Multi-objective Mobile Agent-based Sensor Network Routing using MOEA/D

Andreas Konstantinidis, Christoforos Charalambous, Aimin Zhou and Qingfu Zhang

***Abstract*— Mobile agents are often used in wireless sensor networks for distributed target detection with the goal of minimizing the transmission of non-critical data that negatively affects the performance of the network. A challenge is to find optimal mobile agent routes for minimizing the data path loss and the sensors energy consumption as well as maximizing the data accuracy. Existing approaches deal with the objectives individually, or by optimizing one and constraining the others or by combining them into a single objective. This often results in missing "good" tradeoff solutions. Only few approaches have tackled the Mobile Agent-based Distributed Sensor Network Routing problem as a Multiobjective Optimization Problem (MOP) using conventional Multi-Objective Evolutionary Algorithms (MOEAs). It is well known that the incorporation of problem specific knowledge in MOEAs is a difficult task. In this paper, we propose a problem-specific MOEA based on Decomposition (MOEA/D) for optimizing the three objectives. Experimental studies have shown that the proposed problem-specific approach performs better than two conventional MOEAs in several WSN test instances.**

I. INTRODUCTION

Mobile agents are often used in Wireless Sensor Networks (WSNs) [1] for dealing with distributed-based problems, often providing energy conservation, low latency and minimum non-critical data transmissions [2]. Typical applications of mobile agents include e-commerce, target detection, surveillance and classification [2]. A common scenario in these applications is that a number of targets is uniformly randomly appear in an area of interest. Then a mobile agent is dispatched from a central point, also known as the fusion center, visits a sequence of sensors, fuses important information and returns to the fusion center to classify and/or locate the targets. The goal is usually to find optimal mobile agent routes for minimizing the total path loss and the energy consumption as well as maximizing the data accuracy [2]. Most existing approaches [3] deal with the objectives individually, or by optimizing one and constraining the others, or by combining them into a single objective. This often results in ignoring and loosing "good" solutions, due to the conflicting correlation of the objectives, having a negative impact on the decision making process. Under these circumstances, formulating the mobile agent routing problem as a Multi-objective Optimization Problem (MOP) might be more appropriate.

Andreas Konstantinidis and Christoforos Charalambous are with the Department of Computer Science and Engineering, Frederick University, Nicosia, Cyprus (emails: {com.ca, com.cc}@fit.ac.cy.)

Aimin Zhou is with the Department of Computer Science & Technology, East China Normal University, Shanghai, China (email: amzhou@cs.ecnu.edu.cn)

Qingfu Zhang is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK (email:qzhang@essex.ac.uk)

In Multi-Objective Optimization (MOO) [4], there is no single solution that optimizes all objectives simultaneously, but one is interested in a set of best trade-off candidates, which can be defined in terms of Pareto optimality [4]. Considering a maximization MOP with n objectives, a solution X^* is considered non-dominated or Pareto optimal with respect to another solution Y, if $\forall i \in \{1,...,n\}, X_i \geq Y_i \wedge \exists i \in \{1,...,n\} : X_i > Y_i$, this is denoted as $X \succ Y$. The set of all Pareto optimal or non-dominated solutions in the search space, also called Pareto Set (PS), is often mapped to a Pareto Front (PF) in the objective space [4].

The concept of MOO in WSNs has surprisingly received limited attention, and although there are a number of proposed multi-objective approaches, the applications of these techniques is still relatively scarce. Multi-Objective Evolutionary Algorithms (MOEAs) are promising techniques for dealing with MOPs. However, general MOEAs, usually tackle a MOP as a "black box" [5], i.e. do not use any problem specific knowledge. For example, Jourdan and de Weck [6] have dealt with a layout optimization problem using a conventional Multi-Objective Genetic Algorithm (MOGA) and Jia et al. [7] have tackled a multiobjective optimization scheduling problem using a conventional Non-dominated Sorting Genetic Algorithm II (NSGA-II) [8]. Moreover, Rajagopalan et. al. have originally dealt with the multiobjective mobile agent routing problem using their own Evolutionary Multi-Objective Crowding Algorithm (EMOCA) [9], showing a better performance than the state-of-the-art in MOEAs based on Pareto dominance, the NSGA-II and the Weighted Genetic Algorithm. Using generic operators might be a drawback for MOEAs when dealing with real life problems having undesirable effects, e.g. force the evolutionary process into unnecessary searches, negatively affecting their performance.

The incorporation of problem specific knowledge in MOEAs [10] to direct the search into promising areas of the search space has been proven very successful [11], [12], [13]. However, designing problem specific operators for a MOP as a whole is difficult. The Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [14] overwhelms this difficulty by decomposing the MOP into many scalar subproblems and optimizing them simultaneously, by using neighborhood information and single-objective techniques. The difficulty, however, on adding knowledge on a decompositional MOEA [15] is that the subproblems have different objective preferences, require different treatment and have to be solved simultaneously, in a single run. Therefore, the problem-specific operators should adapt to the requirements and objective preferences of each subproblem dynamically, during the evolution.

In this paper, we propose several problem-specific operators for MOEA/D to deal with the multiobjective mobile agent routing problem. Particularly, the M-tournament selection that combines mating restriction and tournament selection, the window crossover that dynamically controls its parameters based on instant requirements and an adaptive mutation operator that selectively utilizes different mutation strategies based on the subproblems objective preferences. Simulation results show the efficiency and effectiveness of the proposed operators in improving the performance of MOEA/D and demonstrate the superiority of the proposed MOEA/D against EMOCA in several WSN test instances.

II. The Multi-objective Mobile Agent Routing Problem (MMARP)

A. System model and assumptions

Consider a Mobile Agent-based Distributed Sensor Network (MADSN) [2] as a hierarchical 2-D static WSN composed of three tiers. In the lower tier, the sensors are uniformly randomly deployed in a rectangular area of interest A and form K clusters with k sensors per cluster using algorithms such as those in [16]. Each sensor i in a cluster communicates directly or through nearby sensors with its cluster head. The cluster heads, which compose the second tier of the WSN communicate with each other in a strongly connected manner [17] and with the fusion center in the higher tier, which is placed in the middle of A. We assume a perfect medium access control and we adopt the well known Friss free propagation model [2]. In this model, a wireless link is established between sensor i and sensor j if the received signal of j is above a certain threshold due to path loss. The path loss represents the signal attenuation due to free space propagation. The received signal P_{rj} is expressed as follows:

$$P_{rj} = \frac{P_{ti} \times G_{ti} \times G_{rj} \times \lambda^2}{4\pi^2 \times d_{ij}^2 \times \beta}, \tag{1}$$

where P_{ti} and G_{ti} is the transmission power and gain of sensor i respectively, G_{rj} is the gain of the receiving sensor j, λ is the wavelength, d_{ij} is the Euclidean distance between sensors i and j and β is the system loss factor.

Thereinafter, we assume that after the hierarchical WSN is deployed, and the sensors using their sensing capabilities continuously monitoring the area of interest, T stationary targets randomly (uniform distribution) appear in A. Then, a mobile agent is dispatched from the fusion center, visits a sequence of cluster heads and a sequence of sensors within the corresponding clusters, collects data and returns to the fusion center. The goal is to find optimal routes for the mobile agent for making accurate decisions and classification, maximizing the information obtained from multiple sensors [2]. The route is computed as inter cluster and intra-cluster paths using the geographical locations and transceiver characteristics of the sensors, which are assumed to be known [2]. The fusion center computes the inter-cluster path and the cluster heads the intra-cluster paths which consist a non-cyclic sequence of a subset of cluster heads and sensors, respectively.

The energy expended by each sensor i along the path is:

$$E(i) = ((t_{ai} + t_{pi}) \times H_i^2) + (P_{ti} \times t_m), \tag{2}$$

where t_{ai} and t_{pi} indicate the data acquisition and processing times of sensor i respectively, H_i is the operational power level, which is square of the operational frequency, P_{ti} is the transmission power of sensor i and

$$t_m = (M + D)/B$$

is the message transmission time, M is the mobile agent code, D is the size of data in bits and B is the bandwidth.

Finally, each sensor i at location (x_i, y_i) detects a certain amount of energy $e_i(u)$ emitted by a target u at location (x_u, y_u) that is measured as follows:

$$e_i(u) = e_o/(1 + \alpha \times d_{iu}^p), \tag{3}$$

where e_o is the energy emitted by a target, d_{iu} is the Euclidean distance between the target u and sensor i, $2 \leq p \leq 3$ is the signal decay exponent and α is an adjustable constant.

B. Problem formulation

The Mobile Agent Routing Problem in WSNs can be formulated as a MOP,

Given:

- A: 2-D plane.
- K: number of clusters to be formed.
- k: number of sensors per cluster.
- T: targets, uniformly randomly distributed in A.
- sensors transceiver characteristics (P_{ti},G_{ti},G_{rj}, etc.).
- network parameters (B,λ,α,β, etc.).

Decision variables of solution X:

- $((x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l))$ a mobile agent route, where $l \leq (k \times K)$.

Objectives: Minimize energy consumption $E(X)$, minimize path loss $PL(X)$ and maximize detection accuracy $DA(X)$.

The energy consumption of route X is defined as the sum of the energy expended at each sensor along the route [2]:

$$E(X) = \sum_{i=1}^{l} E(i) \tag{4}$$

where $l \leq (k \times K)$ is the number of sensors enroute and $E(i)$ is calculated using Equation (2).

The total path loss is the sum of the path losses associated with each link along the route X and is calculated as follows:

$$PL(X) = \sum_{i=1}^{l-1} PL(i, i+1) \tag{5}$$

where $l \leq (k \times K)$ is the number of sensors enroute and

$$PL(i, i+1) = 10 \times log(P_{ti}/P_{ri+1})$$

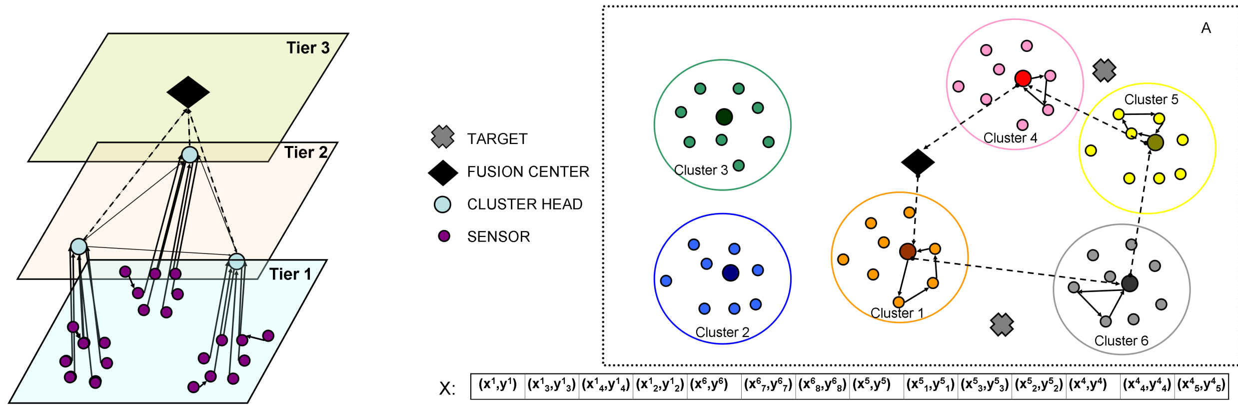

Fig. 1. A hierarchical WSN (left) and an example of a mobile agent route with its encoding representation (right)

is the path loss of link i to $i+1$, P_{ti} is the transmission power of sensor i and P_{ri+1} is calculated using Equation (1).

Finally, the detection accuracy is defined as the sum of the detected signal energy along a route X and is calculated as follows:

$$DA(x) = \sum_{i=1}^{l} e'_i \tag{6}$$

where $l \leq (k \times K)$ is the number of sensors enroute and e'_i is the representative energy of sensor i. The representative energy accounts for faulty sensors and is computed using the randomized median filtering (RMF) [2]. In the RMF, the e'_i is equal to the median of the detected signal energies $e_j(u)$ of a randomly chosen set of neighboring sensors j of the i^{th} sensor, which are calculated using Equation (3).

III. The Proposed Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D)

A. Decomposition and problem representation

The MOP can be decomposed into m subproblems by adopting any technique for aggregating functions [14], e.g. the Tchebycheff approach used here. In this paper, the i^{th} subproblem is in the form

$$minimize \quad g^i(X|w^i, z^*) = min\{w^i_j |f_j(X) - z^*_j|\} \tag{7}$$

where f_j, $j = 1, 2, 3$ are the objectives[1] of the MOP in Subsection II-B, $z^* = (z^*_1, z^*_2, z^*_3)$ is the reference point, i.e. $z^*_j = min\{f_j(P) \in \Omega\}$ for each $j = 1, 2, 3$ and Ω is the decision space. For each Pareto optimal solution X^* there exists a weight vector w such that X^* is the optimal solution of (7) and each solution of (7) is a Pareto optimal solution of the MOP in Subsection II-B. For the remainder of this paper, we consider a uniform spread of the weights w^i_j, which remain fixed for each subproblem i for the whole evolution and $\sum_{j=1}^{3} w^i_j = 1$. Hence, the weight vector w^i is mainly utilized for decomposing a MOP into a set of scalar subproblems by adding different weights to the objectives. In this paper, we have also given a problem-specific meaning to this parameter. Considering the w^i weight vector of a subproblem i, we can predict the objective preference of a particular design and therefore, its position in the objective space. Thereinafter, appropriate scalar strategies can be employed to optimize it accordingly. Note that, this beneficial procedure cannot be utilized by any non-decompositional MOEA framework.

In our approach, a solution X is a sequence of sensor locations to be visited by the mobile agent, composed of a number of cluster head locations (x^i, y^i) and a number of sensor locations (x^i_j, y^i_j) within each corresponding cluster, i.e. $((x^1, y^1), (x^1_1, y^1_1), (x^1_2, y^1_2), (x^2, y^2), (x^2_1, y^2_1))$, where $(x^1_1, y^1_1), (x^1_2, y^1_2)$ are sensor locations traversed in the cluster 1 with the cluster head at location (x^1, y^1). Note that, the size of X is of variable length and it can be from 1 to $(k \times K)$.

[1] a max $f_j(X)$ can be converted to min by inserting a negative sign, i.e. min $-f_j(X)$

B. MOEA/D [14]: an overview

Initially, the Internal Population IP, which stores the best solutions found for each subproblem during the search, is randomly initialized. The genetic operators (i.e. selection, crossover and mutation) are then invoked on IP for offspring reproduction, X^i for each subproblem i, where $i = 1$ to m. Moreover, problem specific heuristics are applied to improve each X^i and obtain Y^i. The update phase of MOEA/D is processed in four steps. (1) Update IP, $IP/\{X^i\}$ and $IP \cup \{Y^i\}$ if $g_i(Y^i|w^i, z^*) < g^i(X^i|w^i, z^*)$, otherwise X^i remains in IP. (2) Update the neighborhood of Y^i, i.e. the solutions of the N closest subproblems of i in terms of their weights $\{w^1, \cdots, w^m\}$ are updated. If $g^j(Y^i|w^j, z^*) < g^j(X^j|w^j, z^*)$, then $IP/\{X^j\}$ and $IP \cup \{Y^i\}$, otherwise X^j remains in IP, where $j \in \{1, ..., N\}$. (3) Update the External Population (EP), which stores all the non-dominated solutions found so far during the search. $EP = EP \cup \{Y^i\}$ if Y^i is not dominated by any solution $X^j \in EP$ and $EP = EP/\{X^j\}$, for all X^j dominated by Y^i. (4) Update reference point z, iff $f_j(Y^i) \leq z_j$ for all $j = 1, 2, 3$ and $f_j(Y^i) < z_j$ for at least one j then set $z_j = f_j(Y^i)$. The search stops after a pre-defined number of generations, gen_{max}.

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

Input:
- network parameters (A, K, k, etc.);
- m : population size and number of subproblems;
- N: neighborhood size;
- uniform spread of weight vectors $w^1, ..., w^m$;
- the maximum number of generations, gen_{max};
- $z = (z_1, z_2, z_3)$ where z_i is the best value found so far for objective f_i

Output: • the external population, EP.

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

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

Step 2: **For** each subproblem $i = 1$ to m **do**

Step 2.1-Genetic Operators: Generate a new solution X^i by using selection, crossover and mutation operators.

Step 2.2-Improvement: Apply a problem specific repair/improvement heuristic on X^i to produce Y^i.

Step 2.3-Update: Update z, IP, EP and the N closest neighbors of subproblem i with Y^i.

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**.

We refer interested readers to [14] for details. In this paper, the focus is on the genetic operators in Step 2.1.

C. The Proposed Problem-specific Operators

In the i-th pass of the loop in Step 2 of the MOEA/D, the genetic operators generate a new solution in Step 2.1.

1) Selection: Initially, we propose a M-tournament selection operator [13] (denoted as M-tourS) that combines the mating restriction considered during selection in [14] and a standard tournament selection [18]. M-tourS, which proceeds as in Algorithm 1, relies on one of the core ideas of MOEA/D to choose promising solutions from the current population, known as parents, to be included for offspring reproduction in the next generation. That is, two neighbor solutions in the weight space (i.e. with respect to the Euclidean distance of their weight vectors $\{w^1, \ldots, w^m\}$) should be similar to each other in the decision space [14]. Stated another way, the optimization of a mobile agent route X^i, should mainly acquire good information (i.e. efficient sensor locations) from a neighbor route X^j; instead of a route X^m which is far away in the weight space (even if X^m is a non-dominated solution) [13]. This is due to the highly non-linear multi-hop nature of WSNs. A tiny change in a route may lead to a big change on the objective values, because of the exponential relationship between the sensors transmission distances and the path loss as well as the energy consumption. Moreover, X^i's neighbors, e.g. X^j and X^k, are competing in i's tournament in terms of $g^i(X|w^i, z^*)$, ignoring their own w^j and w^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. The two selected parent solutions are then forwarded for recombination to the crossover operator.

Algorithm 1 The M-tournament selection operator (M-tourS) for each subproblem i

Input : A population of solutions, IP_{gen};
Output: Two parent solutions, 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 $g^i(X|w^i, z^*)$;
Step 3: Find the best two solutions of the tournament, set them as Pr_1, Pr_2 and stop;

2) Crossover: 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, we propose a window crossover [13], composed of two phases, in which its control parameters change dynamically from subproblem to subproblem based on instant requirements. Initially, the window crossover merges the two parents and deletes the duplicates. Then, it utilizes the weight vector w^i of each subproblem i to predict its objective preference. In phase 1, it uses w^i to choose an appropriate strategy to order the merged solution and facilitate phase 2. In the latter, the window crossover determines two "windows" of size:

$$\begin{aligned} Wc^i &= 1 + (K-1) \times (1 - w_j^i), \\ Ws^i &= 1 + (k-1) \times (1 - w_j^i), \end{aligned} \tag{8}$$

to select promising genetic material from each parent and direct the search into promising areas of the search space for each particular i. The window crossover proceeds as outlined in Algorithm 2 and illustrated in Figure 2.

In Step 0, the two parent solutions are merged to a solution U and the duplicates, i.e. cluster head or sensor locations that appear twice in U, are deleted in Step 1. In Step 2, U is ordered and the windows are determined as follows: each weight coefficient w_j^i of subproblem i corresponds to an objective function and shows the objective preference of each i. Therefore, when w_1^i is the highest, subproblem i favours a near optimal energy consumption $E(X)$ and U is ordered based on the sensors locations in Step 2.1.1. That is, the cluster head locations in U are sorted based on their distance to the fusion center, where 1 is the closest and $l \leq K$ is the farthest location with respect to the fusion center, respectively. In the same way, the sensor locations are ordered for each cluster with respect to the cluster head's location. Then, small windows are calculated in Step 2.1.2 in order to design short

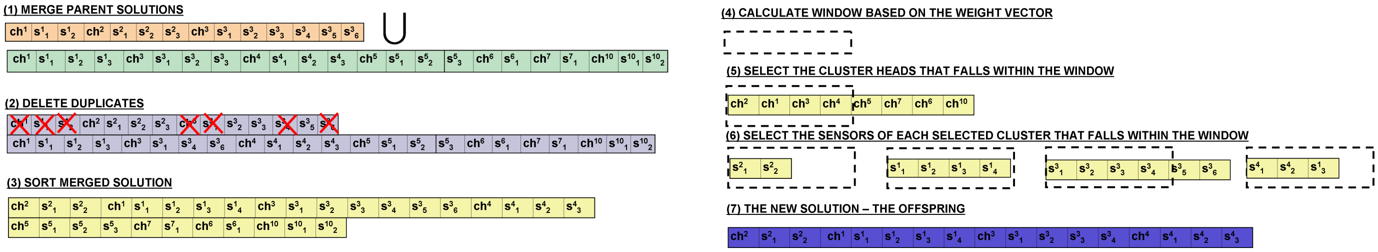

Fig. 2. An illustration of the window crossover operation where ch^i is the cluster head location (x^i, y^i) of cluster i and s^i_j is the sensor location (x^i_j, y^i_j) of sensor j in cluster i

Algorithm 2 Window crossover for a subproblem i

Input: Two solutions Pr_1 and Pr_2;
Output: A solution O;
Step 0: Set $O = \emptyset$; $U = Pr_1 \cup Pr_2$;
Step 1: Delete duplicates in U;
Step 2: Order U and determine Wc^i and Ws^i:
- **Step 2.1: If** $w^i_1 \geq w^i_2$ and $w^i_1 \geq w^i_3$ **then**
  - **Step 2.1.1:** Order U based on the sensors location.
  - **Step 2.1.2:** Calculate Wc^i, Ws^i using w^i_1 in Eq. (8);
- **Step 2.2: Else If** $w^i_2 \geq w^i_1$ and $w^i_2 \geq w^i_3$ **then**
  - **Step 2.2.1:** Order U based on the sensors location.
  - **Step 2.2.2:** Calculate Wc^i, Ws^i using w^i_1 if $w^i_1 > w^i_3$ else w^i_3 in Eq. (8);
- **Step 2.3: Else If** $w^i_3 \geq w^i_1$ and $w^i_3 \geq w^i_2$ **then**
  - **Step 2.3.1:** Order U based on the sensors signal energy calculated using Equation (3) .
  - **Step 2.3.2:** Calculate Wc^i, Ws^i using w^i_1 if $w^i_1 < w^i_2$ else w^i_2 in Eq. (8);

Step 4: Select Wc^i cluster head locations from U and add them in O;
Step 5: For each cluster head location that exists in both O and U, select Ws^i sensor locations from U and add them in O;
Step 6: Stop and output O;

mobile agent routes that favour a low $E(X)$. When w^i_2 is the highest, i favours a near optimal path loss $PL(X)$. In that case, U is ordered based on the sensors locations in Step 2.2.1 as in Step 2.1.1. Thereinafter, a low path loss can be obtained with either short mobile agent routes or long routes composed of multiple short hops. Therefore, average size windows might be the best choice which results in average size mobile agent routes. When w^i_3 is the highest, subproblem i favours a near optimal detection accuracy $DA(X)$. In that case, U is ordered based on the sensors signal energy in Step 2.3.1, where 1 is the sensor with the highest signal energy and $l \leq (k \times K)$ is the one with the lowest. Then, large windows are determined in Step 2.3.2 to design long mobile agent routes that favour a higher data accuracy. Finally, in Steps 5 and 6 the windows sizes determine the number of cluster head locations and sensor locations of each cluster to be included in the new solution O.

3) Mutation: The last operator in Step 2.1 of MOEA/D (Subsection III-B) is the mutation operator, which is responsible for maintaining the diversity of the population by modifying the elements of a solution O with a r_m probability.

Algorithm 3 Adaptive mutation for each subproblem i

Input: A solution O.
Output: A mutated solution Y.
Step 0: Set r_m;
For each element j of solution O **do**
Step 1: Generate a uniform random number $rand \in [0, 1]$;
Step 2: If $rand \leq r_m$ **then**
- **Step 2.1:If** $w^i_1 \geq w^i_2$ and $w^i_1 \geq w^i_3$ **then**
  Delete element j from O, if j is a cluster head location delete all sensor locations of that cluster from O;
- **Step 2.2:Else if** $w^i_2 \geq w^i_1$ and $w^i_2 \geq w^i_3$ **then**
  Uniformly randomly choose two cluster head locations or two sensor locations within the same cluster and swap them in O;
- **Step 2.3:Else if** $w^i_3 \geq w^i_1$ and $w^i_3 \geq w^i_2$ **then**
  Uniformly randomly choose a new cluster head location or a new sensor location in A and add it in O;

End For
Step 3: Output $Y = O$;

Our mutation operator, named adaptive mutation, treats each subproblem i based on its w^i. If w^i_1 is the highest then, a sensor location is deleted to favour short mobile agent routes and therefore a lower $E(X^i)$. If w^i_2 is the highest then, two sensor locations are swapped to modify the links of the route and maybe obtain lower $PL(X^i)$. Finally, if the w^i_3 is the highest then, a new sensor location is added in order to favour long routes and therefore a higher $DA(X^i)$. The mutation operator proceeds as in Algorithm 3.

TABLE I
TEST INSTANCES

Test Inst.	A (m^2)	no. of clusters	no. of sensors per cluster	no. of targets
T1	10000	10	10	5
T2	10000	10	20	3
T3	10000	20	10	3
T4	10000	20	20	5
T5	40000	10	10	5
T6	40000	10	20	3
T7	40000	20	10	3
T8	40000	20	20	5

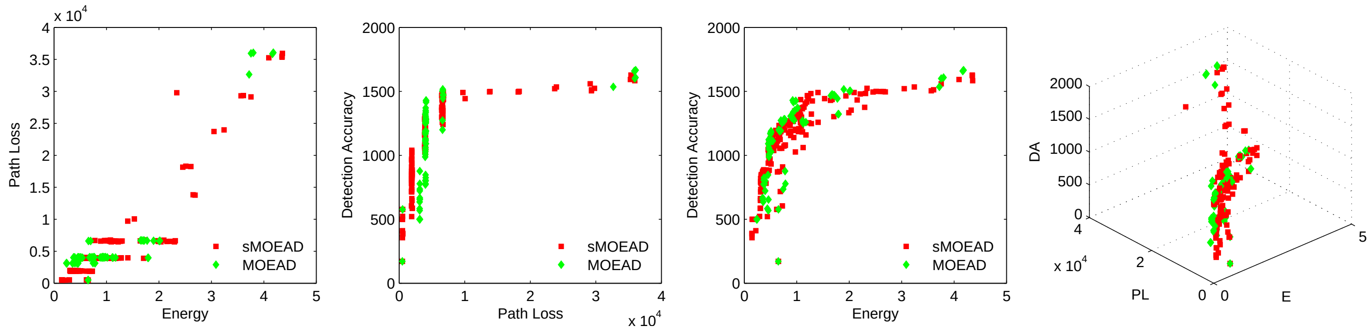

Fig. 3. Conventional MOEAD versus specialized MOEA/D (sMOEAD), i.e. with the proposed operators M-tourS, window crossover and adaptive mutation in test instance 8

TABLE II

CONVENTIONAL MOEA/D (M) VERSUS SPECIALIZED MOEA/D (SM), I.E. WITH THE PROPOSED OPERATORS M-TOURS, WINDOW CROSSOVER AND ADAPTIVE MUTATION IN EIGHT WSN TEST INSTANCES. THE MEAN AND THE STANDARD DEVIATION (SD) IS PROVIDED FOR EACH METRIC. THE BEST PERFORMANCE IN EACH CASE IS GIVEN IN BOLD.

Test Inst.	C(M,sM)	C(sM,M)	Dom (M,sM)	Dom (sM,M)	S(M)	S(sM)	NDS (M)	NDS (sM)	CPU(M)	CPU(sM)
T1	**0.04**	0.22	**0.97**	0.02	2.4 $\times 10^3$	**3.8** $\times 10^5$	49	**75**	0.47	**0.27**
T2	0.17	**0.16**	0.43	**0.56**	**9.8**$\times 10^4$	5.6 $\times 10^4$	**35**	28	0.66	**0.65**
T3	0.46	**0.28**	0.18	**0.81**	2.3$\times 10^6$	4.9$\times 10^6$	**77**	67	0.73	**0.3**
T4	0.33	**0.24**	0.25	**0.75**	2$\times 10^8$	**2.4**$\times 10^8$	119	**121**	**1.13**	1.46
T5	0.47	**0.35**	0.4	**0.6**	1.8$\times 10^5$	**6.7**$\times 10^5$	**42**	37	0.5	**0.43**
T6	**0.13**	0.37	**0.91**	0.8	1.5$\times 10^6$	**2.7**$\times 10^6$	46	**67**	**0.52**	0.71
T7	0.47	**0.36**	0.49	**0.5**	**3.49**$\times 10^5$	2$\times 10^5$	**66**	41	0.64	**0.45**
T8	0.51	**0.22**	0.21	**0.78**	1.62$\times 10^6$	**1.7**$\times 10^6$	82	**203**	**0.98**	1.38
Mean	0.32	**0.27**	0.48	**0.6**	2.5 $\times 10^7$	**3.1** $\times 10^7$	64.5	**79.8**	0.72	**0.7**
SD	0.18	**0.07**	0.3	**0.26**	**7** $\times 10^7$	8.4 $\times 10^7$	**27.7**	57.6	**0.23**	0.46

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The goals of our experimental studies are: (1) to demonstrate the effectiveness of our problem specific operators (i.e. the M-tournament selection, the window crossover and the adaptive mutation) for improving the performance of the conventional MOEA/D (i.e. with a random selection, two-point crossover and random mutation as in [14]). (2) To test the strength of the improved MOEA/D against the EMOCA, in various network test instances giving the trade offs of our objectives and a variety of mobile agent route choices.

We have chosen EMOCA to compare the performance of our approach because it was originally proposed for tackling the mobile agent routing problem and it has shown a better performance than the state of the art in MOEAs based on Pareto dominance, the NSGA-II, as well as a Weighted Genetic Algorithm (WGA) [2], [19]. The key characteristic of EMOCA, which is similar to NSGA-II, is that it addresses the convergence and diversity issues by using non-domination rank and diversity rank (i.e. crowding distance estimation) in multiple stages of the algorithm (i.e. during selection, update etc.)

In our experimental studies, we have used the following network parameter settings as in [2]:data size $400B$, transmitter and receiver gain $2dB$, channel operation frequency $20kHz$, channel bandwidth $320kbps$, transmitter power range $200-1000mw$, operational power range $100-500mw$, data acquisition and processing time $50-100ms$ and $\alpha=\beta=1$. Moreover, after exhaustive simulation tests we have set the algorithm parameters as follows: population size $m=500$, crossover rate $r_c=0.9$, mutation rate $r_m=0.1$, tournament size $s_t=10$ and $gen_{max}=250$, $N=2$ as in [12]. In all experiments, we have used the same resources and function evaluations for fairness. Table I shows various test instances that were specified using the well-known fractional factorial design [20].

In the absence of any prior knowledge on the real Pareto Front, we have used the following performance metrics for comparing set of solutions and therefore evaluating the performance of the MOEAs as in [2], [12]: The pairwise quality metrics $C(A,B)$ and $Dom(A,B)$ show how many non-dominated solutions in set A are dominated and how many they dominate with respect of those in set B, respectively. In that cases, a low C and a high Dom indicate a better quality of A. Besides, we have used the hypervolume metric $S(A)$ to evaluate the diversity of set A, for which the higher the better. A high number of Non-Dominated Solutions ($NDS(A)$, i.e. the cardinality of set A) and therefore more mobile agent route choices within an acceptable CPU time is also desired.

Figure 3 shows the non-dominated solutions obtained by the conventional MOEA/D and the proposed specialized MOEA/D (sMOEA/D) in WSN test instance T8 of Table I. The figure shows the performance of the MOEAs in combinations of

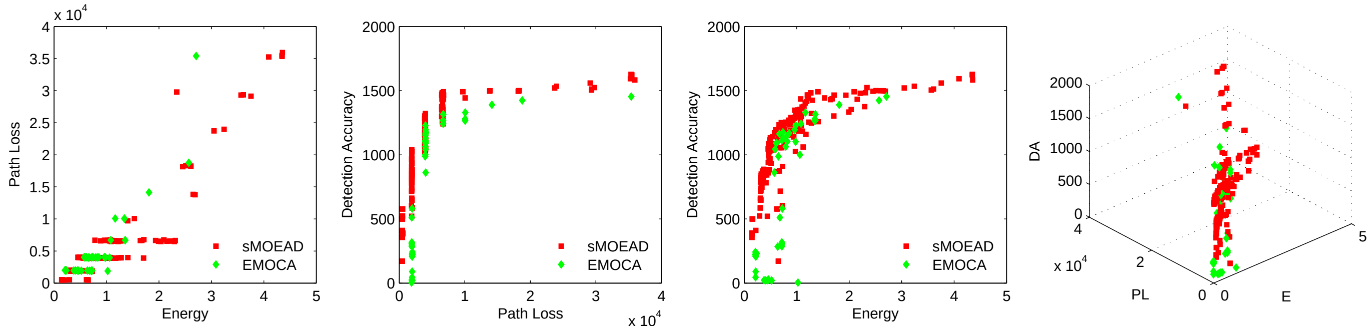

Fig. 4. EMOCA versus specialized MOEA/D (sMOEAD), i.e. with the proposed operators M-tourS, window crossover and adaptive mutation in test instance 8

TABLE III

EMOCA (E) VERSUS SPECIALIZED MOEA/D (SM), I.E. WITH THE PROPOSED OPERATORS M-TOURS, WINDOW CROSSOVER AND ADAPTIVE MUTATION IN EIGHT WSN TEST INSTANCES. THE MEAN AND THE STANDARD DEVIATION (SD) IS PROVIDED FOR EACH METRIC. THE BEST PERFORMANCE IN EACH CASE IS GIVEN IN BOLD.

Test Inst.	C(E,sM)	C(sM,E)	Dom(E,sM)	Dom (sM,E)	S(E)	S(sM)	NDS (E)	NDS (sM)	CPU(E)	CPU(sM)
T1	0.7	**0**	0	**1**	1.5×10^6	$\mathbf{3.8\times10^6}$	47	**75**	0.52	**0.27**
T2	0.54	**0**	0	**1**	$\mathbf{6.5\times10^5}$	5.6×10^4	**93**	28	0.95	**0.65**
T3	**0.36**	0.4	**0.64**	0.35	$\mathbf{3.7\times10^7}$	4.9×10^6	66	**67**	0.68	**0.3**
T4	0.95	**0**	0	**1**	$\mathbf{8.7\times10^8}$	2.4×10^8	70	**121**	2.0	**1.46**
T5	0.75	**0.18**	0.53	**0.94**	5.4×10^6	$\mathbf{6.7\times10^6}$	**64**	32	0.71	**0.43**
T6	0.5	0.5	**0.61**	0.38	$\mathbf{1.6\times10^7}$	2.7×10^6	58	**67**	0.74	**0.71**
T7	0.96	**0.05**	0.36	**0.99**	$\mathbf{2.3\times10^7}$	2×10^5	**78**	41	0.81	**0.45**
T8	1	**0**	0	**1**	9.2×10^7	$\mathbf{1.7\times10^8}$	47	**203**	2.43	**1.38**
Mean	0.72	**0.14**	0.26	**0.83**	$\mathbf{2.3\times10^8}$	3.1×10^7	65.3	**79.8**	1.1	**0.7**
SD	0.23	**0.2**	0.29	**0.28**	4×10^8	$\mathbf{8.4\times10^7}$	**15.4**	57.6	0.7	**0.46**

two of the three objectives as well as all together in a 3D view. From the results we have observed that the proposed operators have increased the performance of MOEA/D along the direction of all the three objectives, giving solutions of higher quality. Besides, sMOEA/D has obtained a more diverse set of non-dominated solutions showing a more efficient exploration of the objective space. Table II supports the observations just mentioned showing a comparison between MOEA/D and sMOEA/D in eight WSN test instances. The conclusions drawn from the results are the following, sMOEA/D obtains a higher quality of solution in six out of eight test instances. Besides, sMOEA/D obtains a more diverse PF with a higher number of NDS within a lowest CPU time in most cases.

Thereinafter, Figure 4 shows the superiority of sMOEA/D compared to EMOCA in WSN test instance T8. This figure shows the non-dominated solutions obtained by the two MOEAs in combinations of two of the three objectives as well as all together in a 3D view. sMOEA/D has obtained a higher quality and a more diverse set of non-dominated solutions. More conclusion can be observed from Table III that shows a comparison between sMOEA/D and EMOCA in eight WSN test instances (i.e. Table I). The results show that sMOEA/D performs better than EMOCA in six out of eight test instances in terms of quality and obtains a higher number of solutions within a lower CPU effort. The diversity of the non-dominated solutions obtained by the two MOEAs is similar.

Finally, Figures 5 and 6 show the mobile agent routes obtained by MOEA/D and EMOCA with the highest Data Accuracy, respectively. The mobile agent route is illustrated as a thin line, the sensors are marked with different shapes and colors per cluster, each having a cluster head in the same shape and colour but in bigger size. Moreover, the targets (five) are marked as solid black diamonds. In Figure 5, the mobile agent of MOEA/D selectively visits effective cluster heads, and only the necessary sensors within each cluster, in the target neighborhood, giving a high data accuracy 1.28×10^4, a low path loss 3.9×10^3 and a low energy consumption 0.94. In contrast, Figure 6 shows that the route obtained by EMOCA consists of unnecessary visits to clusters that are far away from the targets, resulting in poor data accuracy 1.45×10^3, high path loss 3.5×10^4 and more energy consumption 2.7.

V. CONCLUSIONS

In this paper, a M-tournament selection, a window crossover and a mutation operators are proposed for specializing the MOEA/D to deal with the multi-objective mobile agent routing problem in WSNs. Initially, the MOP, which aims at finding optimal mobile agent routes for minimizing the sensors energy consumption, minimizing the path loss and maximizing the data accuracy, is decomposed into a number of subproblems. Then, by using problem-specific knowledge the proposed operators are designed, for which the weight coeffi-

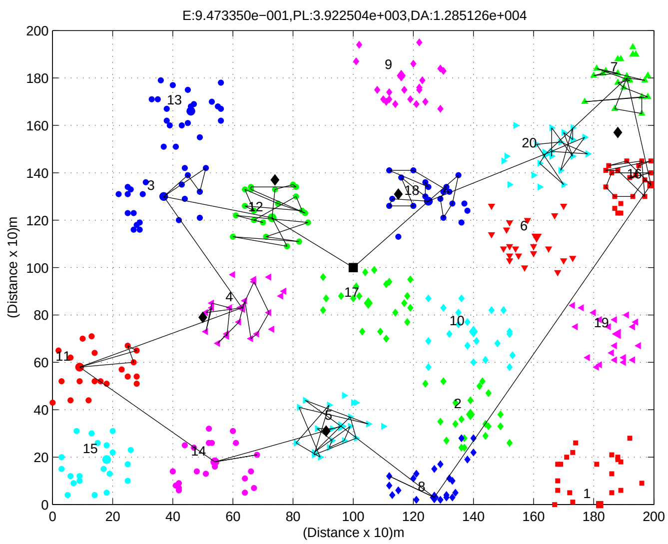

Fig. 5. The mobile agent route obtained by sMOEA/D that provides the highest data accuracy for test instance T8.

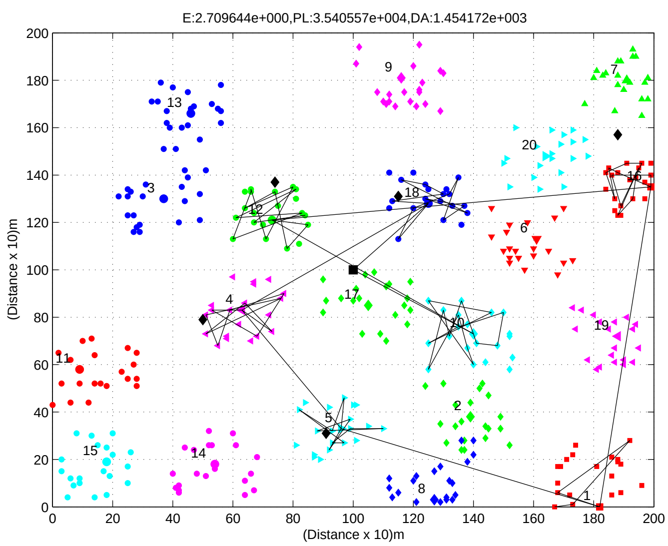

Fig. 6. The mobile agent route obtained by EMOCA that provides the highest data accuracy for test instance T8.

cients of each subproblem play an important role (i.e. shows each subproblems objective preferences.) Finally, simulation results have shown that the specialized MOEA/D performs better than the conventional MOEA/D and the EMOCA in several WSN test instances, in terms of quality, diversity, number of non-dominated solutions and CPU effort.

Problem-specific local improvement and repair strategies are currently under investigation for further improving the performance of MOEA/D and dealing with a constrained MOP in Mobile WSNs. Moreover, the mobile agent routing problem in WSNs include many features (e.g. multiple agents, moving targets) and issues (e.g. latency), which are also important as those adopted in this work. Thus, various multiobjective mobile agent routing problems can be defined and tackled by problem-specific MOEA/Ds in a similar way.

References

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

[2] R. Rajagopalan, C. K. Mohan, K. G. Mehrotra, and P. K. Varshney, *Multi-objective evolutionary algorithms for wireless sensor network design*. Idea Group Inc., 2006, ch. Multi-objective Optimization in Computational Intelligence: Theory and Practice.

[3] Q. Wu, N. S. V. Rao, S. S. Iyengar, V. K. Vaishanavi, H. Qi, and K. Chakrabarty, "On computing mobile agent routes for data fusion in distributed sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16(6), pp. 740–753, 2004.

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

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

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

[7] J. Jia, J. Chen, G. Chang, and Z. Tan, "Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm," *Computers and Mathematics with Applications*, vol. 57, no. 11–12, pp. 1756–1766, 2009.

[8] 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.

[9] R. Rajagopalan, C. K. Mohan, K. G. Mehrotra, and P. K. Varshney, "Emoca: An evolutionary multi-objective crowding algorithm," *Journal of Intelligent Systems*, 2006.

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

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

[12] A. Konstantinidis, Q. Zhang, and 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.

[13] A. Konstantinidis, K. Yang, Q. Zhang, and 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*, 2009.

[14] 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.

[15] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 204–223, 2003.

[16] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[17] 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.

[18] C. Reeves, *Handbook of Metaheuristics*. Kluwer, 2003, ch. Genetic algorithms, pp. 65–82.

[19] R. Rajagopalan, C. K. Mohan, P. K. Varshney, and K. Mehrotra, "Multi-objective mobile agent routing in wireless sensor networks," in *Proc. IEEE CEC'05*, Edinburgh, Scotland, September 2005.

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