as in section III-B, i.e. having the dense part of sensors in the left part of the merged chromosome and the sparse on the right part. Initiate a window, $w^i = min + (max - min) \times (1 - \lambda_i)$, where $min = N$ (the size of the offspring) and $max = 2 \times N$ (the size of the merged chromosome). Note that, w_i varies based on λ_i of each i. Place the window at the left side of $Pr^i_{1,2}$. Randomly, $rand^i \in [1, w^i]$, select $j = 1, ..., N$ genes (=sensors) from $Pr^i_{1,2}$. A gene, e.g. s_{rand^i}, can be selected only once, $Pr^i_{1,2}/\{s_{rand^i}\}$. Add s_{rand^i} in $O_i = O_i \cup \{s_{rand^i}\}$ at position j. Figure 2 exemplifies the ”window”-based crossover operator by giving the wireless sensor network’s interpretation of the extreme subproblem with $\lambda_1 = 1$.

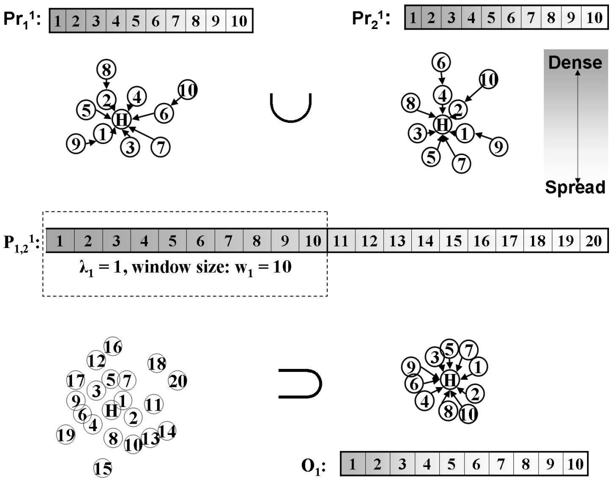


Fig. 2. Example on the problem specific ”*window-based*” crossover operator

3) Mutation: Mutation operator [10] maintains the diversity of the population by randomly modifying the genes of a chromosome based on a mutation rate, m_{rate}. When a gene i is to be modified, a new (x_i, y_i) and R^i_c are randomly chosen based on λ_i. If λ_i is high, then (x_i, y_i) is chosen such that $d_{iH} \leq maxR_c/2]$ and $R^i_c \in [minR_c, maxR_c/2]$ to support dense deployment and f_{lf}, otherwise, (x_i, y_i) is chosen such that $d_{iH} > maxR_c$, $x_i < x_{max}$, $y_i < y_{max}$ and $R^i_c \in [minR_c, maxR_c]$ to support spread deployment and f_{cr}. Parameters $minR_c$ and $maxR_c$ is the minimum and maximum possible R_c assignment.

Note that our algorithm can adopt other kind of genetic operators with minor changes in the algorithm’s design.

E. Local Search: Repair, Improvement and Power Assignment

As mentioned earlier, MOEA/D is suitable on using simple scalar subproblem techniques for locally improving and/or repairing a solution, i.e. Local Search (LS) [6].

1) Repair: Repair heuristic increases the sensors individual utilization in a particular topology. Since, there are no constraints and no infeasible solutions in DPAP, the repair heuristic checks if a sensor i is located: Case #1) close to H, i.e. $d_{i,H} < d_{min}$ and/or Case #2) close to another sensor j, i.e. $d_{i,j} < d_{min}$, where $d_{min} = 1$. In both cases, the solution is considered inefficient and needs repairing (since sensors of Cases #1 or #2 have limited chances to provide extra load balancing and/or cover any uncovered regions in A). The repair heuristic randomly generates a new location $(rand_x, rand_y)$ for sensor i. If λ is high then, i is relocated around H, i.e. $rand_x \in [0, maxR_c]$ and $rand_y \in [0, maxR_c]$ such that $d_{iH} \leq maxR_c$, otherwise, i is relocated sparsely, i.e. $rand_x \in [0, x_{max}]$ and $rand_y \in [0, y_{max}]$ such that $d_{iH} > maxR_c$. The repaired solution X'_R is then forwarded for improvement. The encoding representation of section III-B is proven beneficial for the repair heuristic because of the chromosome sorting (inefficient sensors can be tracked with less computational effort).

2) Improvement: Improvement heuristic is composed by two steps. Each solution should follow one of these two steps according to its λ coefficient and a random variable, $rand$. Each step favors one of the two objectives under consideration. If $rand < \lambda$ then Step 1 is invoked, if $rand \geq \lambda$ then Step 2 is invoked. The two steps work as follows:

Step 1 - Improves lifetime: refers to the subproblems which require high f_{lf}. Hence, the goals of this heuristic are to design dense topologies, to increase load balancing and/or decrease latency. This is attained as follows: Let X^i_R be the repaired solution of subproblem i with N sensors deployed in A. Divide A into four equal spaces. Find the space with the highest density, ($highS$, with high and unbalanced traffic load) and the space with the lowest density, ($lowS$). Select the sensor $j \in highS$ which causes the highest latency. Relocate j in $lowS$ such that it directly communicates with H.

Step 2 - Improves coverage: is mainly invoked on subproblems which prefer high f_{cr}. Therefore, the sensors need to be placed as spread as possible. This is achieved, by withdrawing all sensors as far as $maxR_c/2$ away from their farthest neighbor, starting from the sensors closer to H. Hence, the sensors are spread in A without worrying for disconnections and with increase probability of covering uncovered areas and connecting isolated sensors. Isolated sensors are the sensors placed far away and cannot find a route towards H (mainly because of the genetic operators). Connecting an isolated sensor ensures that a previously uncovered area will be covered, since $R_s \leq maxR_c/2$.

Note that, when $\lambda_i \approx (1 - \lambda_i)$ there is a high probability that both steps are invoked to a subproblem i in different generations giving a hybrid of the two improvements. The improved solution, X^i_I is then forwarded for power assignment.

3) Power Assignment: The main focus of the power assignment heuristic is load balancing. As it is mentioned earlier, the total energy consumption of a sensor i is strictly related to its P_i and the number of k-bit data it transmits (i.e. traffic load). So, each sensor should balance its traffic load through multiple routes and limit its transmission power at the same time.

Figure 3 exemplifies the power assignment heuristic: (Step 1, left) Sensor 4, which is far away from $H (d_{4H} > maxR_c)$, increases its initial transmission range ($R^4_c = avgR_c = maxR_c/2$, indicated by a solid circle) to the maximum possible ($R^4_c = maxR_c$, indicated by a dotted circle). This results to more alternative paths, i.e. $4, 2$ and $4, 3$, and a balance of the traffic load forwarded by 4, alleviating sensor 1 for several cycles. Moreover, it is evident that all four sensors are using more transmission power than the necessary to reach their farthest neighbor. (Step 2, right) Sensors 1,2 and 3 farthest neighbor is H, so all three sensors decrease their R^i_c and

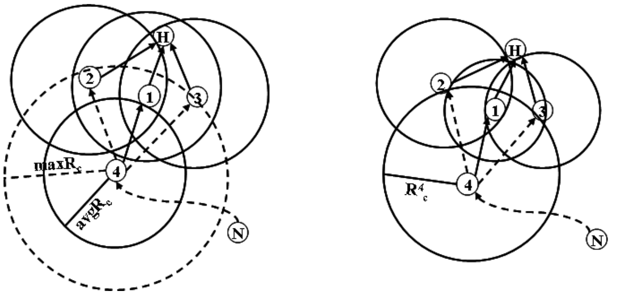


Fig. 3. Power Assignment heuristic

consequently their P_i at the level they can reach H, i.e. $R_c^i = d_{i,H}$ and $P_i = (R_c^i)^\alpha$ where $i = 1, 2, 3$. Meanwhile, sensor 4 is not necessary to transmit with $R_c^4 = maxR_c$. Since, with $P_4 = (R_c^4)^\alpha$, where $R_c^4 = d_{4,3} < maxR_c$, it reaches its farthest neighbor 2, keeps all its alternative paths for load balancing and consumes less transmission energy.

F. Population Update and termination criterion

The population update of MOEA/D is composed by three phases. (I) The Internal Population (IP) update phase. IP keeps all the best solutions found so far for each subproblem, $|IP| = m$. If $g_i(X_i'|\lambda_i) > g_i(X_i|\lambda_i)$ then $IP \cup \{X_i'\}$ and $IP/\{X_i\}$, otherwise X_i remains in IP. (II) The neighboring solutions update phase. The new solution X_i' is compared with its T closest neighbor best solutions j found so far. If $g_j(X_i'|\lambda_j) > g_j(X_j|\lambda_j), \forall j \in 1, ..., T$, then $IP \cup \{X_i'\}$ and $IP/\{X_j\}$, otherwise, X_j remains in IP. (III) The external population (EP) update phase. Population EP stores all the non-dominated solutions found so far during the search, $EP = EP \cup \{X_i'\}$, if X_i' is not dominated by any solution $X_j \in EP$ and $EP = EP/\{X_z\}$, if $X_i' \prec X_z$.

At the end of each generation the termination criterion is checked to decide whether it is time to stop the search. In this paper the maximum number of generations, gen_{max}, is used.

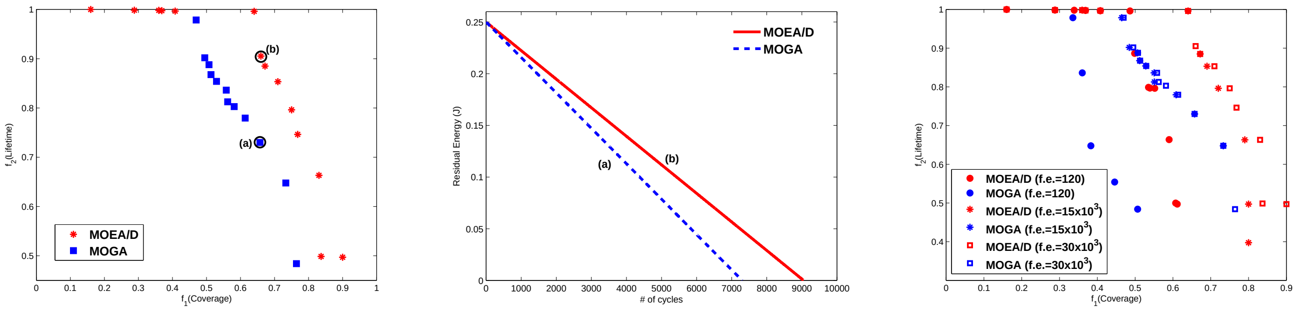
IV. Experimental results and discussion

The performance of MOEA/D is compared with MOGA on several network instances (e.g varying $A = 10000m^2 - 90000m^2$ with fixed density and varying $N = 10 - 500$ with fixed A) in terms of quality of solutions in the PF and convergence speed using the same function evaluations ($f.e. = 120 \times 250 = 30000$). In this paper, due to page limit, we just present the results of the network instance proposed by [9] using the same parameter settings for fairness: $A = 10000m^2$, $N = 10$, $a = 2$, $R_s = 20m$ and $R_c = R_s$. Additionally, for the DPAP we have used a radio dissipation of $E_{elec} = 50nJ/bit$ for the circuitry energy consumption and $\epsilon_{amp} = 100pJ/bit/m^2$ for the transmitter amplifier [13]. In MOEA/D, $R_c \in [1, 40](= [minR_c, maxR_c]$ is calculated using the power assignment heuristic. Moreover, the algorithm's parameter settings are: $m = 120$, crossover rate $c_{rate} = 1$, $m_{rate} = 0.1$ and $gen_{max} = 250$. The two parameters considered only by MOEA/D are the number of subproblems $m = |IP|$, and the neighborhood size $T = 2$, since a large T may affect the diversity of the population.

Figure 4(a) shows: (I) the tradeoffs between the two objectives and how they are competing against each other with a high degree negative correlation of around -0.88. On one hand, the design space of a WSN optimization problem is highly non-linear because of the communication connectivity of the sensors. A small change in the network may cause disconnections or partitions of the network and so big changes in both objective functions. This is the reason why the results, i.e. PF in Figure 4(a), are not continuous. On the other hand, at the top and bottom of MOEA/D 's PF there are similar lifetime solutions. The similarity of these solutions is due to the power assignment heuristic. Specifically, when sensor i is shifted backwards from its farthest neighbor j, the power assignment heuristic increases its P_i to avoid loosing its path towards H. This results to a higher $E_t(k, d_{ij})$ and so a lower f_{lf}, but at the same time it slightly increases f_{cr}. Therefore, two very similar non-dominated solutions are designed.

(II) The superiority of our approach in terms of quality of solutions in the PF. All non-dominated solutions obtained by MOGA are dominated by MOEA/D. (Note that, MOGA's PF is very similar with the one provided by [9] in terms of its trend and spread of solutions. The small difference that appears in the quality is mainly because of the difference in the system model.) MOEA/D has an advantage, of around 12% of coverage over MOGA for the topologies having the same lifetime and a lifetime increase of 18% in average, for the solutions having the same coverage. To be more specific, we decoded and zoom-in two non-dominated solutions, one from each algorithm, giving the same coverage. The solutions (a) on MOGA's PF and (b) on MOEA/D's PF are giving a $f_{cr} = 66\%$ and a lifetime difference around $f_{lf^{(b)}} - f_{lf^{(a)}} = 20\%$, i.e. $f_{lf^{(a)}} = 72\%$ and $f_{lf^{(b)}} = 92\%$. The reason is given by Figure 4(b). During the simulations run we have located the sensor consuming the more energy supply per cycle for each algorithm. This was the same sensor which finally depleted its energy supply first. We are illustrating the per cycle energy consumption of the sensor which is responsible for the lifetime of each (a) and (b) non-dominated topologies.

In both cases, the particular sensors of topologies (a) and (b) start with an initial battery supply of $E = 0.25J$. While the network is "running", it is evident that, sensor of (a) consumes more energy per cycle than sensor of (b). Specifically, sensor of (a) spends its whole energy supply in 7299 cycles, with 72 $e^{-7}J$ more energy per cycle compared to sensor of (b) which spends the same amount of total energy supply in 9051 cycles. This indicates that MOEA/D with a fixed number of sensors in a fixed area can provide the same coverage as MOGA, but, with much higher lifetime. This is mainly due to the following reasons: 1) MOEA/D handles each sub-problem individually, by using problem specific heuristics, and at the same time it absorbs any available beneficial information from the neighborhood. For example, the subproblems dedicated to lifetime (coverage), use the problem specific operators and heuristics to obtain high quality lifetime (coverage) fitness and at the



(a) In terms of quality

(b) Zoom-in (a) and (b) non-dominated solutions with fixed coverage

(c) In terms of Convergence speed

Fig. 4. MOEA/D v.s.MOGA

same time they absorb all the necessary information from the neighborhood for increasing coverage (lifetime). 2) The high load balancing that the problem specific heuristics provide to all solutions in MOEA/D's PF compared to those in MOGA's PF. Similarly, for cases with the same lifetime, MOEA/D provides higher coverage. The reason is that, MOEA/D can spread the sensors without being affected by the increase of latency and consequently the increase of relay data since it provides adequate load balancing to ease the f_{lf}'s decrease.

Figure 4(c) shows three PFs for each algorithm provided after a particular number of function evaluations, $f.e.$, for comparing the methods population initialization and convergence speed. The first PF of each algorithm shows the initial population, i.e. $f.e. = 120$. It is evident, that the initial solutions provided by MOEA/D are of higher quality than those provided by MOGA. This proves the effectiveness of our problem specific population initialization(section III-C) compared to the random approach adopted by MOGA, giving about 15% higher quality of solutions.

The second PF is given for $f.e. = 15 \times 10^3$ (∼125 gens, the middle of evolution) and the last PF for $f.e. = 30 \times 10^3$ (∼250 gens, the end of evolution). In both cases and consequently for the whole evolution, MOEA/D provides high quality PFs faster. MOGA needs 15×10^3 $f.e.$ to provide a similar quality on its top half PF as MOEA/D($f.e = 120$). MOGA($f.e. = 15 \times 10^3$) bottom half PF has higher quality than MOEA/D($f.e = 120$) but much less than MOEA/D($f.e. = 15 \times 10^3$). However, both algorithms indicate a high convergence speed in the initial 125 generations and an increase of around 25% in quality of solutions. The convergence speed of the last 125 generations is not as high as the initials'. The decrease on the convergence speed is expected, since it is much more difficult to increase the quality of the existing non-dominated solutions and/or to obtain new, as the number of generations grows. However, in the last 125 gens MOEA/D gives an average increase on the quality of existing non-dominated solutions of 3%, where MOGA gives 0.05% and MOEA/D provides 3 new non-dominated solutions, where MOGA provides 2. Note that, similar conclusions were drawn for all network instances.

V. Conclusions

In this paper a new Deployment and Power Assignment Problem (DPAP) is formulated as a MOP. Our motivation was to provide a diverse set of high quality solutions for the DPAP in the absence of any prior knowledge on the objectives preference to facilitate a decision maker's choice. An MOEA/D-LS algorithm is designed and showed its superiority against MOGA in terms of quality of solutions and convergence speed. In the future, we intend to add more constraints (e.g. connectivity) on the DPAP to increase its realizability.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Comms Magazine*, pp. 102–114, 2002.

[2] 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.

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

[4] J. Deng, Y. S. Han, P.-N. Chen, and P. K. Varshney, "Optimal transmission range for wireless ad hoc networks based on energy efficiency," *IEEE Trans. on Communications*, vol. 55, no. 9, pp. 1772–1782, 2007.

[5] F. Wang, K. Xu, M. T. Thai, and D. Z. Du, "Fault tolerant topology control for one-to-all communications in symmetric wireless networks," *Int. Journal of Sensor Networks*, vol. 2, no. 3/4, pp. 163–168, 2007.

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

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

[8] 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.

[9] D. B. Jourdan and O. L. de Weck, "Layout optimization for a wireless sensor network using a multi-objective genetic algorithm," in *IEEE Semiannual Vehicular Technology*, vol. 5, 2004, pp. 2466–2470.

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

[11] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, and Y. F. Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards," *Elsevier Computer Communications*, vol. 30, pp. 1655–1695, 2007.

[12] 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.

[13] 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.