Computer Communications xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition

Andreas Konstantinidis [a,*], Kun Yang [b]

[a] Department of Computer Science, University of Cyprus, CY-1678 Nicosia, Cyprus
[b] School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, UK

ARTICLE INFO

Article history:
Received 3 June 2010
Accepted 25 August 2010
Available online xxxx

Keywords:
Wireless sensor networks
Connectivity
Fault tolerance
Multi-objective optimization
Evolutionary algorithms

ABSTRACT

The K-connected Deployment and Power Assignment Problem (DPAP) in WSNs aims at deciding both the sensor locations and transmit power levels, for maximizing the network coverage and lifetime objectives under K-connectivity constraints, in a single run. Recently, it is shown that the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) is a strong enough tool for dealing with unconstraint real life problems (such as DPAP), emphasizing the importance of incorporating problem-specific knowledge for increasing its efficiency. In a constrained Multi-objective Optimization Problem (such as K-connected DPAP), the search space is divided into feasible and infeasible regions. Therefore, problem-specific operators are designed for MOEA/D to direct the search into optimal, feasible regions of the space. Namely, a DPAP-specific population initialization that seeds the initial solutions into promising regions, problem-specific genetic operators (i.e. *M*-tournament selection, adaptive crossover and mutation) for generating good, feasible solutions and a DPAP-specific Repair Heuristic (RH) that transforms an infeasible solution into a feasible one and maintains the MOEA/D's efficiency simultaneously. Simulation results have shown the importance of each proposed operator and their interrelation, as well as the superiority of the DPAP-specific MOEA/D against the popular constrained NSGA-II in several WSN instances.

1. Introduction

The design of Wireless sensor networks (WSNs) [1] is a highly complicated task with substantial impact on the quality, cost and efficiency of real life sensor applications. Sensors are small electronic devices with limited energy, memory and transmit power capabilities, which in some sensor network applications are also limited in number because of their high cost [2]. A typical goal of these network designs is to form a long lived WSN, such that the sensors, using their sensing capabilities and wireless transceivers, effectively cover a region of interest and forward important information to a common collection point, usually referred to as a data sink. The Deployment and Power Assignment Problem (DPAP) [3] in WSNs aims at deciding both optimal sensor locations and transmit power levels for maximizing the network coverage and lifetime in a single run. The multi-objective DPAP is typical for applications that invoke a limited number of expensive sensors, where their operation is significantly affected by their position and communication [4]. In these cases, the random massive deployment [5] and dynamic power assignment [6] is not the only choice; and the application affords the "luxury" of using a centralized or even an off-line algorithm to compute the locations and transmit power levels of the sensors, prior deployment.

In WSNs, connectivity is also crucial for most applications [7,8], since a possible partition of the network into disjoint parts may cause undesirable effects, such as decreasing the coverage and consequently the amount of information forwarded to the interested users. A natural generalization of connectivity is K-connectivity or K-fault tolerance [9,10]. Fault tolerance is a central challenge in WSN design, since the failure of the battery constrained sensors is very common in most applications. A WSN design is usually self-healing when each sensor sustains $K-1$ faulty neighbors (i.e. *K*-fault tolerant WSN design). However, most studies [7–10] focus at deciding either the locations or transmit power levels, for maximizing the network coverage or lifetime individually, or by constraining one and optimizing the other, while maintaining connectivity (i.e. $K=1$) and/or designing *K*-fault tolerant WSNs. This often results in ignoring and losing "better" solutions, since the WSN coverage and lifetime are conflicting objectives and a set of trade-off candidates is required. Moreover, the two decision variables highly influence both objectives and constraints, and should be optimized simultaneously [11,12]. Thus, we have considered it important and challenging to investigate the multi-objective K-connected DPAP in WSNs.

* Corresponding author. Tel.: +357 22 892682; fax: +357 22 892701.
E-mail addresses: akonstan@cs.ucy.ac.cy (A. Konstantinidis), kunyang@essex.ac.uk (K. Yang).

0140-3664/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2010.08.011

Multi-Objective Optimization (MOO) is a relatively new area in WSNs and it is difficult to apply an existing linear/single objective method to effectively tackle a Multi-objective Optimization Problem (MOP), giving a set of non-dominated solutions. Thus, the DPAP is used as a real world benchmark for multi-objective methods such as the Multi-Objective Evolutionary Algorithms (MOEAs), which are good in obtaining a set of non-dominated solutions in a single run. However, in most applications of MOEAs to WSNs the MOP is treated as a "black box" [13], i.e. without using problem-specific knowledge, which may have undesirable effects such as forcing the evolutionary process into unnecessary searches and destructive mating negatively affecting their overall performance [3,12]. This can be considered as a main drawback of the generic MOEAs when dealing with real life problems (such as DPAP). For example, the studies in [14–16] tackled different deployment problems in WSNs with general purpose MOEAs, such as the Non-Dominated Sorting Genetic Algorithm-II [17] (NSGA-II), utilizing mainly a random population initialization, the general purpose tournament selection, the single point crossover and a random mutation for offspring reproduction. NSGA-II is also used by Jia et al. for tackling two multi-objective optimization scheduling problems [18,19]. The authors have utilized a random population initialization, the basic tournament selection, two-point crossover and random/swap mutation operators, respectively.

In [3], we have shown that the Multi-Objective Evolutionary Algorithm based on Decomposition [20] (MOEA/D) is a strong tool to tackle unconstrained real world problems (e.g. DPAP) emphasizing the importance of incorporating WSN knowledge for increasing its efficiency. However, the addition of constraints in DPAP necessitates constraint handling and render the tailoring of the existing DPAP-specific MOEA/D, to match the abundance of the constraints and objectives of the K-connected DPAP [21] for WSN design in its full practical complexity, as a major challenge. MOEAs with constraint handling focus at obtaining a set of feasible Pareto optimal solutions, i.e. Pareto Front (PF) [17], providing the trade-off between two or more conflicting objectives. Feasible are the solutions that satisfy all constraints, and infeasible are those that do not. In the literature, there are several constraint handling techniques [22], including the use of a penalty function, adopting the rules of the superiority of feasible solutions, using specialized operators for generating only feasible solutions and using a repair heuristic [23]. For the latter, Coello [22] has effectually declared that "*any heuristic which would guide the repair process, and the success of this approach relies mainly on the ability of the user to come up with such a heuristic*." Hence, repair heuristic is a good choice when an infeasible solution can be easily transformed to a feasible one without harming the optimization process and is carefully designed with problem domain knowledge. In WSNs, the concept of constrained MOO has surprisingly received limited attention, and although there are a considerable number of proposed hybridizations of MOEAs with constrained handling methods, the number of reported applications of those techniques is still relatively scarce.

Raich and Lizkai [24] investigated the optimization of sensor layouts for detecting damages on structural systems. The goal of their study was to minimize the number of sensors while maximizing the amount of information collected under a maximum number of available sensors. The authors adopted the generic NSGA [25] and a simple repair method for handling the infeasible solutions. Dunn and Olague [26] investigated a sensor planning problem, in which the goal was to optimize a group of sensing actions embedded on a robot so that an object is highly and accurately reconstructed in 3D, by minimizing the 3D reconstruction uncertainty and the process cost (i.e. the motion of the robot) under some local and global restrictions. The authors adopted the NSGA-II [17] utilizing a binary tournament selection, a simulated binary crossover and a deterministic mutation operator as well as different deterministic repair operators for constraint handling. More recently (in 2008), Kim et al. [27] have also used NSGA-II to tackle a surveillance sensor placement problem, using a random population initialization, the single point crossover for offspring reproduction with a restriction, i.e. the common elements in the two parents are not exchanged and a specialized node-exchange mutation for handling the infeasible solutions. Finally, Molina et al. [28] investigated a WSN layout optimization problem with the goal of maximizing the lifetime while minimizing the number of sensors, under a fully covered sensing area. The authors compared different MOEAs (e.g. NSGA-II, MOGA), utilizing a specialized variable length representation, a deterministic population initialization, the shift and add-remove mutation operators and the traditional two-point crossover operator in all cases. A penalty function is used for dealing with infeasible solutions.

In this paper, the K-connected DPAP in WSNs is defined and formulated as a constrained MOP. The proposed problem is tackled by a problem-specific MOEA/D approach composed of: (a) a DPAP-specific population initialization that seeds the initial solutions into the feasible regions of the search space, (b) specialized genetic operators (i.e. *M*-tournament selection, adaptive crossover and mutation) for generating new, high quality, feasible solutions at each iteration and (c) a DPAP-specific Repair Heuristic for transforming the infeasible solutions to feasible without deteriorating the objective functions. At this point, it is important to note that the genetic operators of (b) are, initially, successfully utilized in [3] for dealing with the unconstrained DPAP in WSNs. However, they are also adopted here because it is our belief that the same operators can direct the search into high quality, feasible regions of the objective space and therefore improve the performance of MOEA/D and reduce the number of repair function evaluations in the proposed constrained DPAP. Simulations studies demonstrate the importance of each operator, individually, at increasing the overall performance of MOEA/D and decreasing the overall function evaluations under various parameter settings. Finally, it is shown that the proposed MOEA/D approach performs better than the popular constrained NSGA-II in several network test instances.

2. The K-connected Deployment and Power Assignment (K-connected DPAP)

2.1. System model and assumptions

Consider a 2D 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. We assume a perfect medium access control, such as SMAC [29], which ensures that there are no collisions at any sensor during data communication and we adopt the simple but relevant path loss communication model as in [2]. 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}, where d_{ij} is the Euclidean distance between sensors i and j. The communication range of each sensor i is $R_c^i = d_{ij}$, s.t. $R_c^i \leqslant R_{max}$, where R_{max} is the maximum communication range that is determined by the maximum transmit power level that a sensor can be assigned, denoted as P_{max}. The transmit power level P_i and the coordinates (x_i, y_i) are the DPAP's decision variables that are considered fixed for the whole network lifetime, for sensor $i = 1, \ldots, N$. The residual energy of sensor i, at time t, is calculated as follows:

$$E_i(t) = E_i(t-1) - \left[E_{tx}^i(t) + E_{rx}^i(t) + E_s\right], \quad (1)$$

where $E_{tx}^i(t) = k \times (r_i(t)+1) \times (P_i \times amp + E_{ct})$, $E_{rx}^i(t) = k \times r_i(t) \times E_{ct}$ is the amount of energy consumed by sensor i for transmission

Please cite this article in press as: A. Konstantinidis, K. Yang, Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.08.011

and reception, respectively, E_s is the amount of energy consumed for sensing and processing k, which is the amount of data sensed and collected by a sensor with a fixed sensing range R_s, $(r_i(t)+1)$ is the total traffic load that sensor i forwards towards H at t, $r_i(t)$ is the traffic load that i receives and relays and "+1" is the data packet generated by i to forward its own data information, amp is the power amplifier's energy consumption and E_{ct} is the energy consumption due to the transmitter and receiver electronics.

Furthermore, it is assumed that A is divided into G uniform consecutive grids to make the coverage problem more computationally manageable. The size of the grids is several times smaller than $\pi \times R_s$ for a more accurate approximation within the sensing disk. A sensing model based on the definite range law approximation is considered [7]

$$g(x',y') = \begin{cases} 1, & \text{if } \exists j \in \{1,\ldots,N\},\ d_{(x_j,y_j),(x',y')} \leqslant R_s, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

is the monitoring status of a grid centered at (x',y') with *1* indicating that the grid is covered and 0 otherwise.

Finally, the connectivity status of a sensor j is denoted as

$$c_j = \begin{cases} 1, & \text{if } j \text{ is K-connected}, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where sensor j is usually considered K-connected [9] if it directly communicates with H, or if it sustains K neighbors with positive-advance towards H (i.e. neighbors are closer to H than j [30]), considering the many-to-one communication nature of WSNs.

2.2. Problem formulation

The K-connected DPAP in WSNs can be formulated as a constrained MOP,

Given:

- A: 2D 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.
- P_{max}: maximum transmission power level, the same for all sensors.

Decision variables of solution X:

- (x_j,y_j): the location of sensor j.
- P_j: the transmission power level of sensor j.

Objectives: Maximize coverage $Cv(X)$ and lifetime $L(X)$, **subject to** K-connectivity $Cn(X) = 1$.

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:

$$Cv(X) = \left[\sum_{\text{all}(x',y')} g(x',y') \right] / G, \tag{4}$$

where, G is the total grids of A and $g(x',y')$ is calculated using Eq. (2).

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

Algorithm 1 (*Lifetime Evaluation*).

Step 0: Set $t := 1$; $E_j(0) := E$, $\forall j \in \{1,\ldots,N\}$;
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** $\exists j \in \{1,\ldots,N\}$ such that $E_j(t) = 0$ **then** stop and set:

$$L(X) := t; \tag{5}$$

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

The same algorithm can be easily modified to consider different energy models in Step 1.1 (e.g. [12]) and routing algorithms in Step 1.3 (e.g. geographical-based [30,31] routing algorithms).

The percentage of K-connected sensors in X can be measured as follows:

$$Cn(X) = |CS|/N, \tag{6}$$

where $CS = \{j | c_j = 1\}$, $Cn(X) = 1$ when all sensors are K-connected and c_j is calculated using Eq. (3).

In Multi-Objective Optimization (MOO) [32] we need the following definitions. We assume that we have n objectives $f_1,\ldots,f_n$ to maximize.

Definition 1 (*Pareto dominance*). Suppose x and y are two decision variables, x is said to *dominate* y, denoted by $x \succ y$, if and only if $f_i(x) \geqslant f_i(y)$ for every $i \in \{1,2,\ldots,n\}$ and $f_j(x) > f_j(y)$ for at least one index $j \in \{1,2,\ldots,n\}$.

Definition 2 (*Pareto optimality*). $x^* \in \Omega$ is said to be *Pareto optimal (or non-dominated)* if there is no another $x \in \Omega$ so that x dominates

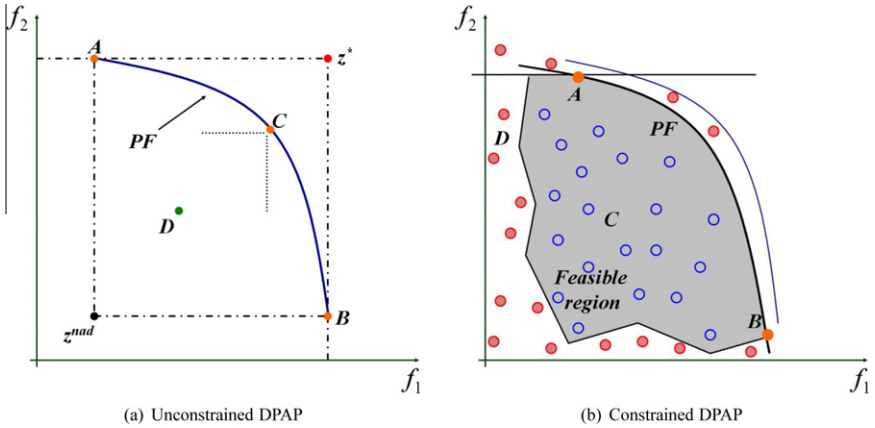


Fig. 1. The PF of unconstraint and constrained DPAPs.

Please cite this article in press as: A. Konstantinidis, K. Yang, Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.08.011

x^*. The set of all Pareto optimal solutions in the decision space is called the *Pareto optimal Set (PS)*. The image of the PS in the objective space is called the *Pareto optimal Front (PF)*.

Fig. 1(a) illustrates a PF with $n = 2$. The Pareto optimal solutions in the PF (marked as solid circles) provide better lifetime and/or coverage than any other solution in the objective space. The remaining solutions are all dominated by at least one solution of the PF. Solutions X^A and X^B are often called the extreme points of the PF [32], since they provide the highest f_1 and f_2, respectively, than any other solution in the objective space. The ideal solution z^* and the nadir solution z^{nad} provide the maximum and minimum objective values, respectively, and they are often considered unreachable.

In addition to multiple objectives, most real world optimization problems involve constraints as well. In that case, the search space Ω can be defined as follows:

$$\Omega = \{x | x \in X,\ x \text{ meets constraints } c_1, \ldots, c_q\}, \quad (7)$$

where, X is the base search space and x is a solution of the *feasible region* Ω that satisfies all q constraints, i.e. $c_1, \ldots, c_q$. The solutions that satisfy all constraints are called *feasible*, and those that do not, *infeasible*. In Fig. 1(b), the bold curve indicates the PF of a constrained MOP, including the extreme solutions A and B. The shaded area shows the feasible region composed of feasible solutions, such as C. All other solutions, e.g. D, are considered infeasible.

2.3. DPAP solution representation and ordering

In DPAP [3], a candidate solution X consists of N items. Its j-th item has two parts, (x_j, y_j) and P_j, which represent the location and the transmit power level of sensor j, respectively. The items of a solution X are ordered as follows: the sensor locations in X are sorted based on their distance to H, where *1* is the closest and N is the farthest sensor location with respect to H, respectively. This results in having the locations of the sensors that are densely deployed around H at the beginning of each solution and the locations of the sensors that are spread away at the end. Thereinafter, each sensor j is assigned a transmit power level P_j proportional to $R_c^j \leqslant R_{max}$, such that it reaches its closest neighbor sensor, e.g. k, where $k < j$.

3. The problem-specific Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D)

This section details the proposed MOEA/D operators designed for tackling the K-connected DPAP. Note that, the underlying idea behind the problem-specific operators might shed some light on the design of MOEA/Ds for other MOPs.

3.1. Decomposition and analysis

Initially, MOEA/D needs to decompose a MOP into a set of subproblems. Any decompositional technique can serve for this purpose [20]. In this paper, the Weighted Sum approach is used, as follows. The multi-objective DPAP is decomposed into m scalar optimization subproblems considering two objectives. The ith scalar optimization subproblem can be defined as:

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

where λ^i is the weight coefficient of subproblem $i = 1, \ldots, m$. For the remainder of this paper, we consider a uniform spread of the weights λ^i, which remain fixed for each i for the whole evolution and are determined as follows:

$$\lambda^i = 1 - (i/m),$$

for $i = 2, \ldots, m$ and $\lambda^1 = 1$. Hence, the λ^i coefficient is mainly utilized for decomposing a MOP into a set of scalar subproblems by adding different weights to the objectives.

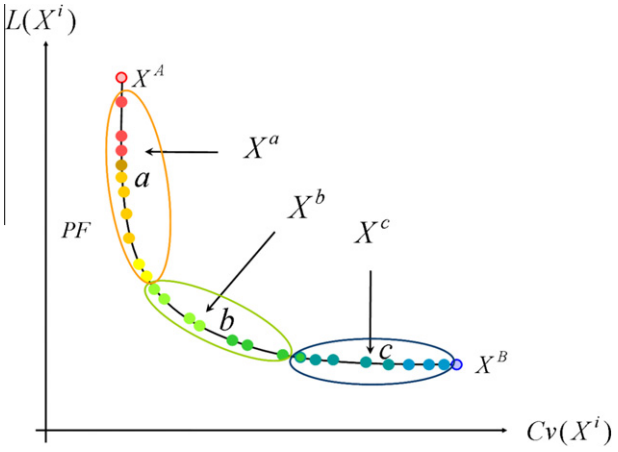


Fig. 2. Classifying the optimal network designs in DPAP.

In this paper, we have also given a problem-specific meaning to this parameter. Considering the λ^i weight coefficient of a subproblem i, we can predict the objective preference of a particular design and therefore, its position in the objective space. For example, the *extreme solutions* X^A and X^B in Fig. 2 focus at optimizing one objective each. The extreme solution X^A provides the maximum lifetime and minimum coverage and the extreme solution X^B provides the maximum coverage and minimum lifetime among all the solutions in PF, respectively. The goal of DPAP, however, is to provide the interested users with a diverse set of network design choices (e.g. X^a, X^b, and X^c in Fig. 2), giving the trade-off between the extreme solutions X^A and X^B. However, the procedure of designing non-extreme topologies is complicated, since there is not a scalar method which can design all of them in a single run. However, appropriate scalar strategies can be employed and controlled to optimize different feasible areas of the objective space accordingly. Note that, this beneficial procedure cannot be utilized by any non-decompositional MOEA framework.

3.2. MOEA/D: an overview

A general MOEA/D approach usually proceeds as in Algorithm 1.

Algorithm 1. The MOEA/D general framework

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

Output: the external population, EP.
Step 0 – Setup: Set $EP := \emptyset$; $gen := 0$; $IP_{gen} := \emptyset$;
Step 1 – Initialization: Uniformly randomly generate an initial internal population $IP_0 = \{X^1, \ldots, X^m\}$;
Step 2: For $i = 1, \ldots, m$ **do**
  Step 2.1 – Genetic Operators: Generate a new solution Y using the genetic operators.
  Step 2.2 – Repair heuristic: Apply a problem-specific repair heuristic on Y to produce Z.
  Step 2.3 – Update Populations: Use Z to update IP_{gen}, EP and the T closest neighbor solutions of Z.
Step 3 – Stopping criterion: If stopping criterion is satisfied, i.e. $gen = gen_{max}$, **then** stop and output EP, **otherwise** $gen = gen + 1$, go to Step 2.

Please cite this article in press as: A. Konstantinidis, K. Yang, Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.08.011

The following remarks are related to the MOEA/D framework:

- The internal population IP_{gen} of size m keeps the best solution found so far for each subproblem. The initial solutions of IP_0 are generated either randomly [3] or by a problem-specific heuristic (which will be introduced in Section 3.3).
- Solution Y is generated by using a selection operator (which will be detailed in Section 3.4.1) to choose two parent solutions from the IP_{gen}, e.g. Pr_1, Pr_2, a crossover operator (which will be detailed in Section 3.4.2) to produce a new solution from Pr_1, Pr_2 and a mutation operator (which will be detailed in Section 3.4.3) to modify the new solution Y. Solution Z is produced by using a repair method on Y (which will be detailed in Section 3.5).
- The T closest neighbor solutions of Z are the solutions of the T closest subproblems of i in terms of their weights $\{\lambda^1,\ldots,\lambda^m\}$. This is commonly known as the T neighborhood of subproblem i.
- The external population EP stores all the non-dominated solutions found so far during the search.

In the following the MOEA/D-based, DPAP-specific operators are presented.

3.3. Population initialization

In Step 1 of MOEA/D, we adopt a problem-specific heuristic to generate m solutions for the initial internal population (i.e. IP_0). The proposed heuristic seeds the initial solutions into promising feasible regions of the objective space to improve the overall performance of the MOEA/D.

In DPAP, we have m subproblems, where X^i is the solution of subproblem $i = 1,\ldots,m$. Following the discussion in Section 3.1, solution X^1 should have the highest lifetime, solution X^2 a better coverage than X^1 but a worse lifetime and so on, where solution X^m should have the highest coverage. Thus, the sensors of X^1 should be densely deployed in a small area A^1 around H, where for X^2 the area A^2 should be larger than A^1 so that the sensors are more sparsely deployed to benefit the coverage objective and so on. Finally, for X^m the sensors should be sparsely deployed in the whole area $A^m = A$. In this paper, the subarea $A^1 \subseteq A$ is defined as follows:

- Assuming unit side length grids and that each grid can host only one sensor then

$$\begin{aligned} &A^1 \subseteq A \text{ is a 2D area of length } x^1 \text{ and width } y^1,\\ &\text{centered at location } (x_H, y_H),\\ &\text{with size } x^1 \times y^1 = N\\ &\text{and } x^1 = \frac{x}{y} \times y^1, \end{aligned} \tag{8}$$

where $A = [0,x] \times [0,y]$ is the whole area of length x and width y.

Thereinafter, for the remaining subproblems $i = 2,\ldots,m$ the subarea A^i, that should gradually increase as the i increases based on the weight coefficient λ^i, is defined as follows:

- Assuming unit side length grids and that each grid can host only one sensor then

$$\begin{aligned} &A^i \subseteq A \text{ is a 2D area of length } x^i \text{ and width } y^i,\\ &\text{centered at location } (x_H, y_H),\\ &\text{where } x^i = x^1 + (x - x^1) \times (1 - \lambda^i)\\ &\text{and } y^i = y^1 + (y - y^1) \times (1 - \lambda^i), \end{aligned} \tag{9}$$

where for solution X^m of subproblem m with $\lambda^m = 0$ the $A^m = A$. X^m should be a good approximation of solution X^B with the highest coverage.

For each subproblem $i = 1,\ldots,m$, an initial solution X is generated using Algorithm 2.

Algorithm 2. An initial solution X for a subproblem i

Input: The size of a solution, N;
Output: A solution X;
Step 0: Set $X := \emptyset$;
Step 1: Calculate

$$A^i = \begin{cases} \text{Eq. (8)}, & \text{if } i = 1,\\ \text{Eq. (9)}, & \text{otherwise}, \end{cases}$$

Step 2: Uniformly randomly generate N locations $(x_j, y_j) \in A^i$ and add them in X;
Step 3: Order solution X as in Section 2.3;
Step 4: Assign transmit power levels P_j, where $j = 1,\ldots,N$, to solution X, as in Section 2.3;
Step 5: Output X;

- The proposed population initialization constrains the initial locations within a subarea A^i, which is calculated based on N and the weight coefficient λ^i.

Remark 1. The random population initialization methods are usually "simple" and fast. Thus, they may be efficient in cases like [3] in which the entire objective space is feasible. In constrained MOPs, a random method often obtains low quality feasible solutions and/or infeasible solutions, which has a negative impact on the performance of the algorithm.

Remark 2. The problem-specific population initialization methods are usually more computationally expensive than random methods. In constrained MOPs, however, there might be a trade-off between the computational increase of MOEA/D due to a problem-specific population initialization, and the computational increase due to a high number of repair function evaluations on the infeasible solutions obtained by the random process. A high number of infeasible solutions usually requires a high number of repair function evaluations, which often increases the overall computational effort of the algorithm.

Remark 3. The proposed problem-specific population initialization aims at: (a) improving the performance of MOEA/D by obtaining good, feasible initial solutions and (b) reducing the probability of obtaining infeasible solutions, with respect to an uniformly random method. This probability increases as λ^i decreases.

3.4. Genetic operators

In the ith pass of the loop in Step 2 of the MOEA/D, the genetic operators generate a new solution in Step 2.1. Note that, the same genetic operators are initially proposed in [3] and successfully applied to the unconstrained DPAP. However, it is our belief that their problem-specific nature might reduce the generation of infeasible solutions (i.e. do not satisfy the K-connectivity constraints) and direct the search into promising feasible regions of the objective space. Therefore, they are adopted and introduced from a feasibility point of view in the following, for clarity.

3.4.1. Selection operator

The first genetic operator in Step 2.1 is the M-tournament selection operator (denoted as M-tourS) that combines a mating restric-

Please cite this article in press as: A. Konstantinidis, K. Yang, Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.08.011

tion [20] and a standard tournament selection [33]. M-tourS, proceeds as in Algorithm 3.

Algorithm 3. The M-tournament selection operator (M-tourS) for each subproblem i

Input: A population of solutions, IP_{gen};
Output: Two parent chromosomes, Pr_1, Pr_2;
Step 1: Select the solutions $X \in IP_{gen}$ of the M closest subproblems of i to compete in the tournament;
Step 2: Evaluate each solution X of the tournament in terms of $g^i(X,\lambda^i)$;
Step 3: Find the best two solutions of the tournament, set them as Pr_1, Pr_2 and stop;

The aim of M-tourS is to select solutions of the M closest subproblems of a subproblem i in IP_{gen}, in terms of the Euclidean distance of their weights $\{\lambda^1,\ldots,\lambda^m\}$, which are called X^i's neighbors and compete to a tournament. In that case, X^i's neighbors, e.g. X^j and X^k, are competing in i's tournament in terms of $g^i(X,\lambda^i)$, ignoring their own λ^j and λ^k, their Pareto domination and/or ranking. In this way, more selection pressure is provided towards the optimal point of each particular i for better exploitation of the feasible regions of the objective space. That is, the optimization of a network design X^i, should mainly acquire good topological information from a neighbor network design X^j; instead of a network design X^m which is far away in the weight space. This is due to the highly non-linear multi-hop nature of WSNs. A tiny change in the topology may lead to a big change on the objective values due to the exponential relationship between the sensors transmission distance and energy consumption, as well as disconnections and/or partitions of the network. According to Section 3.1, the solutions of the subproblems in area a are mostly dense network designs comprised of sensors located close to H with low transmit power levels to facilitate the lifetime objective. In contrast, the solutions of the subproblems in area c are spread network designs comprised of sensors spread along the sensing field with high transmit power levels to facilitate the coverage objective. Thus, by combing two solutions that are far away in the objective space (e.g. a dense deployment and a spread deployment) might cause deterioration of the objective values and/or violation of the constraints.

The two selected parent solutions Pr_1 and Pr_2 are then forwarded for recombination to the crossover operator.

3.4.2. Crossover operator

In Step 2.1 (Algorithm 1), the crossover combines the two parents Pr_1 and Pr_2 to generate a new solution-the offspring denoted as O, with a probability rate r_c. In this paper, the adaptive crossover operator (denoted as aX) is adopted [3], which probabilistically controls two crossover strategies (i.e. window and clustering) each favoring different feasible areas of the objective space.

The window crossover control parameters (behaviors) change dynamically from subproblem to subproblem based on instant requirements. To do so, it determines a "window" of size:

$$w^i := N + N \times (1 - \lambda_i), \tag{10}$$

to select promising genetic material from each parent and direct the search into promising feasible areas of the search space for each particular i. That is, when λ^i is large and $L(X)$ favors $Cv(X)$, the window is small such that the sensor locations that will be added in O are as close to H as possible with low transmit power levels to provide higher network lifetime and maintain K-connectivity. When λ^i decreases and $Cv(X)$ starts favoring $L(X)$, w^i gradually increases to give the chance to the sensor locations which are spread in A to be added in O and therefore to provide better network coverage. The window always start at position 1 of solution U to always include the sensor locations of the "dense" part of the network (i.e. close to H) and therefore to maintain the connectivity as the sensor locations spread in the topology.

Algorithm 4. Adaptive crossover operator for each subproblem i

Input: Two solutions Pr_1 and Pr_2;
Output: A solution O;
Step 1:

$$Set\ \delta = \begin{cases} 1, & \text{if } \lambda^i \geqslant 0.5, \\ \lambda^i + 0.1, & \text{if } 0.3 < \lambda^i < 0.5, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Step 2: Uniformly randomly generate a number *rand* from $[0,1]$
Step 3: If *rand* $< \delta$ **then**//Apply Window
  Step 3.1: Set $O = \emptyset$; $U = Pr_1 \cup Pr_2$;
  Step 3.2 Order solution U as in Section 2.3;
  Step 3.3: Uniformly randomly generate an integer j from $\{1,2,\ldots,[w^i]\}$, where w^i is defined as in Eq. (10);
  Step 3.4 If there exists a (x_j,y_j) in $U = \{(x_1,y_1),(x_2,y_2),\ldots,(x_{2N},y_{2N})\}$ **then**
    Step 3.4.1: Delete (x_j,y_j) from U and add it in O;
    Step 3.4.2: If the size of O is not N **then** goto Step 3.3;
      otherwise stop and output O;
Step 4: Else//Apply clustering:
  Step 4.1: Set $O = Pr_1 \cup Pr_2$; $d' = d_c$;
  Step 4.2: Order infeasible solution $O = \{(x_1,y_1),\ldots,(x_{2N},y_{2N})\}$ as in Section 2.3;
  Step 4.3: For $j = 1$ to $2N$
    While $(x_j,y_j) \in O$ and $\exists (x_k,y_k) \in O | d_{jk} \leqslant d'$ **do**;
      Step 4.3.1: Uniformly randomly delete either location (x_j,y_j) or (x_k,y_k) from O;
      Step 4.3.2: If the size of O is equal to N **then** stop and output O;
    End while
  Step 4.4: Set $d' = d' + d_c$ and goto Step 4.3;

However, the window crossover has some undesirable effects for low weights (e.g. λ^m) and particularly for area c of the objective space, generating infeasible offspring [3]. Particularly, when $\lambda^i \to 0$ then $w^i \to 2 \times N$, which basically drives the crossover operation into an uniform random selection of sensor locations from the merged solution U. In that case, there is a high probability of selecting locations which are far away to each other, resulting in a high number of disconnected sensors, violating the K-connectivity constraint. This is not beneficial for the particular subproblems and consequently for feasible offspring reproduction of network designs that require high coverage quality. The clustering crossover overcomes this undesirable effect and obtains feasible network topologies of high coverage. That is, each sensor j at location $(x_j,y_j) \in O$ represents a cluster, having d' as the minimum Euclidean distance measure between each cluster. Two clusters centered at locations (x_j,y_j) and (x_k,y_k) are merged if $d_{jk} \leqslant d'$. In that case, either location (x_j,y_j) or (x_k,y_k) is deleted from O. This continues until N locations remain in O. Otherwise, if there are no more locations with $d_{jk} \leqslant d'$ and the size of solution O is less than N, the Euclidean distance measure is increased to further spread the locations in the solution.

The proposed adaptive crossover composed of the window and the clustering crossovers just described is outlined in Algorithm 4. In this kind of crossovers, different mechanisms are adopted with a probability δ for producing a new solution, where $\delta = \lambda^i$ means that

Please cite this article in press as: A. Konstantinidis, K. Yang, Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.08.011

two crossover strategies are almost equally applied in each generation. In this paper, we suggest a δ probability such that the window crossover is applied with highest probability in areas a and b and the clustering in area c to maintain the K-connectivity constraint for the reasons just mentioned.

3.4.3. Mutation operator

The last operator in Step 2.1 of MOEA/D (Algorithm 1 in Section 3.2) is the mutation, which is responsible for maintaining the diversity of the population by modifying the locations of a solution O with a r_m probability. However, the choice of the new location should be carefully determined, since an improper choice may damage all the preceding actions of the problem-specific selection and crossover operators, resulting in deterioration of the objective values and violation of the K-connectivity constraints. That is, disconnected sensors, partition of the network, since the deletion of a sensor in multi-hop communication may disconnect other parts of the network.

Thus, it is considered reasonable to allow the mutation operator to randomly modify the locations of a solution with an r_m probability, but restricting the modification to close to the current value or at least to bias the probability distribution in its favor. This may maintain the diversity of the population without destructive behavior (i.e. generating infeasible solutions) or unnecessary searches. The adaptive mutation operator that proceeds as in Algorithm 5 is composed of two problem-specific mutation strategies, namely the local and global mutations that favor different feasible areas of the objective space, respectively.

Algorithm 5. Adaptive mutation operator for each subproblem i

Input: A solution O
Output: A mutated solution Y
Step 0: Set r_m;
Step 1: Order solution O as in Section 2.3;
If $\lambda^i > 0.5$ **then**
Step 2: For $j = 1$ to N **do**
Step 2.1: Generate an uniform random number $rand \in [0,1]$;
Step 2.2: If $rand \leqslant r_m$ **then**
Calculate $\left(x'_j, y'_j\right)$ using Eq. (12). Replace $(x_j,y_j) \in O$ with $\left(x'_j, y'_j\right)$;
Else
Step 3: For $j = 1$ to N **do**
Step 3.1: Generate an uniform random number $rand \in [0,1]$;
Step 3.2:If $rand \leqslant r_m$ **then**
Calculate A' and $\left(x'_j, y'_j\right)$ using Eq. (13). Replace $(x_j,y_j) \in O$ with $\left(x'_j, y'_j\right)$;
End if
Step 4: Output $Y = O$;

If λ^i favors the lifetime objective (i.e. area a and the beginning of area b) then a location (x_j,y_j) is modified "locally", i.e.:

$$\begin{aligned}&\text{Uniformly randomly generate } x'_j \in [x_j - d_c, x_j + d_c]\\ &\text{and } y'_j \in [y_j - d_c, y_j + d_c],\end{aligned} \tag{12}$$

to provide a minimum shift from its current position, where d_c is the distance between the centers of two adjacent diagonal grids. This may result in improving $Cv(X)$ in the sake of increasing P_j, when the shift is backward with respect to H or, improving $L(X)$ by decreasing the P_j as well as decreasing the probability of violating the K-connectivity constraint. If λ^i favors the coverage objective (i.e. the end of area b and area c) then a location (x_j,y_j) is modified "globally", i.e. a new location $\left(x'_j, y'_j\right)$ is generated in a subarea $A' \subseteq A$ which is defined as follows:

$$\begin{aligned}&x_{min} = (x_H - |x_H - x_j|) - R_{max}, \quad y_{min} = (y_H - |y_H - y_j|) - R_{max};\\ &x_{max} = (x_H + |x_H - x_j|) + R_{max}, \quad y_{max} = (y_H + |y_H - y_j|) + R_{max};\\ &x' = x_{max} - x_{min}, \quad y' = x_{max} - x_{min};\\ &A' \subseteq A \text{ is a 2D area with length } x' \text{ and width } y'.\\ &\text{Uniformly randomly generate} \left(x'_j, y'_j\right) \in A';\end{aligned} \tag{13}$$

where x' and y' are the width and height of A', respectively. Note that when $\lambda \to 0$ then it should be that $A' \to A$. In that case the sensors are already spread in the topology having a high probability that the modified sensor will find a path towards H and not violate the constraints. The modified offspring is then forwarded to the repair operator. For more details on the genetic operators and their interrelation please refer to [3].

3.5. Constraint handling: repair heuristic

In K-connected DPAP, a solution X is infeasible if $Cn(X) \neq 1$, i.e. there exists at least one sensor j at location $(x_j,y_j) \in X$ with $c_j = 0$ (defined in Section 2.1). Thus, a repair heuristic is designed for identifying and transforming an infeasible solution into a feasible one (i.e. move it to a feasible region of the search space). An infeasible solution can be generated during initialization and/or reproduction of an offspring solution. A good repair heuristic should have the following properties:

- the repaired solution should be as close as possible to the infeasible solution, so that the search space is more efficiently explored.
- the objective functions should not be deteriorated.

Algorithm 6. The DPAP-specific Repair Heuristic (RH) for a subproblem i

Input: A solution X;
Output: A feasible solution Y;
Step 0: Set K; s;
Step 1: if $Cn(X) = \begin{cases} 1, & \text{goto Step 2;} \\ 0, & \text{goto Step 5;} \end{cases}$
Step 2: Find the origin of infeasibility using Eq. (3), i.e. a sensor j at $(x_j,y_j) \in X$ with $c_j = 0$;
Step 3: if $\lambda^i \geqslant 0.5$ **then**
Step 3.1: Divide the circle with radius $r = R_{max}$ centered at (x_H,y_H) into s equal sectors;
Step 3.2: Find the sparsest sector;
Step 3.3: Uniformly randomly generate a location $\left(x'_j, y'_j\right)$ within the sparsest sector. Replace $(x_j,y_j) \in X$ with $\left(x'_j, y'_j\right)$ and set $P_j = (d_{jH})^{\alpha}$;
Else
Step 3.4: Find the K^{th} closest location to (x_j,y_j), e.g. $(x_v,y_v) \in X$;
Step 3.5: Calculate a new location $\left(x'_j, y'_j\right) \in A$ using Eq. (14). Replace $(x_j,y_j) \in X$ with $\left(x'_j, y'_j\right)$ and set $P_j = \left(R_c^j\right)^{\alpha}$;
EndIf
Step 4: If $\exists j | (x_j,y_j) \in X$, $c_j \neq 1$ **then** goto Step 2;
Step 5: Output $Y = X$;

Please cite this article in press as: A. Konstantinidis, K. Yang, Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.08.011

Based on these properties, we design the DPAP-specific Repair Heuristic (denoted as RH), Algorithm 6, in which the weight coefficient λ^i of subproblem i plays an important role.

- In Step 1, the heuristic checks whether a solution is feasible or infeasible (i.e. calculates $Cn(X)$ as in Eq. (6) defined in Section 2.1).
- In Step 2, the origin of infeasibility is identified, i.e. a sensor j at location $(x_j,y_j) \in X$ with $c_j = 0$ (given by Eq. (3) defined in Section 2.1).
- In Step 3, when λ^i is high, and subproblem i requires a feasible solution with long network lifetime, the *RH*:
  Step 3.1: divides the circle with radius $r = R_{max}$ centered at (x_H,y_H) into s equal sectors (e.g. $s = 4$).
  Step 3.2: finds the sparsest sector (i.e. the sector with the lowest number of sensors).
  Step 3.3: Replaces location (x_j,y_j) with an uniform random location (x'_j,y'_j) within the sparsest sector, so that $R^j_c = d_{jH} \leqslant R_{max}$ and sets $P_j = \left(R^j_c\right)^{\alpha}$.
  In that case, while the RH is repairing an infeasible network design it might also provide the following benefits:
  – Supports the network load balancing and prevents a premature energy elimination of the sensors that are already directly connected to H, increasing the network lifetime (Fig. 3(a), for $K = 1$).
  – Covers any previously uncovered area close to H, increasing the network coverage without decreasing the network lifetime (Fig. 3(b), for $K = 1$).
- In Step 3, when λ^i is low, subproblem i requires a feasible solution with high network coverage, the RH:
  Step 3.4: finds a location (x_v,y_v), which is the K^{th} closest location to (x_j,y_j) and sensor v is a positive-advance neighbor of sensor j (i.e. v is closer to H than j [30]). If there does not exist such a location (x_v,y_v) then RH considers (x_H,y_H) as the closest location to (x_j,y_j) and sets $R^j_c = R_{max}$ to repair the infeasibility.
  Step 3.5: Replaces location (x_j,y_j) with a new location (x'_j,y'_j) calculated as follows:

$$\left(x'_j,y'_j\right) = (x_j,y_j) + (d_{ju} - R^j_c) \times [(x_u,y_u) - (x_j,y_j)]/d_{jx}, \qquad (14)$$

where

$$u = \begin{cases} v, & \text{if } c_j = 1,\ d_{jv} \leqslant d_{jH}, \\ H, & \text{otherwise}, \end{cases}$$

$$R^j_c = \begin{cases} 2R_s, & \text{if } R_{max} \geqslant 2R_s, \\ R_{max}, & \text{otherwise}, \end{cases} \qquad P_j = \left(R^j_c\right)^{\alpha}.$$

This results in low sensing range overlaps between the sensors that might increase the network coverage while repairing the infeasible solution (Fig. 3(c), for $K = 1$).

The repaired solution is then used to update the internal (*IP*) and external (*EP*) populations as follows.

3.6. Update of populations and termination criterion

In Step 2.3, the populations (defined in Section 3.2) are updated for each solution Z^i as follows: (1) The (IP_{gen}) update phase. If $g^i(Z^i|\lambda^i) > g^i(X^i|\lambda^i)$ then $IP_{gen} \cup \{Z^i\}$ and $IP_{gen}/\{X^i\}$, otherwise X^i remains in IP_{gen}. (2) The neighborhood (defined in Section 3.2) update phase. The new solution Z^i is compared with its T closest $X^j \in IP_{gen}$ neighbor solutions. If $g^j(Z^i|\lambda^j) > g^j(X^j|\lambda^j)$ then, $IP_{gen} \cup \{Z^i\}$ and $IP_{gen}/\{X^j\}$, otherwise, X^j remains in IP_{gen}, where $j = 1,\ldots,T$. (3) The (*EP*) update phase. $EP = EP \cup \{Z^i\}$ if Z^i is not dominated by any solution $X^j \in EP$, and $EP = EP/\{X^j\}$ if $Z^i \succ X^j$, for all $X^j \in EP$.

At the end of each generation the termination criterion (the maximum number of generations, gen_{max}) is checked to decide whether the search should stop.

4. Experimental setup

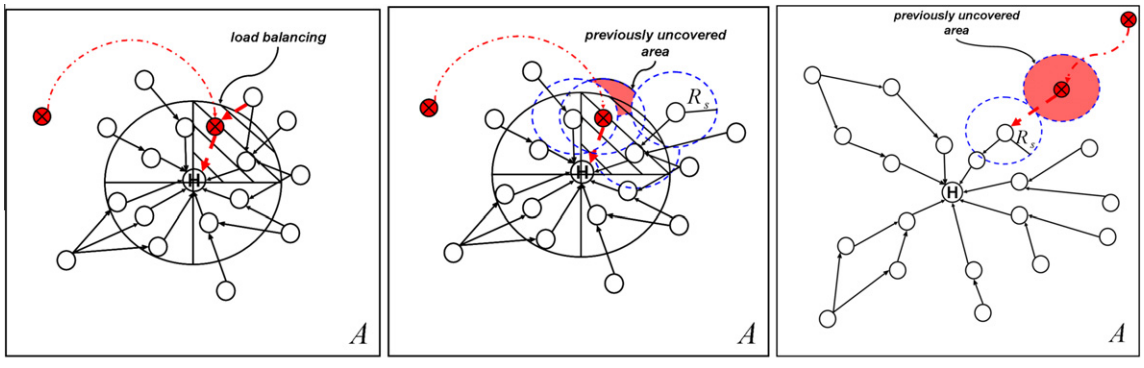
In this paper, we study six network test instances (Table 1), which represent a broad class of the small-scale and dense K-connected DPAP WSN topologies.

Moreover, we have fixed the parameter levels, following the experimental studies in [3], as follows: max number of generations, $gen_{max} = 250$, population size and # of subproblems $m = 120$, crossover rate $r_c = 0.9$, mutation rate $r_m = 0.5$, tournament size $M = 20$ and neighborhood size $T = 2$.

In all simulation studies, the following network parameters are set [7,34]: $R_s/R_{max} = 100/200$, $E = 5$ J, $d_{min} = 100$ m, $a = 2$, $amp = 100$ pJ/bit/m^2 and square-grids of side length 10 m. The network life-

Table 1
Network instances.

Network instances	A (m^2)	N	Density (N/A)
NIn1	2500 (50 × 50)	25	0.01
NIn2	2500 (50 × 50)	50	0.02
NIn3	2500 (50 × 50)	63	0.025
NIn4	10,000 (100 × 100)	100	0.01
NIn5	10,000 (100 × 100)	150	0.015
NIn6	10,000 (100 × 100)	250	0.025



(a) Repairing an infeasible solution, by increasing lifetime without affecting coverage. (b) Repairing an infeasible solution, by increasing coverage without affecting lifetime. (c) Repairing an infeasible solution, by increasing coverage without affecting lifetime.

Fig. 3. (a and b) Repairing an infeasible solution for subproblems with high λ^i. The circle centered at (x_H,y_H) with radius $r = R_{max}$ is divided into $s = 4$ sectors. The shadowed sector is the sparsest. (c) Repairing an infeasible solution for subproblems with low λ^i. The dash-dotted line indicates the redeployment and the dotted line indicates the new link discovered after redeployment.

Please cite this article in press as: A. Konstantinidis, K. Yang, Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.08.011

time and coverage are evaluated as in Section 2.2 and the lifetime objective is normalized by the $L(X^A)$ as in [35]. All algorithms were coded in Java programming language and run on an Intel/circledR Pentium 4 3.2 GHz Windows XP server with 1.5 GB RAM.

5. Performance metrics

This section briefly describes the performance metrics used for comparing the performance of MOEAs. Since MOEAs generate a set of solutions for approximating the PF, it is not easy to compare the algorithms performances and there is not a single metric that can satisfy all requirements. For this purpose, several metrics have been proposed [17,13,36], including the IGD-metric, the hypervolume measure, the Δ-metric, C-metric etc. In this paper, the DPAP network topologies are designed off-line and there is no critical need of real-time decision making. Thus, our main focus is at obtaining a good approximate PF within an acceptable computational effort. In the absence of the real PF of a constrained DPAP instance, the following four metrics are adopted:

The Δ-metric [17] measures the extent of spread achieved among the obtained solutions. In the case of two objectives, the Δ value of a set of candidate solutions A is defined as follows:

$$\Delta(A) = \frac{d_f + d_l + \sum |d_j - \bar{d}|}{d_f + d_l + |A|\bar{d}},$$

where d_f and d_l are the extreme Pareto optimal solutions in the objective space, d_j is the distance between two neighboring solutions and $\bar{d}$ is the mean of all the distribution. The smaller the $\Delta(A)$ metric is, the better the diversity performance of A. $\Delta(A) = 0$ means an uniform spread of solutions in the objective space.

A straightforward comparison metric between two sets of non-dominated solutions A and B is the C-metric [17,36]. The $C(A,B)$ metric, which is usually considered as a MOEA quality metric, evaluates the ratio of the non-dominated solutions in A dominated by the non-dominated solutions in B, divided by the total number of non-dominated solutions in A. Hence

$$C(A,B) = \frac{|A - \{x \in A | \exists y \in B : y \prec x\}|}{|A|}.$$

The smallest $C(A,B)$ is, the better the A. Note that $C(A,B) \neq 1 - C(B,A)$.

Another commonly used metric, usually considered in cases of real life discrete optimization problems [27,37], such as DPAP, is the number of Non-Dominated Solutions ($NDS(A)$) in set A, i.e.

$$NDS(A) = |A|.$$

In DPAP, it is very difficult to obtain many different NDS. Therefore, a high number of $NDS(A)$ is desirable to provide an adequate number of Pareto optimal choices. However, the NDS should be considered in combination with other metrics (e.g. Δ and C metrics), since it is usually desirable to have a high number of NDS when the solutions is of high quality and spread in the objective space, within an acceptable CPU effort. In contrast, and usually in cases of continuous optimization [20], a high number of NDS is not desirable, since the decision making procedure becomes more complicated and more time consuming.

6. Experimental results and discussion

The goals of our simulation studies are: (1) to demonstrate the difficulty in obtaining feasible solutions for the K-connected DPAP through a purely random process, (2) to test the strength of the proposed operators at improving the performance of MOEA/D at dealing with the K-connected DPAP and (3) to demonstrate the effectiveness of the proposed DPAP-specific MOEA/D-RH against the popular constrained NSGA-II in several WSN instances, giving the trade-off of the objectives and a variety of feasible network design choices.

6.1. Random method

To study the difficulty of the proposed K-connected DPAP, we use a purely random method in NIn1, 2 and 3 under a $K = 1,\ldots,5$ connectivity constraint. Specifically, we uniformly randomly generate 30,000 samples (i.e. network topologies) and we examine (i) the ϱ metric [38] that is defined as follows:

$$\varrho = |F|/|S|, \tag{15}$$

where $|F|$ is the number of feasible solutions, and $|S|$ is the total number of random solutions generated (i.e. $S = 30{,}000$), (ii) the total number of infeasible solutions and (iii) the total and (iv) average number of disconnected sensors.

The results of Table 2 show that the random method obtains solutions in the feasible region of the objective space only when $K = 1$. For $K = 2$ to 5 all 30,000 network designs are infeasible in all network instances (i.e. NIn1, NIn2 and NIn3). Moreover, when $K = 1$ and the density is low (i.e. NIn1, $N = 25$), there are only 2.55% feasible solutions, which means *29,235* out of 30,000 network designs are infeasible, having about 9/25 sensors disconnected per network design (i.e. about 36%). When the density is high (e.g. NIn3, $N = 63$) this number decreases to about 0.567/63 sensors, on average (i.e. 0.99%). This is the reason why sometimes

Table 2
Simulation results on $S = 30{,}000$ random network designs.

NIn	K	Infeasible sol.	ϱ(%)	Disconnected sensors	
				Total	Average
1	1	29,235	2.55	275,337	9.175
	2	30,000	0.0	511,665	17.05
	3	30,000	0.0	649,890	21.675
	4	30,000	0.0	717,879	23.925
	5	30,000	0.0	741,642	24.725
2	1	12,870	57.1	16,766	1.7
	2	30,000	0.0	110,425	11.05
	3	30,000	0.0	214,441	21.45
	4	30,000	0.0	321,518	32.15
	5	30,000	0.0	405,060	40.5
3	1	6030	79.9	5412	0.567
	2	30,000	0.0	82,646	8.253
	3	30,000	0.0	174,914	17.514
	4	30,000	0.0	293,468	29.358
	5	30,000	0.0	411,594	41.139

Please cite this article in press as: A. Konstantinidis, K. Yang, Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.08.011

it is assumed [39] that a dense sensor deployment implies network connectivity. Table 2, however, shows that even when the number of disconnected sensors is low, the $\varrho = 79.9\%$ indicates that a relatively high number of solutions is still infeasible, i.e. about 20.1% or 6030/30,000 solutions.

6.2. The effect of the DPAP Repair Heuristic (RH)

In this subsection, we study the effect of the proposed Repair Heuristic (denoted as RH) on MOEA/D and we evaluate its impact on dealing with the proposed DPAP. To do so, we compare the MOEA/D w/RH that adopts the proposed RH, with the MOEA/D w/SoFs and the MOEA/D w/PenF that replace the RH with the following constraint handling techniques, respectively:

- **Superiority of Feasible solutions (SoFs) [40]:** A comparison of two solutions X and Y is performed based on the following rules:
  - If X is feasible and Y is not feasible then select X.
  - If both X and Y are feasible then select the one with the highest scalar fitness.
  - If both X and Y are infeasible then select the one with the least constraint violation (i.e. number of disconnected sensors).

  It aims at favouring the good feasible, or least infeasible solutions to be copied in the next generation.
- **Penalty Function (PenF) [41]:** Transforms a constrained optimization problem into an unconstrained one by subtracting (in the case of DPAP) a certain value (also known as penalty, measured by a penalty function) from the scalar fitness value, based on the amount of constraint violation. It aims at favouring the feasible solutions over infeasible ones during the selection process. In the proposed DPAP there is only one constraint, i.e. the K-connectivity constraint, and the amount of violation is measured based on the number of sensors in the network that violate this particular constraint. The penalty of a solution X is measured as follows:

$$pn(X) = 1 - Cn(X),$$

where $Cn(X)$ is calculated as in Eq. (6) of Section 2.2. A constrained subproblem i can then be transformed into an unconstrained one as follows:

$$max\ g^i(X, \lambda^i) = [\lambda^i L(X) + (1 - \lambda^i) Cv(X)] - pn(X). \quad (16)$$

The three MOEA/Ds. are compared in NIn1–3 for $K = 1,\ldots,5$ and $S = m \times gen_{max} = 30{,}000$ in terms of the (i–iv) metrics, defined in Section 6.1. Note that the metrics are evaluated at the beginning of each generation for the MOEA/D w/PenF and the MOEA/D w/SoFs, and before repairing for the MOEA/D w/RH. Besides, the following standard genetic operators [33] are used, random population initialization, standard tournament selection operator (tourS), two-point crossover operator (2X), standard random mutation operator (rM).

The results of Table 3 show that MOEA/D w/RH helps the evolutionary process at obtaining feasible solutions in all network instances for most Ks. In contrast, MOEA/D w/SoFs and w/PenH perform poorly when the network is sparse (i.e. NIn1) and for high Ks when the network is dense. Specifically, MOEA/D w/RH obtains feasible solutions in NIn1 for all Ks. Moreover, MOEA/D w/RH found it difficult to direct the search into the feasible regions of the objective space for $K = 4, 5$ in NIn2, 3. MOEA/D w/SoFs generates 30,000 infeasible solutions ($\varrho = 0.0$) for $K = 2,\ldots,5$ in NIn1, for $K = 3,\ldots,5$ in NIn2 and for $K = 3,\ldots,5$ in NIn3. MOEA/D w/PenF performs slightly better than MOEA/D w/SoFs, since it obtains feasible solutions in NIn2, 3 for $K = 3$. MOEA/D w/RH generates less infeasible solutions and there exists a lower number of disconnected sensors in its solutions for sparse networks with respect to MOEA/D w/SoFs and MOEA/D w/PenF. The MOEA/D versions perform similarly for dense networks. More insights are given in the following example.

Fig. 4 illustrates the total number of disconnected sensors of the solutions obtained by MOEA/D per generation with each technique in NIn1, 2 and 3 for $K = 1$. In NIn1, all techniques start with about 1100 disconnected sensors out of a total of 3000 deployed sensors (i.e. $m \times N = 120 \times 25$). The latter is sharply decreased to about 600 disconnected sensors after one generation when RH is adopted and is smoothly decreased to about 800 disconnected sensors after about 20 generations when PenF and SoFs are adopted. This indicates that RH directs the search into the feasible regions of the search space more effectively. When the network becomes denser (i.e. NIn2, 3) the number of disconnected sensors decreases and the three techniques perform similarly. The total number of disconnected sensors in NIn2 is around 150 and in NIn3 is around 50. The effect of the hybridization of MOEA/D with the three techniques (i.e. RH, SoFs and PenF) is further studied in terms of the performance metrics introduced in Section 5, as follows.

Tables 4 and 5 show the comparison between the MOEA/Ds with the three constraint handling techniques in the network instances

Table 3
The feasibility results of MOEA/D w/RH, w/SoFs and w/PenF in NIn1, 2 and 3, $K = 1,\ldots,5$.

NIn	K	MOEA/D w/RH				MOEA/D w/SoFs				MOEA/D w/PenF			
		Infeasible	ϱ(%)	Disconnected sensors		Infeasible	ϱ(%)	Disconnected sensors		Infeasible	ϱ(%)	Disconnected sensors	
				Total	Average			Total	Average			Total	Average
1	1	27,751	7.5	153,857	5.12	27,941	6.9	197,800	6.59	27,892	7.0	196,318	6. 54
	2	19,665	34.5	52,634	1.754	30,000	0.0	313,096	10.43	30,000	0.0	159,560	5.31
	3	20,106	33.0	60,080	2.0	30,000	0.0	458,423	15.28	30,000	0.0	217,785	7.25
	4	20,448	31.8	70,083	2.33	30,000	0.0	558,941	18.63	30,000	0.0	242,473	8.08
	5	21,898	27.0	80,357	2.67	30,000	0.0	625,390	20.84	30,000	0.0	289,071	9.63
2	1	11,871	60.4	39,248	1.308	11,384	62.1	39,811	1.327	10,966	63.4	38,419	1.281
	2	29,843	0.5	188,221	6.274	29,549	1.5	138,796	4.627	29,725	0.9	148,505	4.950
	3	29,999	0.003	360,056	12.002	30,000	0.0	468,820	15.627	29,999	0.003	334,209	11.140
	4	30,000	0.0	454,796	15.160	30,000	0.0	760,618	25.354	30,000	0.0	575,897	19.197
	5	30,000	0.0	546,465	18.215	30,000	0.0	995,700	33.190	30,000	0.0	771,261	25.709
3	1	5206	82.6	13,027	0.434	4800	84.0	12,939	0.431	4729	84.2	12,417	0.414
	2	29,102	3.0	123,893	4.130	28,047	6.5	87,545	2.918	27,684	7.7	82,122	2.737
	3	29,989	0.04	267,333	8.911	30,000	0.0	332,153	11.072	29,987	0.04	249,124	8.304
	4	30,000	0.0	431,934	14.398	30,000	0.0	645,220	21.507	30,000	0.0	473,512	15.784
	5	30,000	0.0	527,812	17.594	30,000	0.0	974,595	32.486	30,000	0.0	715,402	23.847

Please cite this article in press as: A. Konstantinidis, K. Yang, Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.08.011

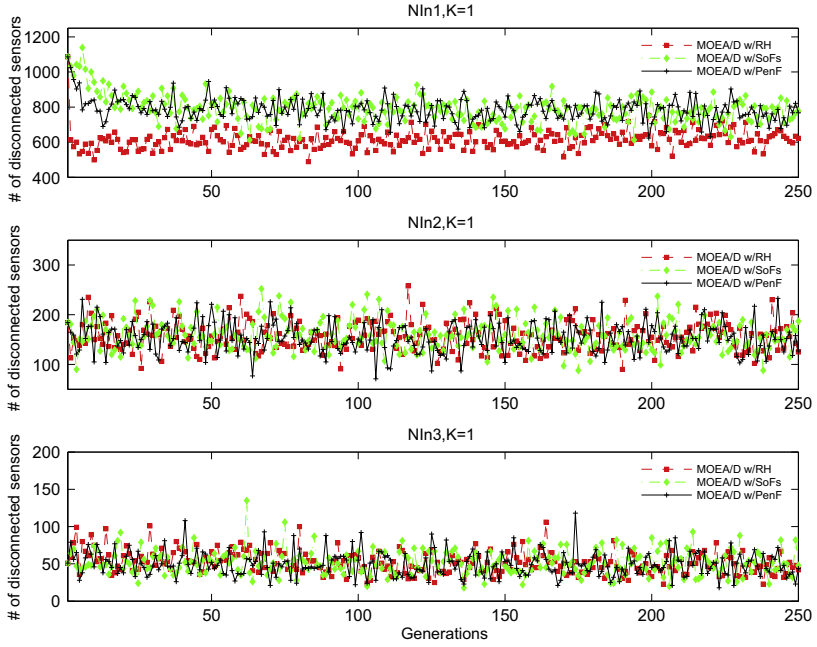


Fig. 4. The number of disconnected sensors per generation obtained by MOEA/D with RH, SoFs and PenF in NIn1–3, $K = 1$.

Table 4
Statistical results of MOEA/D with RH, SoFs and PenF constraint handling techniques in NIn1–3 of K-connected DPAP for $K = 1$. The best performance in each instance for each performance metric is indicated in bold.

Network instance	K	Δ			*NDS*			*CPU*		
		(w/RH)	(w/SoFs)	(w/PenF)	(w/RH)	(w/SoFs)	(w/PenF)	(w/RH)	(w/SoFs)	(w/PenF)
1	1	**0.9075**	0.9701	0.9627	**11**	7	7	0.1296	0.0976	**0.0815**
2	1	0.9776	**0.9642**	0.9797	7	**8**	5	0.2611	0.1514	**0.1137**
	2	**0.7781**	0.9273	0.9246	7	**9**	8	1.4930	1.0221	**0.9952**
	3	**0.8507**	–	0.9269	**9**	–	6	1.7614	1.5178	**1.4876**
3	1	0.9799	0.9787	**0.9763**	**7**	**7**	5	0.3194	0.2001	**0.1817**
	2	**0.7116**	0.9074	0.9326	9	**10**	6	1.6059	1.0213	**0.9344**
	3	**0.8249**	–	0.8950	**10**	–	6	1.9295	1.4998	**1.3567**
Average		**0.8614**	0.9495	0.9425	**8.57**	8.2	6.14	1.07	0.787	**0.7358**

Table 5
The C-metric results of MOEA/D with RH, SoFs and PenF constraint handling techniques in NIn1–3 of K-connected DPAP for $K = 1$. The best performance in each instance for each performance metric is indicated in bold.

NIn	K	C(w/RH, w/SoFs)	C(w/SoFs, w/RH)	C(w/RH, w/PenF)	C(w/PenF, w/RH)
1	1	**0.0**	0.8571	**0.0**	0.7143
2	1	**0.3333**	0.7143	0.5	**0.4**
	2	0.2857	**0.0**	0.2857	**0.0**
	3	–	–	**0.0**	0.1666
3	1	**0.0**	1.0	**0.0**	0.6
	2	0.2222	0.1250	**0.0**	0.3333
	3	–	–	0.7	**0.0**
Average		**0.1682**	0.5392	**0.2122**	0.3163

Table 6
Statistical results of MOEA/D w/RH+ (i.e. MOEA/D w/RH + proposed evolutionary operators) vs. MOEA/D w/RH (i.e. generic MOEA/D w/RH) in NIn1–3 of K-connected DPAP for $K = 1$. The best performance in each instance for each performance metric is indicated in bold.

Network instance	Δ		*NDS*		*CPU*		C	
	(w/RH+)	(w/RH)	(w/RH+)	(w/RH)	(w/RH+)	(w/RH)	(w/RH+, w/RH)	(w/RH, w/RH+)
1	0.9209	**0.9075**	5	11	0.1481	0.1296	**0**	0.9091
2	**0.9475**	0.9776	8	7	0.2594	0.2611	**0**	1.0
3	**0.9767**	0.9787	5	7	0.323	0.3194	**0**	1.0
Average	**0.9483**	0.9550	6	**8.3333**	0.2435	**0.2367**	**0**	0.9697

Please cite this article in press as: A. Konstantinidis, K. Yang, Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.08.011

where feasible solutions are obtained. The results show that RH is more beneficial to MOEA/D's performance than PenF and SoFs. Particularly, MOEA/D w/RH provides a better average Δ metric and the highest average number of NDS in the PF. In terms of quality, the non-dominated solutions obtained by MOEA/D w/RH dominate 53% and 31%, on average, of the non-dominated solutions obtained by MOEA/D w/SoFs and w/PenF, respectively. Its superiority comes at the cost of a slightly more computational effort.

6.3. The effect of the specialized genetic operators

In the previous subsection, MOEA/D w/RH used the generic evolutionary operators (i.e. tournament selection, two-point crossover, random mutation). In this subsection, we replace the generic operators with the proposed evolutionary operators (defined in Section 3.4) to study their effect on the K-connected DPAP and evaluate their impact on the performance of MOEA/D. Specifically, the MOEA/D composed of a random population initialization, the *M*-tournament selection operator, the adaptive crossover operator with the dense to spread ordering (defined in Section 2.3) and the adaptive mutation operator is combined with the *RH*. The new MOEA/D version, called MOEA/D w/RH + hereinafter, is compared with MOEA/D w/RH in NIn1, 2 and 3 for $K = 1$.

The results in Table 6 show an increase in the performance of MOEA/D in all network instances when the proposed problem-specific evolutionary operators are adopted. Specifically, MOEA/D w/RH+ provides a better average diversity and a higher quality of Pareto optimal solutions than MOEA/D w/RH, in all network in-

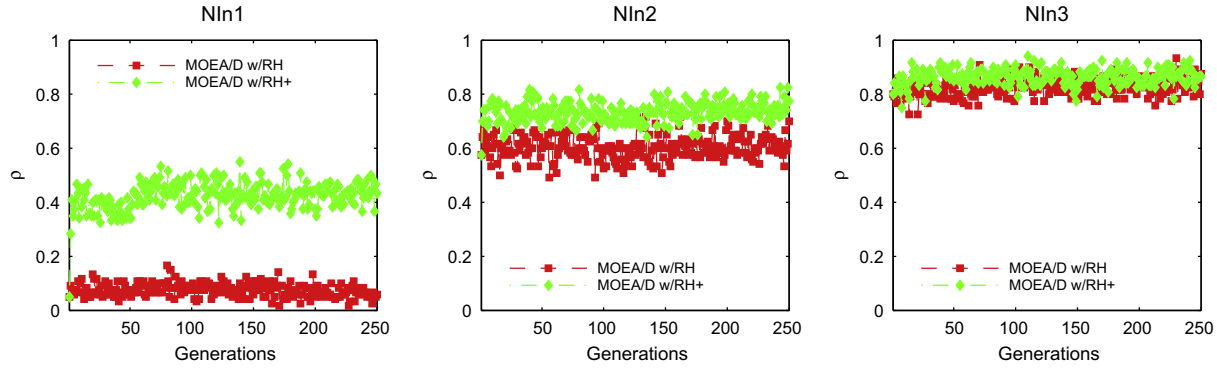


Fig. 5. MOEA/D w/RH (i.e. generic MOEA/D w/RH) vs. MOEA/D w/RH+ (i.e. MOEA/D w/RH+ proposed evolutionary operators) in terms of the ϱ-metric per generation in NIn1–3, $K = 1$.

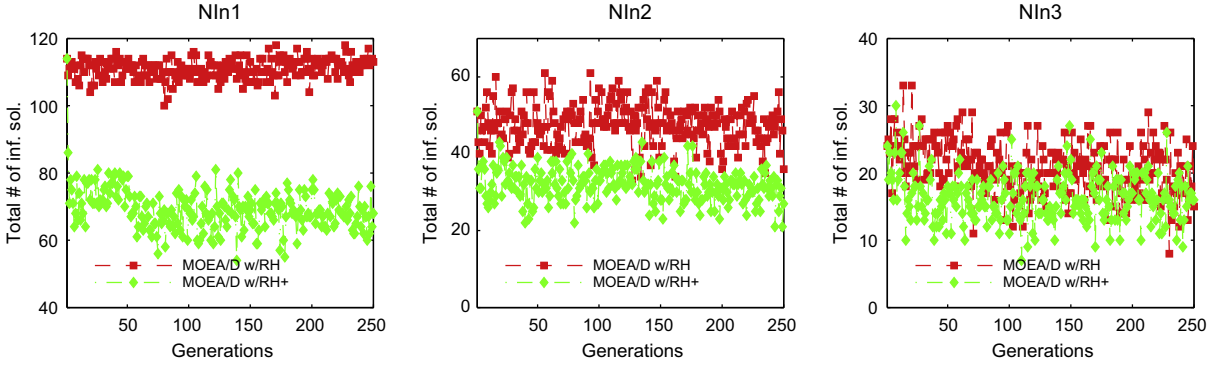


Fig. 6. MOEA/D w/RH (i.e. generic MOEA/D w/RH) vs. MOEA/D w/RH+ (i.e. MOEA/D w/RH+ proposed evolutionary operators) in terms of the total number of infeasible solutions per generation in NIn1–3, $K = 1$.

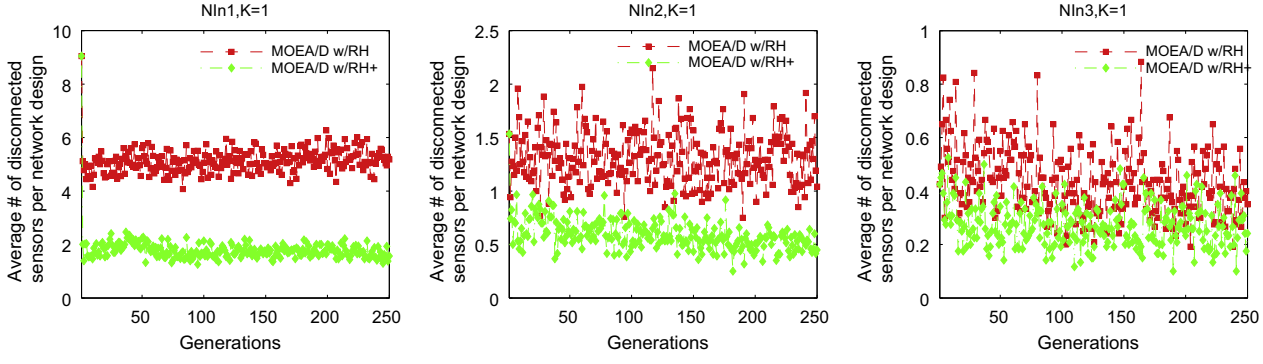


Fig. 7. MOEA/D w/RH (i.e. generic MOEA/D w/RH) vs. MOEA/D w/RH+ (i.e. MOEA/D w/RH+ proposed evolutionary operators) in terms of the average number of disconnected sensors/network design per generation in NIn1–3, $K = 1$.

Please cite this article in press as: A. Konstantinidis, K. Yang, Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.08.011

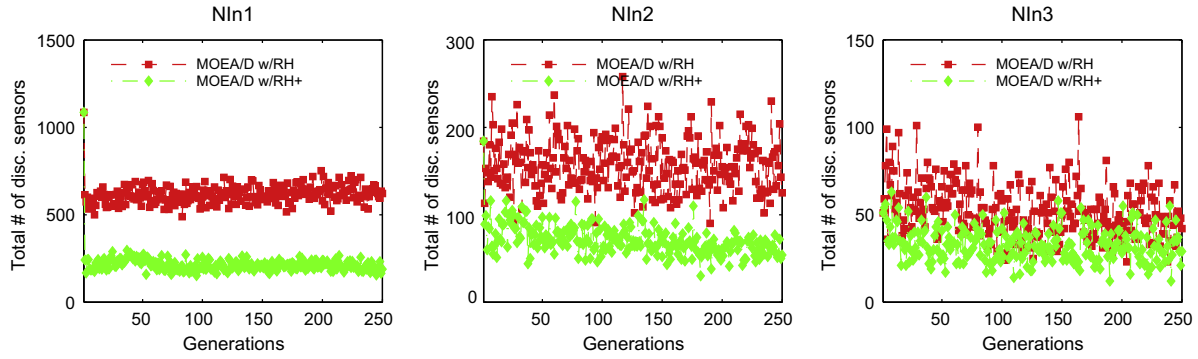


Fig. 8. MOEA/D w/RH (i.e. generic MOEA/D w/RH) vs. MOEA/D w/RH+ (i.e. MOEA/D w/RH+ proposed evolutionary operators) in terms of the total number of disconnected sensors per generation in NIn1–3, $K = 1$.

stances. MOEA/D w/RH+ performs slightly worse in terms of the number of NDS and the CPU effort with respect to MOEA/D w/RH.

In the following, the feasibility of the solutions obtained by the MOEA/D w/RH+ with respect to those obtained by MOEA/D w/RH is studied. The results are illustrated in Figs. 5–8. Fig. 5 show an average increase of 30%, 15% and 10% on the number of feasible solutions obtained for NIn1–3, respectively. Specifically, Fig. 6 shows that MOEA/D w/RH+ obtains about 40, 20 and 10 more network designs in the feasible region of the objective space than MOEA/D w/RH in NIn1, 2 and 3, respectively. Moreover, Fig. 7 shows that MOEA/D w/RH+ decreases the average number of disconnected sensors by 15%, 3% and 1% in NIn1, 2 and 3, respectively, which means that the number of repair function evaluations is decreased, from around 600 to 250, 150 to 80 and 50 to 30, as it is shown in Fig. 8.

At this point, it is important to notice the high number of infeasible solutions obtained by the random population initialization method. This can be seen in all network instances as a peak point at the beginning of each scatter plot in Figs. 5–8. Generation $gen = 0$ of Fig. 5 shows a ϱ-metric of almost zero for NIn1, 0.6 for Nin2 and 0.8 for NIn3. In addition, Fig. 6 shows that the number of infeasible solutions of $gen = 0$ is close to 120, 50 and 25 for NIn1, 2 and 3, respectively. Fig. 7 shows that in $gen = 0$, around 35%, 6% and 2% of the total $N \times m$ sensors deployed per generation are disconnected. Finally, Fig. 8 shows that more than 1000 sensors are disconnected and require repairing in NIn1. In NIn2 and 3 the disconnected sensors are around 200 and 50, respectively. Therefore, in all cases, the feasibility of the initial population is relatively poor.

6.4. The effect of the proposed DPAP-specific population initialization

To overcome the drawback mentioned in the previous subsection, the proposed DPAP-specific population initialization (defined in Section 3.3) is further adopted. To do so, the MOEA/D w/RH++ approach is designed, which replaces the random population initialization of MOEA/D w/RH+ with the proposed population initialization. MOEA/D w/RH++ is compared with the MOEA/D w/RH+ in NIn1–3, for $K = 1$.

Fig. 9 shows the total number of disconnected sensors (before repairing) in the initial population for each subproblem. The results show the effectiveness of the proposed DPAP-specific population initialization on providing feasible solutions. The proposed operator seeds almost half of the initial network designs in the feasible region of the objective space in NIn1, around 85 network designs in NIn2 and almost all network designs (113/120) in NIn3. This should save repair function evaluations as well as direct the search into new feasible regions of the search space and consequently increase the diversity of the population. It is also important to notice that the proposed initialization obtains feasible solutions for most subproblems with high λ^i.

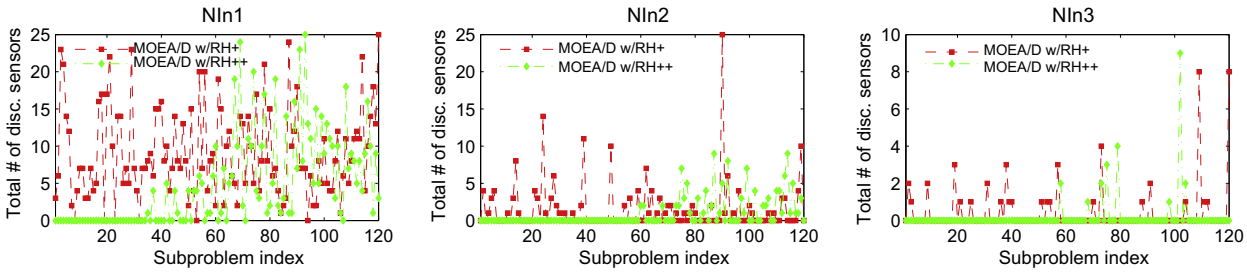


Fig. 9. MOEA/D w/RH+ (i.e. MOEA/D+ proposed evolutionary operators, random population initialization) vs. MOEA/D w/RH++ (i.e. MOEA/D+ proposed evolutionary operators + proposed population initialization) in terms of the total number of disconnected sensors per generation in NIn1–3, $K = 1$.

Table 7
Performance measure of MOEA/D with RH+ and RH++ in NIn1–3 of K-connected DPAP. The best performance in each instance for each performance metric is indicated in bold.

NIn	Δ (RH+)	Δ (RH++)	NDS (RH+)	NDS (RH++)	CPU (RH+)	CPU (RH++)	C (RH+,RH++)	C (RH++,RH+)
NIn1	0.9209	**0.8544**	5	**15**	**0.15**	0.17	**0.2000**	**0.2000**
NIn2	0.9475	**0.8733**	8	**14**	**0.26**	**0.26**	**0.1250**	0.3571
NIn3	0.9867	**0.8419**	5	**14**	**0.32**	0.33	**0.2000**	0.2857
Average	0.9517	**0.8565**	6	**14.33**	**0.243**	0.253	**0.175**	0.28

Please cite this article in press as: A. Konstantinidis, K. Yang, Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.08.011

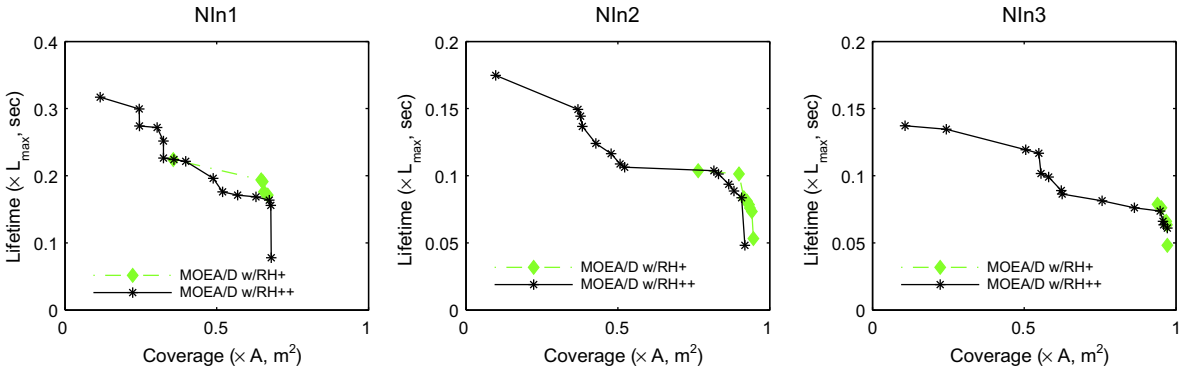


Fig. 10. MOEA/D w/RH+ vs. MOEA/D w/RH++ in NIn1–3.

Table 8
MOEA/D w/RH vs. MOEA/D w/RH++ in NIn1, 2 and 3, $K = 1,\ldots,5$.

NIn	K	MOEA/D w/RH				MOEA/D w/RH++			
		Infeasible	ϱ(%)	Disconnected sensors		Infeasible	ϱ(%)	Disconnected sensors	
				Total	Average			Total	Average
1	1	27751.0	7.5	153857.0	5.12	19086.0	36.38	52566.0	1.752
	2	19665.0	34.5	52634.0	1.754	17721.0	40.93	49141.0	1.638
	3	20106.0	33.0	60080.0	2.0	17601.0	41.33	43963.0	1.465
	4	20448.0	31.8	70083.0	2.33	18467.0	38.44	51322.0	1.711
	5	21898.0	27.0	80357.0	2.67	17142.0	42.86	44093.0	1.470
2	1	11871.0	60.4	39248.0	1.308	13492.0	55.027	34202.0	1.140
	2	29843.0	0.5	188221.0	6.274	24119.0	19.603	66267.0	2.209
	3	29999.0	0.003	360056.0	12.002	25325.0	15.583	87706.0	2.924
	4	30000.0	0.0	454796.0	15.160	25528.0	14.907	119312.0	3.977
	5	30000.0	0.0	546465.0	18.215	27246.0	9.18	175012.0	5.834
3	1	5206.0	82.6	13027.0	0.434	9932.0	66.893	28902.0	0.963
	2	29102.0	3.0	123893.0	4.130	25150.0	16.167	71813.0	2.394
	3	29989.0	0.04	267333.0	8.911	27379.0	8.737	111951.0	3.732
	4	30000.0	0.0	431934.0	14.398	28767.0	4.11	195061.0	6.502
	5	30000.0	0.0	527812.0	17.594	28899.0	3.67	223841.0	7.461

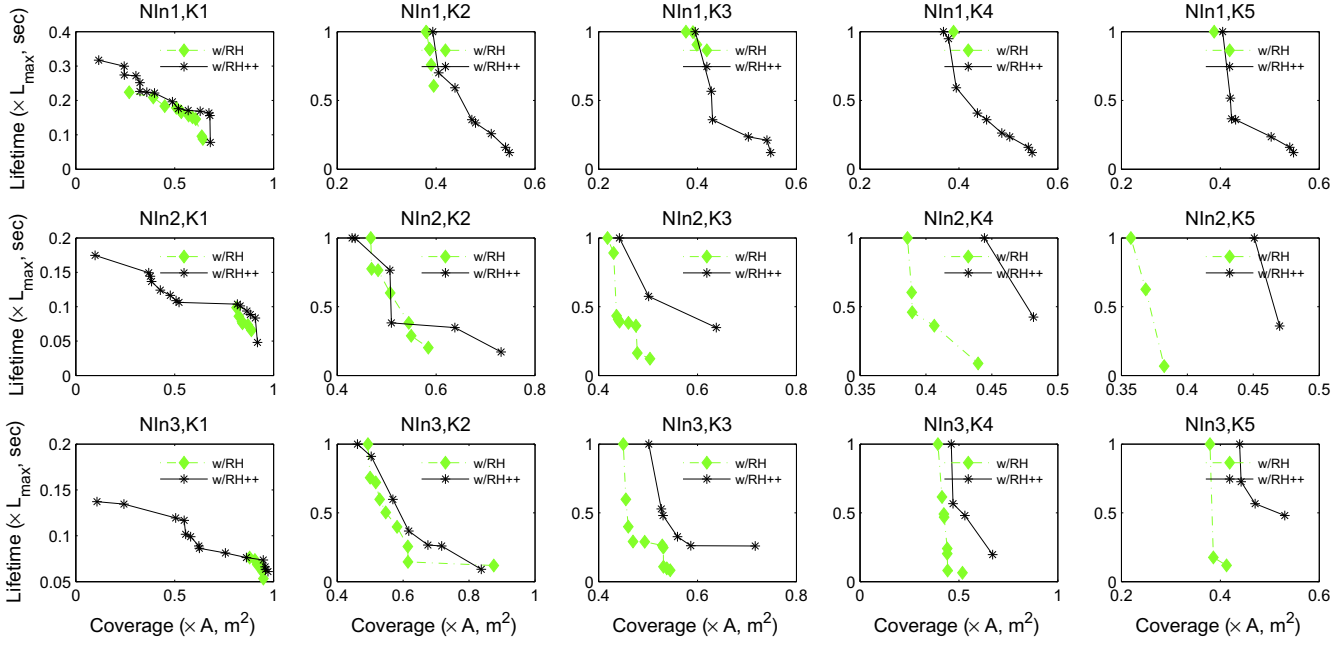


Fig. 11. MOEA/D w/RH vs. MOEA/D w/RH++ in NIn1, 2 and 3, $K = 1,\ldots,5$.

Please cite this article in press as: A. Konstantinidis, K. Yang, Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.08.011

Table 7 and Fig. 10 show the comparison between the MOEA/D w/RH++ and MOEA/D w/RH+ in NIn1–3, confirming the latter statement. The results show that the problem-specific population initialization directs the search into new feasible regions, resulting in a considerable increase on the diversity and the number of NDS. The performance of MOEA/D in terms of quality of solutions in the PF and CPU time remains relatively the same. The increase in the number of feasible solutions of the subproblems which prefer long network lifetime (high λ^i, area a), mentioned in the latter paragraph, is advantageous. Fig. 10 demonstrates the increase in the performance of the MOEA/D towards the aforementioned region of the objective space.

6.5. The performance of MOEA/D with all the proposed operators

Table 8 and Fig. 11 demonstrate the effectiveness of the MOEA/D w/RH++ (MOEA/D w/RH+ proposed evolutionary operators + proposed population initialization) with respect to the MOEA/D w/RH (i.e. MOEA/D w/RH, the generic operators, random population initialization) in NIn1–3, for $K = 1,\ldots,5$. Remember that the generic MOEA/D w/RH performs better than the generic MOEA/D with the generic constraint handling techniques, i.e. *SoFs* and *PenF* (please refer to Section 6.2 for more details). The results show that MOEA/D w/RH++ is currently the most efficient and effective MOEA/D version presented in this paper for the K-connected DPAP. MOEA/D w/RH++ performs better than MOEA/D w/RH in terms of feasibility (Table 8), i.e. obtains more feasible solutions before repairing, and in terms of quality (Fig. 11), in all network instances for all *K*s.

6.6. Comparison of MOEAs

In this subsection, we study the efficiency and effectiveness of the proposed problem-specific MOEA/D on the constrained DPAP. To do so, we compare the proposed method with the state of the art in MOEAs based on Pareto dominance. Namely, the constrained Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [17]. NSGA-II maintains a population IP_{gen} of size m at each generation gen, for gen_{max} generations. NSGA-II adopts the evolutionary operators (i.e. selection, crossover and mutation) for offspring reproduction as MOEA/D. The key characteristic of NSGA-II is that it uses a fast non-dominated sorting and a crowded distance estimation for comparing the quality of different solutions during selection and to update the IP_{gen} and the *EP*. We refer interested readers to [17] for details.

In this paper, NSGA-II adopts the following non-decompositional operators that have shown promising performance in [3]: the x-y axis ordering (xyOr) (Ordering-II), the standard tournament selection (tourS) (Selection-I), the two-point crossover (2X) (Crossover-II) and the random mutation (rM) (Mutation-I) as well as the Superiority of Feasible solutions (SoFs) constraint handling technique as proposed by Deb et al. [17]. For comparing the two MOEAs we have adopted both visual and statistical comparison, through the performance metrics introduced in Section 5, in all network test instances of Table 1.

Fig. 12 and Table 9 show the superiority of the proposed MOEA/D against the NSGA-II. MOEA/D performs better than NSGA-II in terms of quality, diversity and number of NDS in most network instances, at the cost of a higher computational effort. Particularly, MOEA/D provides a diversity of around 0.82 which is better than the 0.94 obtained by NSGA-II and six more NDS on average. The Pareto optimal solutions obtained by MOEA/D dominate all solutions obtained by NSGA-II and none is dominated. This indicates that NSGA-II has difficulties at obtaining good and feasible solutions for the K-connected DPAP and effectively shows the necessity of incorporating problem-specific knowledge in MOEA/D for obtaining a feasible, diverse and high quality set of Pareto optimal solutions.

7. Conclusions and future research

In this paper, the K-connected DPAP in WSNs is formulated as a constrained MOP and is decomposed into a set of scalar subproblems. The subproblems are classified based on their objective preferences and tackled by MOEA/D using problem-specific knowledge, simultaneously. A solution representation dedicated to DPAP and

Table 9
MOEA/D (M) vs. NSGA-II (N) in NIn1–16. The best performance in each instance for each performance metric is indicated in bold.

Net. ins.	Δ		NDS		CPU		C	
	(N)	(M)	(N)	(M)	(N)	(M)	(N,M)	(M,N)
NIn1	0.9439	**0.8417**	8	**14**	**0.20**	0.37	1.0	**0.0**
NIn2	0.9196	**0.8286**	10	**17**	**0.31**	4.99	1.0	**0.0**
NIn3	0.8991	**0.7203**	8	**12**	**0.39**	6.80	1.0	**0.0**
NIn4	0.9869	**0.8290**	**10**	5	**1.22**	10.69	1.0	**0.0**
NIn5	0.9844	**0.9377**	12	**17**	**2.71**	24.43	1.0	**0.0**
NIn6	0.9598	**0.8374**	13	**19**	**2.98**	42.12	1.0	**0.0**
Average	0.9489	**0.8324**	8	**14**	**1.3**	13.4	1.0	**0.0**

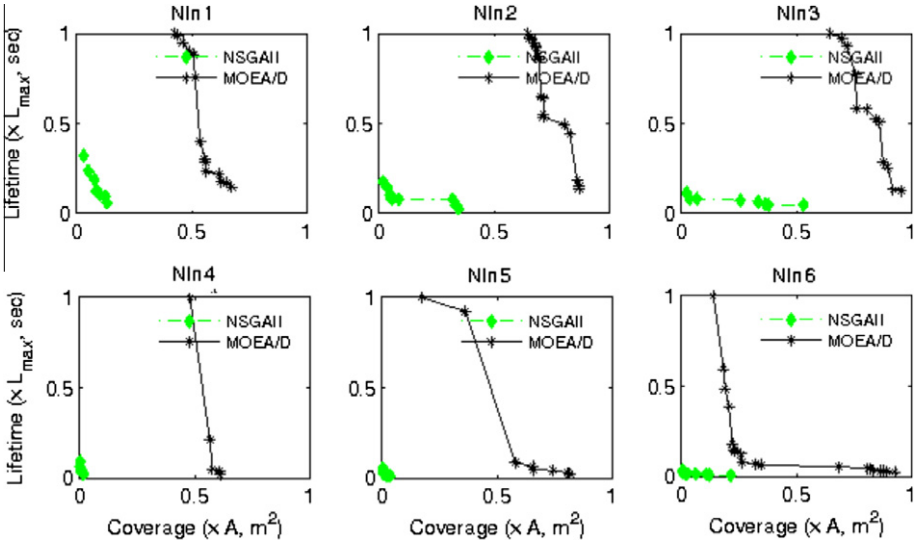


Fig. 12. MOEA/D vs. NSGA-II in NIn1–6.

Please cite this article in press as: A. Konstantinidis, K. Yang, Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.08.011

several DPAP-specific, MOEA/D-based evolutionary operators are proposed and/or adopted. Namely, a DPAP-specific population initialization, specialized genetic operators (*M*-tournament selection, the adaptive crossover and the adaptive mutation operators) and a problem-specific repair heuristic. Simulation results have shown the effectiveness of the proposed operators at improving the performance of MOEA/D and outperforming the constrained NSGA-II in several network test instances. MOEA/D obtains a feasible, diverse set of high quality WSN designs without any prior knowledge on the objective preferences to facilitate the decision maker's choice.

There is a number of avenues for further research. The DPAPs in WSNs include many features (e.g. system models, uncertainty on sensors detection capabilities) and issues (e.g. interference, mobility), which are also important as those in the proposed DPAP. Thus, various multi-objective DPAPs can be defined and tackled by problem-specific MOEA/Ds, similarly to this work. Besides, the hybridization of MOEA/D with problem-specific local improvement techniques for further improving the performance of MOEA/D is also a future study.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE Communications Magazine (2002) 102–114.
[2] X. Liu, P. Mohaparta, On the deployment of wireless data back-haul networks, IEEE Transactions on Wireless Communications 6 (4) (2007) 1426–1435.
[3] A. Konstantinidis, K. Yang, Q. Zhang, D. Zeinalipour-Yazti, A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks, New Network Paradigms, Elsevier Computer Networks 54 (2010) 960–976.
[4] P. Baronti, P. Pillai, V.W.C. Chook, S. Chessa, A. Gotta, 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 30 (2007) 1655–1695.
[5] S. Toumpis, Mother nature knows best: a survey of recent results on wireless networks based on analogies with physics, Computer Networks 52 (2) (2008) 360–383.
[6] P. Santi, Topology control in wireless ad hoc and sensor networks, ACM Computing Surveys 37 (2) (2005) 164–194.
[7] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C. Gill, Integrated coverage and connectivity configuration in wireless sensor networks, in: Proceedings of the First International Conference on Embedded Networked Sensor Systems, 2003, pp. 28–39.
[8] W. Mo, D. Qiao, Z. Wang, Lifetime maximization of sensor networks under connectivity and k-coverage constraints, in: Distributed Computing in Sensor Systems, Lecture Notes in Computer Science, Springer, Berlin/Heidelberg, 2006, pp. 422–442.
[9] M. Hajiaghayi, N. Immorlica, V.S. Mirrokni, Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks, in: MobiCom'03: Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking, ACM, New York, NY, USA, 2003, pp. 300–312.
[10] X.-Y. Li, P.-J. Wan, Y. Wang, C.-W. Yi, Fault tolerant deployment and topology control in wireless networks, in: Proceedings of the Fourth ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), ACM Press, 2003, pp. 117–128.
[11] A. Konstantinidis, K. Yang, Q. Zhang, An evolutionary algorithm to a Multi-Objective Deployment and Power Assignment Problem in wireless sensor networks, in: IEEE Global Communications Conference, GlobeCom08, vol. AH16, 2008, pp. 475–481.
[12] A. Konstantinidis, K. Yang, Q. Zhang, Problem-specific encoding and genetic operation for a Multi-Objective Deployment and Power Assignment Problem in wireless sensor networks, in: International Conference on Communications, ICC09, 2009.
[13] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G. da Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 117–132.
[14] D.B. Jourdan, 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.
[15] R. Rajagopalan, P.K. Varshney, C.K. Mohan, K.G. Mehrotra, Sensor placement for energy efficient target detection in wireless sensor networks: a multi-objective optimization approach, in: Conference on Information Sciences and Systems, Baltimore, Maryland, 2005.
[16] S.C. Oh, C.H. Tan, F.W. Kong, Y.S. Tan, K.H. Ng, G.W. Ng, K. Tai, Multiobjective optimization of sensor network deployment by a genetic algorithm, in: IEEE Congress on Evolutionary Computation 2007 (CEC 2007), 2007, pp. 3917–3921.
[17] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.
[18] J. Jia, J. Chen, G. Chang, Z. Tan, Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm, Computers and Mathematics with Applications 57 (11–12) (2009) 1756–1766.
[19] J. Jia, J. Chen, G. Chang, Y. Wen, J. Song, Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius, Computers and Mathematics with Applications 57 (11–12) (2009) 1767–1775.
[20] Q. Zhang, H. Li, MOEA/D: a multi-objective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731.
[21] A. Konstantinidis, K. Yang, Q. Zhang, F. Gordejuela-Sanchez, Multiobjective K-connected Deployment and Power Assignment in WSNs using constraint handling, in: IEEE Global Communications Conference, GlobeCom09, vol. AHSN-23: Topology Management, 2009.
[22] C.A.C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Computer Methods in Applied Mechanics and Engineering 191 (11–12) (2002) 1245–1287.
[23] A. Konstantinidis, K. Yang, H.-H. Chen, Q. Zhang, Energy aware topology control for wireless sensor networks using memetic algorithms, Elsevier Computer Communications 30 (14–15) (2007) 2753–2764.
[24] A. Raich, T. Liszkai, Multi-objective genetic algorithms for sensor layout optimization in structural damage detection, in: Intelligent Engineering Systems Through Artificial Neural Networks, 2003, pp. 889–894.
[25] N. Srinivas, K. Deb, Multiobjective optimization using nondominated sorting in genetic algorithms, Evolutionary Computation 2 (3) (1994) 221–248.
[26] E. Dunn, G. Olague, Multi-objective sensor planning for efficient and accurate object reconstruction, in: ALLEMAGNE (2004), Lecture Notes in Computer Science, Springer, Berlin, 2004.
[27] K. Kamyoung, M.T. Alan, N. Xiao, A multiobjective evolutionary algorithm for surveillance sensor placement, Environment and Planning B: Planning and Design 35 (2008) 935–948.
[28] G. Molina, E. Alba, E.-G. Talbi, Optimal sensor network layout using multi-objective metaheuristics, Journal of Universal Computer Science 14 (15) (2008) 2549–2565.
[29] W. Ye, J. Heidemann, D. Estrin, An energy-efficient mac protocol for wireless sensor networks, in: INFOCOM, 2002, pp. 1567–1576.
[30] T. Melodia, D. Pompili, 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 23 (3) (2005) 520–532.
[31] J. You, Q. Han, D. Lieckfeldt, J. Salzmann, D. Timmermann, Virtual position based geographic routing for wireless sensor networks, Elsevier Computer Communications 33 (11) (2010) 1255–1265.
[32] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, Wiley and Sons, 2002.
[33] C. Reeves, Genetic algorithms, Handbook of Metaheuristics, Kluwer, 2003.
[34] M. Cardei, M.O. Pervaiz, I. Cardei, Energy-efficient range assignment in heterogeneous wireless sensor networks, in: International Conference on Wireless and Mobile Communications, 2006.
[35] A. Konstantinidis, Q. Zhang, K. Yang, A subproblem-dependent heuristic in MOEA/D for the deployment and power assignment problem in wireless sensor networks, in: IEEE Congress on Evolutionary Computation, CEC09, 2009.
[36] H. Ishibuchi, T. Yoshida, T. Murata, Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 204–223.
[37] N. Weicker, G. Szabo, K. Weicker, P. Widmayer, Evolutionary multiobjective optimization for base station transmitter placement with frequency assignment, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 189–203.
[38] E. Mezura-Montes, C.A.C. Coello, Constrained optimization via multiobjective evolutionary algorithms, in: Multiobjective Problem Solving from Nature, Springer, Berlin, Heidelberg, 2007, pp. 53–75.
[39] S. Toumpis, L. Tassiulas, Optimal deployment of large wireless sensor networks, IEEE Transactions on Information Theory 52 (7) (2006) 2935–2953.
[40] K. Deb, An efficient constraint handling method for genetic algorithms, in: Computer Methods in Applied Mechanics and Engineering, 2000, pp. 311–338.
[41] C.A.C. Coello, A Survey of Constraint Handling Techniques Used with Evolutionary Algorithms, Technical Report, Laboratorio Nacional de Informtica Avanzada, 1999.

Please cite this article in press as: A. Konstantinidis, K. Yang, Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.08.011