Problem-specific Encoding and Genetic Operation for a Multi-Objective Deployment and Power Assignment Problem in Wireless Sensor Networks

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

***Abstract*— Wireless Sensor Networks Deployment and Power Assignment Problems (DPAPs) for maximizing the network coverage and lifetime respectively, have received increasing attention recently. Classical 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 paper, the two problems are formulated as a multi-objective DPAP and tackled simultaneously. Problem-specific encoding representation and genetic operators are designed for the DPAP and a Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) is specialized. The multi-objective DPAP is decomposed into many scalar subproblems which are solved simultaneously by using neighborhood information and network knowledge. Simulation results have shown the effectiveness of the proposed evolutionary components by providing a high quality set of alternative solutions without any prior knowledge on the objectives preference, and the superiority of our problem-specific MOEA/D approach against a state of the art MOEA.**

I. Introduction

Two of the main network configuration requirements in Wireless Sensor Networks (WSNs) [1] topology designs are: (1) high quality location assignment (deployment [2]) and (2) energy efficient power assignment [3]. Several approaches have been proposed for the deployment and power assignment problems for maximizing the coverage [4] and lifetime [5] objectives, respectively. Few approaches, however, have tackled the two problems at the same time, such as [6]. Even though, the latter approaches optimize the objectives individually, or by combining them into a single objective, or by constraining one and optimizing the other, which often results on ignoring and losing "better" solutions.

The conflicting correlation of the WSN's coverage and lifetime objectives directs to a Multiobjective Optimization Problem (MOP) [7] formulation for analysis of the objectives trade-offs [8]. In MOPs, all objectives are considered equal and there is not a single solution to optimize them at the same time. Hence, a set of optimal/nondominated solutions exists, known as the Pareto Front (PF) [9]. Therefore, classical methods are not applicable in such complex, non-linear problems and the adaptation of Multi-Objective Evolutionary Algorithms (MOEAs) [9], which poses all the desirable characteristics for this type of problems, can be proven beneficially.

Conventional MOEAs, however, usually tackle real life problems (e.g. the above WSN problem) as a "black" box (i.e. without any problem knowledge) which may force the evolutionary approaches into unnecessary searches and destructive mating. Therefore, problem specific operators should be designed for MOEAs to direct the search into promising areas in the search space and to provide high quality solutions. Designing a problem specific operator or heuristic, to benefit the MOP as a whole, is difficult. Hence, the decomposition of the MOP into many single objective subproblems [7], which are optimized simultaneously by using neighborhood information and simple scalar optimization strategies, can be a promising technique. The difficulty on designing problem specific genetic operators for a decompositional MOEA is that, each subproblem has a different objective preference and requires a different treatment. Therefore, the genetic operators should adapt to the requirements and the objective preference of each subproblem dynamically during the evolution.

In [8] we have investigated the multiobjective deterministic pre-Deployment and Power Assignment Problem (DPAP). DPAP is typical in applications that invoke a limited number of expensive sensors, where their operation is significantly affected by their position and communication. Besides, we have briefly introduced an approach relying on the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) and we have shown its superiority against a widely used MOEA. In this work, we describe and illustrate the new encoding representation and the problem specific genetic operation designed specifically for the DPAP and adapted to the MOEA/D framework. We show the necessity of our encoding and genetic operators by comparing them with conventional cases. Finally, we test the strength of our problem-specific MOEA/D approach against the state of the art in MOEAs, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [9].

II. Problem Definition

A. System Model

Consider a 2-D static WSN 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 to monitor and periodically report an event of interest to H. Hence, each sensor i, 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 [6]. 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 and j and $R_c^i = d_{ij}$ is i's transmission 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 $r_i(t)$ is the total traffic load that sensor i receives and relays towards H at t (the "+1" is the data packet generated by sensor i to forward its own data information) and amp is the power amplifier's energy consumption. Note that, the energy consumed by the transceiver electronics, as well as, for reception and generation of data are considered negligible and ignored, since we assume that the sensor nodes communicate through long transmission distances. Thus, the transmit power consumption is the dominant parameter on the total energy consumption [6].

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 [4]. 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 [4].

B. Problem formulation

The DPAP can be formulated as a MOP,

Given:

- A: 2-D plane of 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.

Design variables:

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

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

The network coverage $Cv(X)$, defined as the percentage of the covered grids over the total grids of A, 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

$$g(x', y') = \begin{cases} 1 & \text{if } \exists j \in \{1, ..., N\},\ d_{(x_j, y_j),(x', y')} \leq R_s \\ 0 & \text{otherwise} \end{cases}$$

is the monitoring status of the grid centered at (x', y').

The network lifetime is defined as the percentage of 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 [10] routing algorithms).

III. The proposed problem-specific evolutionary components in MOEA/D

A. Briefing on MOEA/D

The MOP can be decomposed into m subproblems using any technique that constructs aggregation functions, e.g. the Weighted Sum Approach [7]. Then, a subproblem i with a weight coefficient λ^i can be defined as:

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

Thereinafter, the Internal Population, IP (stores the best solutions found for each subproblem during the search) is randomly initialized. Then O^i solutions are generated using the genetic operators, where $i = 1$ to m, and an improvement heuristic is applied on each O^i to produce X^i. In the update phase, $IP/\{O^i\}$ and $IP \cup \{X^i\}$, if $g_i(X^i|\lambda^i) > g^i(O^i|\lambda^i)$, otherwise O^i remains in IP. The neighborhood of X^i (i.e. the solutions of the T closest subproblems of i in terms of their weights $\{\lambda^1, \cdots, \lambda^m\}$) is then updated. If $g^j(X^i|\lambda^j) > g^j(O^j|\lambda^j)$, then $IP/\{O^j\}$ and $IP \cup \{X^i\}$, otherwise, O^j remains in IP, where $j \in \{1, ..., T\}$. The external population (EP) (stores all the non-dominated solutions found so far during the search) is finally updated, $EP = EP \cup \{X^i\}$, if X^i is not dominated by any solution $O^j \in EP$, and $EP = EP/\{O^j\}$, for all O^j dominated by X^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 preference and they have to be solved simultaneously, in a single run. In order to overcome this difficulty, we have developed problem specific operators 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 [7] 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, $EP = \{X^*\}$.
Step 0-Setup: Set $EP := \emptyset$; $gen := 0$; $IP := \emptyset$;
Step 1-Decomposition: Initialize m subproblems, i.e. max$g^i(Y^i|\lambda^i)$, for $i = 1, ..., m$.
Step 2-Initialization: Randomly generate an initial internal population $IP = \{Y^1, \cdots, Y^m\}$;
Step 3: **For** each subproblem $i = 1$ to m **do**
Step 3.1-Genetic Operators: Generate a new solution O^i by using selection, crossover and mutation operators.
Step 3.2-Improvement: Apply a problem specific repair/improvement heuristic on O^i to produce X^i.
Step 3.3-Update Populations: Update IP, EP and the T closest neighbors of subproblem i with 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**.

In this paper, the focus is on the encoding representation and the problem-specific genetic operators, which are described and illustrated below. More details about our DPAP-specific local heuristic of MOEA/D can be found in [8].

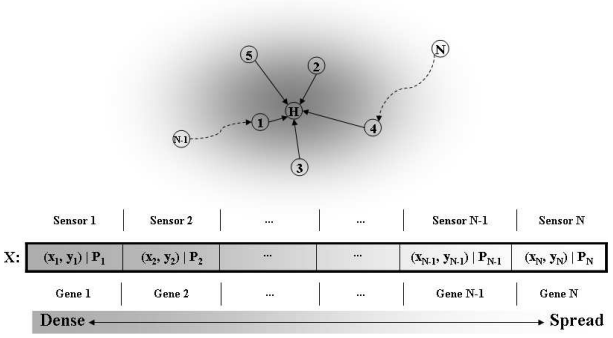


Fig. 1. Encoding Representation

B. Encoding Representation

The encoding represenation is a set Y^i of size N, composed by the DPAP's design variable vectors:

$$Y^i = \{(L_1^i, P_1^i), \cdots, (L_N^i, P_N^i)\}$$

where,

$$L_j^i = (x_j^i, y_j^i), \text{location of sensor j}$$

$$P_j^i, \text{transmit power level of sensor j}$$

For the benefit of our approach the set is ordered and named *"dense-to-spread"* representation as follows: the sensor nodes j in Y^i are sorted based on their distance to H, i.e. d_{jH}, where sensor 1 is the closest and sensor N is the farthest sensor towards H, respectively. This results on having all the sensors that are densely deployed around H at the beginning of each chromosome and the sensors that are spread away at the end. Thereinafter, each sensor j is assigned a power level P_j^i proportional to $R_c^{i,j} \in [1, maxR_c]$ (where 1 and $maxR_c$ are the minimum and maximum possible transmission ranges, respectively) such that it reaches its closest neighbor sensor, e.g. k, where $k < j-1$. A "dense-to-spread" encoding representation is designed as follows:

Input: A set Y of size N;
Output: An ordered set X of size N;
Step 1: for each subproblem i **do**
Calculate the order in Y^i to get X^i;
Step 1.1:for each sensor j in X^i **do**

$$P_j^i = \begin{cases} (d_{j,H})^\alpha & \text{if } j = 1, d_{j,h} \leq maxR_c \\ & \text{if } k \text{ is } j\text{'s closest sensor,} \\ (d_{j,k})^\alpha & k < j, P_k^i \neq 0 \\ & d_{jk} \leq maxR_c \end{cases}$$

- In Step 1, the ordering facilitates our problem-specific genetic operators (will be introduced shortly) and improvement heuristics [8].
- In Step 1.1, each sensor j is assigned the smallest possible P_j^i following the concept that multiple short hops are more beneficial than a long hop [6] (e.g. in applications where N is small and the sensor nodes communicate through long transmissions). The reason is that

$$r_j(t) \times d_{jk}^\alpha > r_j(t) \times d_{jl}^\alpha + r_l(t) \times d_{lk}^\alpha$$

since $d_{jk} > d_{jl} + d_{lk}$.
- Initially $P_j^i = 0$ for all sensors j. So, after step 1.1 if a sensor j has $P_j^i = 0$ then is considered disconnected.

C. Genetic Operators

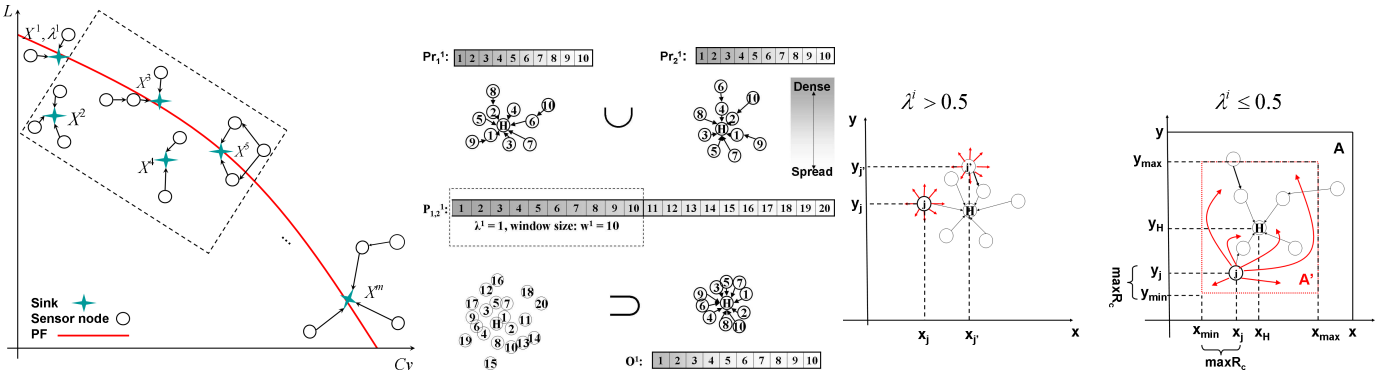
This subsection introduces the proposed genetic operators.

1) Weight-tournament selection operator: is responsible to emulate the survival of the fittest concept and choose high quality solutions from the current population to be included for reproduction. In this paper, we have designed a tournament selection operator which is simple and fast. Our weight-tournament selection has two major differences compared to the conventional:

- the solutions selected to compete in a subproblem i's tournament are the solutions of the s_t closest subproblems of i in IP, which can also be called X^i's neighbors.
- X^i's neighbors, e.g. X^j and X^k, are competing in i's tournament, in terms of λ^i, ignoring their own λ^j and λ^k, their Pareto domination and/or ranking.

Figure 2(a) exemplifies a tournament for subproblem 1 of size $s_t = 4$. Solutions, e.g. $\{X^2, X^3, X^4, X^5\}$, are competing in terms of λ^1. In this case, a neighbor solution X^2 is better than a neighbor solution X^3, if

$$\lambda^1 L(X^2) + (1 - \lambda^1) Cv(X^2) > \lambda^1 L(X^3) + (1 - \lambda^1) Cv(X^3)$$



(a) The weight-tournament selection operator (b) The "window"-based crossover operator (c) The adaptive mutation operator

Fig. 2. The problem specific genetic operators

Our weight-tournament selection operator relies on one of the core ideas of MOEA/D, i.e. two neighbor solutions in the weight space should be similar to each other in the decision space. Hence, a solution X^i is more likely to absorb good topological information (i.e. efficient sensor locations and transmission power levels) from a neighbor solution X^j than a solution, e.g. X^m, which is far away in the weight space, even if X^m is a non-dominated solution. The selected solutions, denoted as Pr_1^i and Pr_2^i parent solutions, are then forwarded for recombination to the crossover operator.

2) Crossover: is the operator that recombines Pr_1^i and Pr_2^i to produce one, or more offspring, e.g. O^i, with probability c_{rate}. The "window"-based crossover determines a "window" of size

$$w^i := min + (max - min) \times (1 - \lambda_i)$$

where $min := N$ and $max := 2 \times N$, to direct the search into promising areas in the search space for each particular i. Our *"window"*-based crossover works as follows:

Input: Two sets Pr_1^i and Pr_2^i of size N;
Output:A set O^i of size N;
Step 0: Set $O^i = \emptyset$;$Pr_{1,2}^i = \emptyset$
For $i = 1$ to m **do**
Step 1: Merge Pr_1^i and Pr_2^i and create a "dense-to-spread" set of $Pr_{1,2}^i$ of length $2 \times N$;
Step 2: Calculate w^i;
Step 3: Randomly generate $rand \in [1, w^i]$;
If there exists a sensor $rand$ in $Pr_{1,2}^i$ **then**
Step 3.1:$Pr_{1,2}^i \setminus \{rand\}$; $O^i \cup \{rand\}$;
Step 3.2: If O^i's length is not equal to N goto Step 3 o.w. stop;

- When λ parameter is large and $L(X^i)$ favors $Cv(X^i)$, the "window" is small such that the sensor nodes that are going to be added in O^i are as close to H as possible with small transmit powers to provide higher lifetime.
- When λ decreases, and $Cv(X^i)$ starts favoring $L(X^i)$, w^i gradually increases to give the chance to the sensors that are spread in A to be added in O^i, and therefore to provide better coverage. In this case, the N sensors are uniformly selected from w^i to prevent disconnections of the sensors which are far away from H.

Figure 2(b) exemplifies the WSN interpretation of our crossover operator on the extreme subproblem 1.

3) Adaptive mutation operator: The mutation operator [5] maintains the diversity of the population by randomly modifying the genes of a chromosome based on a mutation rate, m_{rate}. Our mutation operator treats each subproblem i based on its λ^i. If λ^i sustains $L(X^i)$ then a sensor j is modified with "local" mutation. That is, provide a minimum shift from its current position $(x_j \pm 1, y_j \pm 1)$, trying to either slightly increase $Cv(X^i)$ with minimum increase of P_j^i, or increase $L(X^i)$ by decreasing d_{jk}, assuming that k is j's closest neighbor. In any case, j's shift should benefit X^i.

If λ^i sustains $Cv(X^i)$, then j is modified with "global" mutation, i.e. j is re-deployed anywhere in a sub-area $A' \in A$, which is defined as follows:

$$\begin{aligned} x_{min} &= (x_H - |x_H - x_j|) - maxR_c; \\ x_{max} &= (x_H + |x_H - x_j|) + maxR_c; \\ y_{min} &= (y_H - |y_H - y_j|) - maxR_c; \\ y_{max} &= (y_H + |y_H - y_j|) + maxR_c; \\ x' &= x_{max} - x_{min}; \\ y' &= x_{max} - x_{min}; \\ A' &= x' \times y' \end{aligned} \tag{4}$$

where (x_{min}, y_{min}) are the coordinates of the left bottom corner of A' and x', y' are the width, height of A' respectively.

Sensor j is then assigned a $P_j^i = (d_{jk})^\alpha$, assuming that k is j's closest neighbor. Usually, when $\lambda \approx 0$ then $A' \approx A$. Figure 2(c) illustrates our adaptive mutation operator which proceeds as follows:

Input: A set O^i of size N.
Output: A modified set O^i of size N.
Step 0:If $\lambda^i > 0.5$ **then** goto Step 1, o.w. goto Step 2
Step 1: Uniformly randomly set $x'_j \in [x_j - 1, x_j + 1]$; $y'_j \in [y_j - 1, y_j + 1]$; $P'^i_j = P^i_j \pm 1$;
Step 2: Calculate A' (Eq. 4); Uniformly randomly set $(x'_j, y'_j) \in A'$;$R_c^{ij} \in [0, maxR_c]$; $P'^i_j = (R_c^{ij})^\alpha$;

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

IV. Experimental results and discussion

The goal of our experimental studies is: 1) to test the effectiveness of our problem specific encoding and genetic operators, and 2) to test the strength of our problem-specific MOEA/D based approach against NSGA-II in various network instances, giving the trade offs of our objectives and a variety of network design choices.

TABLE I

The network instances

Instances	A (m^2)	Density (sensor per m^2)
Inst.0	10000	0.003
Inst.1	10000	0.0032
Inst.2	40000	0.0032
Inst.3	90000	0.0032
Inst.4	10000	0.0015
Inst.5	10000	0.002
Inst.6	10000	0.006

Table I shows various network instances. Instances 1-3 represent networks of different As and same density (i.e. N/A). Instances 4-6 represent networks with different densities in the same A. In all experiments we have used the following network parameter settings: $a = 2$, $R_s = 10m$, $amp = 100pJ/bit/m^2$ [10], $maxR_c = 20m$. Moreover, we have used the following algorithm's parameter settings: $m = 120$, $c_{rate} = 1$, $m_{rate} = 0.1$, $s_t = 10$ and $gen_{max} = 250$. The parameter considered only by MOEA/D is $T = 2$, since a large T may affect the PF's diversity.

TABLE II

MOEA/D based algorithms with different genetic components

Algorithm	Encoding	Selection	Crossover	Mutation
Alg.1	Random	Tournament	One-point	Random
Alg.2	Dense-spread	Tournament	One-point	Random
Alg.3	Dense-spread	λ-based Tour.	One-point	Random
Alg.4	Dense-spread	λ-based Tour.	"Window"	Random
Alg.5	Dense-spread	λ-based Tour.	"Window"	Adaptive

Table II shows five algorithms with different genetic components combinations. The first algorithm (i.e. Alg. 1) is composed by the "default" operators which are simple and popular: random encoding representation (sensors are added randomly in X), tournament selection operator [5] (initialize a tournament of random solutions from IP), one-point crossover operator [5] and random mutation operator (randomly generate (x'_i, y'_i) and P'_i for sensor i, s.t. $x'_i \in [0, x]$, $y'_i \in [0, y]$ and $R^i_c \in [1, maxR_C]$. The rest of the algorithms replace some of the default operators with our problem specifics' to investigate their necessity. The last algorithm (i.e. Alg. 5) includes all the problem specific genetic operators proposed in this work, as well as, our encoding representation. Note that, we have investigated more combinations of the genetic components that are not presented in this paper due to page limit.

Initially, we have compared the five algorithms, on network instance 0, in terms of quality of solutions in the PF (C metric) and diversity (Δ metric [9]). C(A,B) measures the solutions in an algorithm A's PF dominated by the solutions in an algorithm B's PF (i.e. the smaller the better). $\Delta(A)$ shows the diversity of the PF obtained by an algorithm A, i.e. the spread/variety of the network design choices. $\Delta = 0$ is the maximum, which means that the solutions are evenly spread along the PF.

TABLE III

Comparison in terms of C_{metric} and D_{metric}

Δ :	$\Delta(1)$	$\Delta(2)$	$\Delta(3)$	$\Delta(4)$	$\Delta(5)$
Value:	0.89	0.76	0.53	0.76	0.501
C:	C(1,2)	C(2,1)	C(2,3)	C(3,2)	
Value:	36.3%	23.0%	69.2%	4.7%	
C:	C(3,4)	C(4,3)	C(4,5)	C(5,4)	
Value:	36.3%	23.0	46.8%	41.5%	

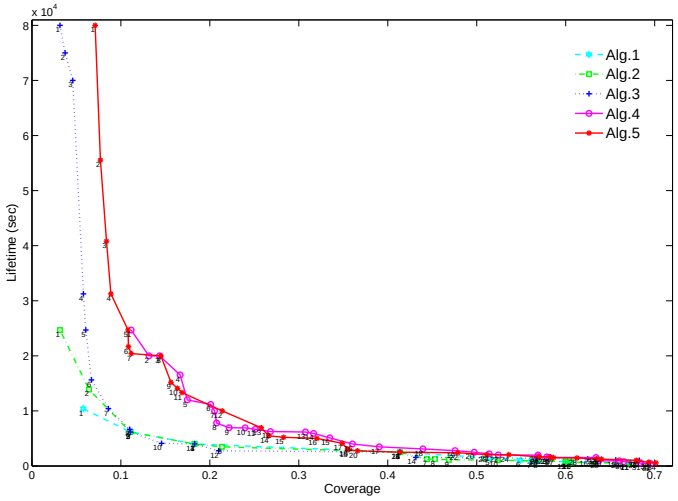
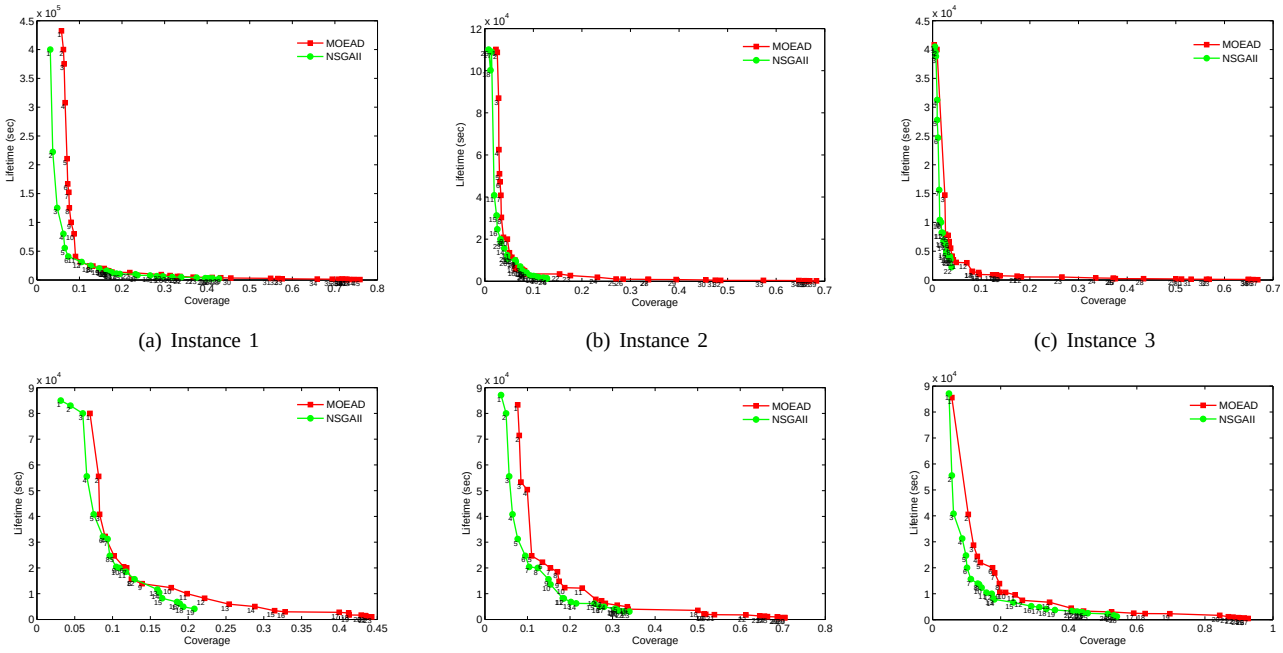


Fig. 3. Comparison of various genetic components on Inst.0

It is evident, from both Figure 3 and the C metric in Table III that Alg. 5 obtained the highest quality on the network designs. Alg. 2 provides about 500 sec. lifetime and 1% coverage increase compared to Alg.1. Alg.3 in turn provides an increase of 3500 sec. of lifetime and 2% coverage, in average. Alg.4 further increases the average solutions lifetime and coverage of around 2000 sec. and 10% respectively. Finally, Alg.5 provides an increase of 250 sec. lifetime and 0.1% coverage, in average, compared to Alg.4.

The Δ metric in Table III, however, shows an increase on the diversity of the solutions obtained by Alg.1 to 3. The diversity is then suddenly decreased from Alg.3 to Alg.4 because of the problem specific crossover operator. The reason is that, our "window"-based crossover increases the solutions quality, but it converges into local optima solutions easily, which results on a decrease of the PF's diversity. This drawback, however, is eliminated by our adaptive mutation operator included in Alg.5. Alg.5 preserves the increase on the PFs quality achieved by Alg.4 and, at the same time, maintains the diversity. Note that, similar conclusions are drawn for all network instances.

Thereinafter, we have tested the strength of our MOEA/D-based approach (Alg.5) against NSGA-II on network instances 1-6. In all experiments, we have used the same number of function evaluations (f.e) [8] for fairness. Our MOEA/D-based



(a) Instance 1 (b) Instance 2 (c) Instance 3

(d) Instance 4 (e) Instance 5 (f) Instance 6

Fig. 4. Comparison of MOEAD (Alg.5) against NSGAII in various network instances

approach adopted the proposed problem-specific operators and NSGA-II the default operators. Figure 4 clearly shows the superiority of the proposed approach against NSGA-II in quality of solutions in the PFs. In instances 1-3, MOEA/D provides 20% more lifetime in network topologies with the same coverage and 26% more coverage for network topologies with the same lifetime, in average.

In instances 4-6, MOEA/D provides network topologies of 25% higher lifetime (for the same coverage) and 14% higher coverage (for the same lifetime), in average, than those obtained by NSGA-II. Note that, NSGA-II with the default encoding and genetic operators does not provide any topologies with more than 50% coverage in any network instance. The reason is that, under the parameter settings and requirements of the above DPAP, it is difficult to obtain high quality spread-like topologies without network knowledge, because the 'default' genetic operators fall into destructive mating and unnecessary search. Thereupon, NSGA-II directs the search into areas in the search space with high quality dense-like topologies which can be obtained more easily. Therefore, it is clear from Figure 4 that MOEA/D provides a more diverse set of solutions in all network instances giving more network topologies than NSGA-II, without any prior knowledge on the objectives preference, to facilitate the decision maker's choice.

V. Conclusions

In this paper, an encoding representation and various genetic operators were designed specifically for the multi-objective DPAP in WSNs and a MOEA/D based approach is proposed. Initially, we have shown the necessity of incorporating network knowledge in the genetic components in terms of quality and diversity of solutions in the PF. Thereinafter, simulation evaluations have shown the superiority of the proposed approach against NSGA-II. Our motivation was to design a DPAP-specific MOEA/D and to provide a diverse set of high quality WSN designs in the absence of any prior knowledge on the objectives preference. In the future, we intend to add constraints in DPAP to further 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] X. Xu and S. Sahni, "Approximation algorithms for sensor deployment," *IEEE Transactions on Computers*, vol. 56, no. 12, 2007.

[5] 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.

[6] 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.

[7] 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.

[8] 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 GlobeCom08*, 2008, accepted.

[9] 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.

[10] 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.