2005

Multi-objective mobile agent routing in wireless sensor networks

Ramesh Rajagopalan
Syracuse University, rarajago@syr.edu

Chilukuri K. Mohan
Syracuse University, ckmohan@syr.edu

Pramod Varshney
Syracuse University, varshney@syr.edu

Kishan Mehrotra
Syracuse University, mehrotra@syr.edu

Multi-objective Mobile Agent Routing in Wireless Sensor Networks

Ramesh Rajagopalan, Chilukuri K. Mohan, Pramod Varshney, Kishan Mehrotra

Electrical Engineering and Computer Science Dept., Syracuse University, NY 13244-4100, USA

{rarajago, ckmohan, varshney, mehrotra}@syr.edu

Abstract- **A recent approach for data fusion in wireless sensor networks involves the use of mobile agents that selectively visit the sensors and incrementally fuse the data, thereby eliminating the unnecessary transmission of irrelevant or non-critical data. The order of sensors visited along the route determines the quality of the fused data and the communication cost. We model the mobile agent routing problem as a multi-objective optimization problem, maximizing the total detected signal energy while minimizing the energy consumption and path loss. Simulation results show that this problem can be solved successfully using evolutionary multi-objective algorithms such as *EMOCA* and *NSGA-II*. This approach also enables choosing between two alternative routing algorithms, to determine which one results in higher detection accuracy.**

1. Introduction

Due to their flexibility and cost effectiveness, wireless sensor networks (WSNs) have been used for numerous applications including environmental monitoring, facility monitoring, and military surveillance for tasks such as target detection. In distributed detection problems, the sensors transmit data to the fusion center. However, the transmission of non-critical data involves use of excessive battery power and network bandwidth. To circumvent this problem, Qi et al. [Qi01] have proposed the concept of *Mobile Agent based Distributed Sensor Networks* (*MADSN*s) where the mobile agent (program dispatched from a source node and executed at remote nodes) selectively visits the sensors and incrementally fuses the data. It has been found that mobile agent implementation saves almost 90 percent of data transfer time since it avoids raw data transfers [Qi01].

Algorithms based on *local closest first* (LCF) and *global closest first* (GCF) heuristics [Qi01] have been used to compute mobile agent routes for distributed data integration. The performance of these algorithms deteriorates as the network size grows and the sensor distributions become more complicated. These approaches consider only spatial distances between sensor nodes for route computation. However, other important factors must also be considered when computing a route for mobile agent.

The computation of mobile agent routes involves tradeoffs between energy consumption, path loss and detection accuracy. For instance, as the number of sensors in the route increases, the quality of fused data improves but the energy consumption and path loss increase. We investigate algorithms to compute routes for a mobile agent with high detection accuracy, low path loss and low energy consumption. The tradeoffs are addressed using a multi-objective optimization *(MOO)* framework employing evolutionary algorithms. Research in such algorithms has gathered significant attention in the recent past, with detailed surveys in [Coello00, Coello02].

For the agent routing task, a preliminary attempt considering multiple objectives has been pursued by Wu *et al.* [Wu04], who have combined three objectives (communication cost, path loss and detected signal energy level) into a single function and optimized it using a genetic algorithm that outperforms the LCF and GCF strategies. To evaluate the effectiveness of MOO algorithms against a single-objective approach, we have implemented a *weighted genetic algorithm* (WGA), iterated with different weights in order to obtain different non-dominated solutions. WGA is a generalization of the approach pursued by Wu *et al.* [Wu04]. However, this is not a true MOO approach, and cannot find optimal solutions of interest that reflect the tradeoffs if the Pareto-optimal region is non-convex. NSGA-II is a widely known and applied MOO algorithm, motivating its use for this problem; in addition, we also use EMOCA due to its superior performance in our recent studies on other problems and the benchmark problems discussed in [Deb00].

In Section 2, we explain the mobile agent routing problem. Section 3 elaborates the various objectives to be optimized. Section 4 summarizes the evolutionary MOO algorithms used in our simulations. Simulation results and conclusions are presented in Sections 5 and 6 respectively.

2. Mobile Agent Routing Problem

We consider the task of routing mobile agents in a hierarchical MADSN, shown in Figure 1. Sensor nodes within the communication range of each other form a cluster, using algorithms such as those in [Chan04]. The sensors within each cluster form a completely connected graph. Each sensor in a cluster communicates with its *cluster head,* which is a sensor node with special features such as additional processing power and battery life compared to other sensors. The cluster heads form a completely connected graph: they can communicate with each other and with the fusion center. Elements of the network are connected through wireless communication links.

Sensor nodes are randomly distributed in each cluster and collect measurements of different modalities (such as acoustic, seismic and infrared) from the environment. A mobile agent sequentially migrates among the sensors and the cluster heads through the network, integrates raw data

with a desired resolution, and carries the final result back to the fusion center. Each sensor processes the received signal from the target and transmits the signal strength level (not the raw data) to the cluster head.

The cluster heads and the fusion center have predetermined knowledge necessary for computing the route, such as the geographical locations (through GPS interfaces) and transmitting/receiving parameters of the sensor nodes. The fusion center computes an inter-cluster path consisting of a non-cyclic sequence of cluster heads. Each cluster head in the inter-cluster path also computes a path consisting of a non-cyclic sequence of sensor nodes within its cluster. The mobile agent (dispatched from the fusion center) would visit the sequence of cluster heads and a sequence of sensors within the corresponding clusters, collect data, and then return to the fusion center.

3. Objectives to be optimized

Our multi-objective optimization algorithms must: (a) minimize energy consumption, (b) minimize path loss, and (c) maximize total detected signal energy. These objectives [Wu04] are discussed below in greater detail.

3.1. Energy Consumption

Sensors are equipped with limited battery power and the total energy consumption of the WSN is a very critical consideration. Each sensor consumes some energy in data acquisition, processing and transmission. We consider a heterogeneous WSN, in which some sensors might have more power and data processing capability compared to other sensors. Hence the energy consumption of the WSN depends on the capacity of a sensor and its functionality. The messages transmitted between sensors include the mobile agent code of size M bits and the data of size K bits. The partially integrated data at each sensor is stored in a fixed data size of K bits. The message transmission time over a wireless channel of bandwidth B is given by

$$t_m = (M+K)/B.$$

The energy consumption of a path P is the sum of the energy expended at each sensor node along the path. If $(n_0, n_1, n_2..., n_l)$ denotes the sequence of nodes along a path P, then the total energy consumption $E(P)$ is given by

$$E(P) = \sum_{k=0}^{l} ((t_{ak} + t_{pk}) \times H_k^{2}) + (P_{tk} \times t_m)$$

where t_{ak} and t_{pk} indicate the data acquisition time and data processing time for node k [Wu 2004], H_k and P_{tk} denote the operational power level and transmission power of node k. The operational level H_k corresponds to the operational frequency of the sensor k, the square of which determines its operating power level.

3.2. Path Loss

Wireless communication links need to be established between neighboring sensor nodes as the mobile agent traverses a route. The received signal level may not be acceptable if it is below a certain threshold due to path loss. The path loss represents the signal attenuation due to free space propagation, and should be minimized to guarantee reliable communication.

The total path loss along a path is the sum of the path losses associated with each link along the path. The path loss associated with a single link is the ratio between the power P_{ti} transmitted by sensor i and the power P_{rj} received by sensor j, computed (in dB) as:

$$PL(d_{i,j}) = 10 \times \log(P_{ti}/P_{rj})$$

where d_{ij} is the Euclidean distance between the co-ordinates of sensors i and j. The path loss is computed using the well-known Friis free space propagation model [Friss46], which defines the relation between the power P_{rj} received by a sensor and the power P_{ti} transmitted by a sensor:

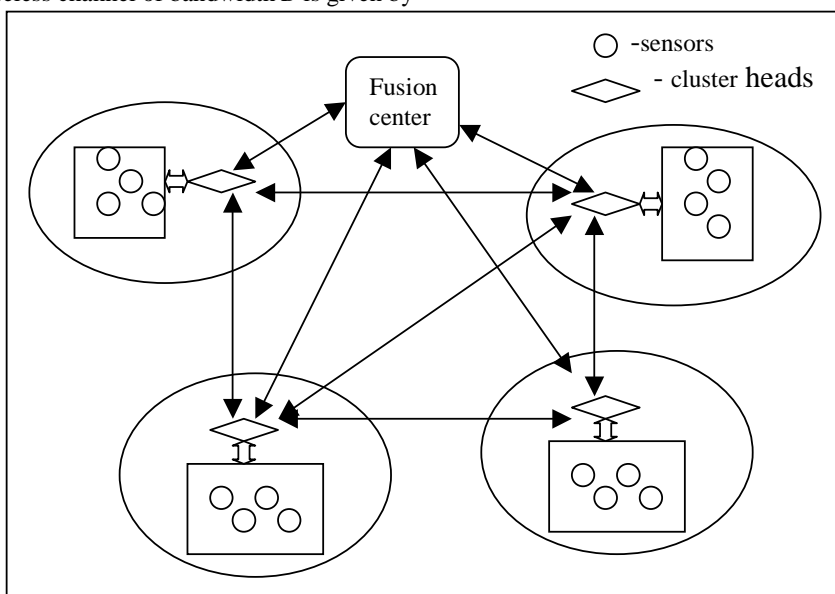

Figure 1: Hierarchical MADSN architecture: the arrows indicate the wireless communication links

$$P_{ri} = \frac{P_{ti} \times G_{ti} \times G_{rj} \times \lambda^2}{(4\Pi^2) \times d_{ij}^2 \times \beta}$$

where G_{ti} is the gain of transmitting sensor i, G_{rj} is the gain of the receiving sensor j, λ is the wavelength, and β is the system loss factor. The total path loss of a path P is calculated as

$$PL(P) = \sum_{k \in E} PL_k$$

where PL_k is the path loss associated with an edge k.

3.3. Detection Accuracy

High detection accuracy is also an important goal for accurate inference about the target. Each sensor detects a certain amount of energy $e_k(u)$, emitted by a target. If K_o is the energy emitted by a target at location $u = (x_t, y_t)$, the signal energy e_k measured by a sensor i is $e_k(u) = K_o/(1 + \alpha\, d_i^{\ p})$, where d_i is the Euclidean distance between the target location and sensor location, p is the signal decay exponent that takes values between 2 and 3, and α is an adjustable constant[1].

The goal of the mobile agent is to accumulate maximum information from each sensor for accurate decisions in target detection and classification. A path P is a non-cyclic sequence of sensor nodes within a set of selected clusters of the hierarchical MADSN. The fusion center decides the sequence of clusters the mobile agent should visit based on the representative energy of the cluster head. The sum of the detected signal energy along a path P is defined as

$$DE(P) = \sum_{i=1}^{n} E_i$$

where E_i is the representative energy of the ith sensor as described below in 3.3.1 and 3.3.2.

Mobile agent routing in a WSN should be robust, allowing for faulty sensors. To the best of our knowledge, mobile agent routing has not been studied in the context of fault tolerance in WSNs. Byzantine faulty sensors [Lamport 82] send incorrect data that tends to be on either extreme in comparison with data sent by non-faulty sensors. In the model we consider, each sensor detects a certain amount of energy from the target and transmits the detected signal energy level to the cluster head. The cluster head computes two representative energy values, viz.,
1. Cluster head representative energy, and
2. Representative energy for each sensor in its cluster.

We propose and evaluate two different approaches to calculate these representative energies: randomized median filtering and randomized censored averaging.

3.3.1 Randomized median filtering: In this approach, the representative energy of each cluster head is computed as the median of all the detected energy values sent by the sensors of that cluster. In a cluster with s sensor nodes,

each sensor has s-1 neighbors. For each sensor, the median of $e_k(u)$'s of m randomly chosen neighbors is used, to calculate the representative energy E_i of sensor i. Here, m is an algorithm parameter randomly chosen in the interval $[3, s-1]$.

3.3.2 Randomized censored averaging: This approach improves the detection accuracy of the system by eliminating extreme values on both ends of the received signal energy values.
1. The cluster head drops the r highest and r lowest values among the set of detected energy values sent by all s sensors to the cluster head. To accommodate for Byzantine faulty behavior [Lamport82, Clouqueur01], r is chosen to be $r = (s-1)/3$. The average of the remaining s-$2r$ values is computed as the representative energy of the cluster head.
2. For each sensor i, the cluster head randomly chooses m sensors in the interval $[2r+2, s-1]$ among its s-1 neighbors. The cluster head drops r highest and r lowest values among the set of m of these $e_k(u)$'s and averages the remaining m-$2r$ values to compute the representative energy E_i of sensor i.

4. Multi-objective evolutionary algorithms

We have solved the mobile agent routing problem using two evolutionary MOO algorithms that strive towards finding multiple diverse high quality candidate solutions: *EMOCA* [Rajagopalan04a, Rajagopalan04b]*, and *NSGA-II* [Deb00]. Figure 2 shows the common high-level description of both algorithms, as applied to this problem.

Randomly generate an initial population;
While computational bounds are not exceeded, do:
*   Generate mating population;
*   Generate offspring by two-point crossover followed by mutation;
*   Trim the new pool consisting of parents and offspring to generate the population for the next iteration, with the primary criterion of non-domination and secondary criterion of diversity.

Figure 2: Multi-objective evolutionary algorithm

The main differences between the two algorithms are the following:
1. **Mating selection:** EMOCA employs binary tournament selection to generate the mating population where the fitness of each individual equal to the sum of its non-domination rank and diversity rank (higher crowding distance corresponds to a better or lower diversity rank), and some dominated individuals would be selected for mating if they contribute to population diversity. NSGA-II primarily uses non-domination rank for selection, with diversity being used to break ties.
2. **Archiving:** EMOCA is akin to *PAES* [Knowles 00] in that it maintains an explicit archive separate from the

---

[1] For our simulations, we chose $\alpha = 1$.

evolving population, whereas NSGA-II retains good solutions within the evolving population.

3. **New pool generation:** In EMOCA, each offspring is compared with one of the parents to form the new pool, considering both domination and crowding density. There are three possible cases:

   Case 1: If the offspring dominates the parent, then the offspring is added to the new pool.

   Case 2: If dominated by the parent, and if it has a higher crowding distance than the parent, then the offspring is added to the new pool with probability *1-exp(Ψ(parent)- Ψ(offspring)),* where Ψ denotes the "crowding distance" of a solution as defined in NSGA-II. This policy rewards diversity.

   Case 3: Otherwise, if the offspring has a higher crowding distance than the parent, then it is added to the new pool, else the parent is added to the new pool.

Each individual in the population is a sequence of sensor nodes being visited by the mobile agent, represented as a sequence of cluster-head and labels of sensors within clusters, e.g., (*ch1, s7, s6, s3, ch2, s8, s1, s9, s4, ch3, s5, s11, s12)* denotes that *s7, s6,* and *s3* are sensors traversed in the cluster with cluster-head *ch1.* For each individual, path computation proceeds as follows:

Ø The fusion center computes an inter-cluster path between the cluster heads. The initial path consists of a random non-cyclic sequence of cluster heads.

Ø Each cluster head in the inter-cluster path computes an intra-cluster path consisting of a random non-cyclic sequence of sensor nodes within its cluster.

Ø The mobile agent is dispatched from the fusion center and visits the first cluster head in the path, followed by a sequence of sensor nodes within that cluster, and returning to the cluster head. Then it successively visits the remaining cluster heads and sensor nodes in the routing sequence and returns to the fusion center.

Operators: Two-point crossover is applied separately for intra-cluster and inter-cluster paths, removing duplicate occurrences of sensors. For example, crossover between

Parent 1 : *1-2-5* | *13 -7-19-8* | *14-6-12 –0* and
Parent 2: *3-5-6-9-1* | *7-4-2* |*16-19-18*

where " | " denotes the crossover points, results in

Child 1: *1-2-5 -7-4-2-14-6-12-0* and
Child 2: *3-5-6-9-1-13-7-19-8-16-19-18,*

subsequently corrected to:

Child 1: *1-2-5 -7-4-14-6-12-0*
Child 2: *3-5-6-9-1-13-7-8-16-19-18.*

The mutation operator's application results in swapping the positions of two randomly chosen sensors in the path. For instance, 1-3-**7**-9-8-6- **2**-5-0 may be mutated to 1-3-**2**-9-8-6- **7**-5-0. The mutation operator is also applied independently for the intra-cluster and inter-cluster paths. During each iteration, crossover and mutation operators are applied to either an intra-cluster or an inter-cluster path, with equal probability. We have chosen a probability of 0.9 for crossover and 0.1 for mutation. In our simulations, small variations in these probabilities did not have a significant impact on the performance of the algorithm.

5. Performance comparison

Efficient MOO algorithms generate (i) many, (ii) non-dominated[2], and (iii) diverse solutions. Several metrics for comparing the performance of *MOO* algorithms have been analyzed [Knowles02, Zitzler03]. Most of these metrics such as the *S* metric and the convergence metric require the knowledge of the true Pareto-optimal front. For the mobile agent routing problem, the *Pareto-optimal set*[3] is unavailable, hence we chose the *C-metric,* the *domination metric,* and the *S-metric*.

The **Set Coverage metric (*C*-metric)** [Zitzler99] calculates the fraction of solutions in one non-dominated set (obtained by one algorithm) that are dominated by those obtained by the other algorithm. If *A* and *B* are the sets of candidate solutions, then *C(A,B)= | {b∈B|∃ a∈ A:a»b}| /|B|*. Note that *C(A,B)=1* when every solution in *B* is dominated by solutions in *A,* and *C(A,B)=0* when none of the solutions in *B* is dominated by any element in *A*. Although no conclusive inferences can be drawn in general from the C-metric values, we may argue that one algorithm is better than another if C(A,B) is found to be significantly higher than C(B,A) over many trials.

The **Domination metric** [Rajagopalan04a] is defined as

$$Dom(A,B) = d(A,B)/(d(A,B)+d(B,A)),$$
$$\text{where } d(X,Y)=\sum_{x \in X} | \{y\in Y| x»y\} |.$$

Mutually non-dominating solution pairs are ignored in calculating the dominance factor *d(A,B)*. Note that *Dom(B,A)=1-Dom(A,B),* and if each solution of algorithm *A* dominates every solution produced by algorithm *B* then *Dom(A,B)=1.*

The **Spacing (S)** metric [Schott95] determines the uniformity of spacing between neighboring solutions obtained by an algorithm. Formally, $S = \sqrt{(1/(n-1)) \times \sum_{i=1}^{n} (d_i - \bar{d})^2}$ where *n* is the number of non-dominated solutions in the archive, d_i is the sum of difference in objective function values between solution *i* and its two nearest neighbors for each objective ([Deb00]), and $\bar{d}$ is the mean value of these distance measures. A lower value of this metric indicates that the non-dominated solutions are uniformly spaced.

6. Simulation Results

We performed simulations on heterogeneous sensor networks of different sizes and distribution patterns. The sensors were randomly deployed within each cluster of the network. Targets were placed at random locations in the sensor field. The sensor parameters for data acquisition and wireless channel are summarized in Table 1.

---

[2]A solution vector *a dominates b,* written a»b, if and only if $\forall i \in \{1,....m\} : f_i(a) \geq f_i(b)$, and $\exists j \in \{1,....m\} : f_j(a) > f_j(b)$.

[3]The *Pareto-optimal front* consists of candidate solutions not dominated by any others.

Table 1: Parameters for mobile agent routing

Mobile agent size	400 bytes
Average data size	100 bytes
Transmitter gain	2
Receiver gain	2
Carrier frequency	2.4 GHz
Channel width	16 bits
Channel operation frequency	20 KHz
Transmitter power range	200-1000 mw
Operational power range	100 - 500 mw
Data acquisition & processing time	50-100 ms

The data processing time and power level of each sensor is chosen randomly from the specified ranges. Experiments were performed for different network architectures. In each cluster, r randomly chosen sensors were designated as faulty where $r = (s\text{-}1)/3$ and s is the total number of sensors in the cluster.

Our simulation results show that the *MOO* approach is successful for solving the mobile agent routing problem as compared to the *WGA*, a generalization of the approach pursued by Wu et al [Wu04]. The solutions obtained by the *MOO* algorithms have much higher quality compared to the solutions obtained by *WGA*. For instance, for a 500 node MADSN, EMOCA obtains a path with a detection accuracy of 2038, energy consumption of 1 unit and path loss of 9,695,016. On the other hand, the path obtained by WGA has a detection accuracy of 310, energy consumption of 2.7 units and a path loss of 83,701,064.

We executed EMOCA and NSGA-II for 1000 generations in each trial; further execution resulted in no improvements. In NSGA-II, we used a virtual archive that stores new non-dominated solutions obtained at every generation. The results are presented in Table 2 (averages over 30 trials). An algorithm A that produces a set of mutually non-dominating solutions A is considered to be better than algorithm B that produces set B iff $C(A,B)$ is high, $C(B,A)$ is low, $Dom(A,B)$ is high, and $S(A) < S(B)$. The results show that the non-dominated solutions obtained by EMOCA are better and more uniformly spaced compared to those obtained by NSGA-II. The results also indicate that EMOCA consistently outperformed NSGA-II in all trials. The computational effort for EMOCA and NSGA-II are similar. Both algorithms required an average of 800 generations before all non-dominated solutions were discovered in the archive.

We have also compared the performance of EMOCA with a weighted genetic algorithm (WGA) in which all three objectives were normalized and combined into weighted single objective function. WGA employs identical genetic operators as EMOCA with binary tournament selection and an elitist steady-state replacement strategy. The results are presented in Table 3. The results clearly indicate that EMOCA outperforms WGA in all trials with a *C-metric* value of 0 and a *Dom-metric* value of 1. The solutions obtained by WGA correspond to the non-dominated set obtained with 100 randomly chosen weight vectors in the interval [0,1]. For each weight vector, WGA converged within 1000 generations. The computational effort of WGA is significantly higher than EMOCA in order for it to discover the non-dominated solutions corresponding to 100 different weight vectors. (100,000 generations as compared to 800 generations required by EMOCA).

We also compared the performance of EMOCA for randomized median filtering (RMF) and randomized censored averaging (RCA) approaches using both the *C metric* and *Dom-metric* values. The results are presented in Table 4, with a minor abuse of notation: the labels in the parentheses following "*C*" or "*Dom*" indicate the approaches used to obtain the sets being compared. The results indicate that the randomized censored averaging approach outperforms the randomized median filtering in all experiments. For instance, in networks with 100 and 600 nodes, C(RMF,RCA) are zero indicating that none of the solutions obtained by RCA are dominated by the solutions obtained by RMF.

Figures 4 and 5 show the non-dominated solutions obtained by EMOCA and NSGA-II with the RCA approach. The plots indicate that EMOCA discovers several non-dominated solutions with higher quality compared to NSGA-II. Figures 6 and 7 show the non-dominated solutions obtained by EMOCA and NSGA-II with the RCA approach, indicating the tradeoffs between path loss and detected signal energy. The plots also indicate that EMOCA is able to find routes with high detected signal energy, low energy consumption and low path loss. For instance, one of the non-dominated solutions has a detected energy value of 150 units with a very low energy consumption of 0.5 units and a path loss of 500000. The plots show that the non-dominated solutions obtained have good diversity with a large spread in the objective space which is confirmed by the *S* metric values in Table 3.

Table 2: EMOCA (*E*) versus NSGA-II *(N)* using *C*, *Dom* and *S* metrics: All results are averages over 30 trials.

Problem parameters: (no of targets, clusters, sensors per cluster)	Randomized median filtering					Randomized censored averaging				
	C(N,E)	*C(E,N)*	*Dom(E,N)*	*S(E)*	*S(N)*	*C(N,E)*	*C(E,N)*	*Dom(E,N)*	*S(E)*	*S(N)*
1,5,20	0.005	0.95	0.67	0.002	0.07	0.0047	0.81	0.70	0.005	0.26
2,10,20	0.16	0.84	0.85	0.02	0.09	0.10	0.78	0.73	0.012	0.16
2,10,30	0.11	0.86	0.93	0.006	0.02	0.05	0.76	0.86	0.011	0.14
3,10,40	0.16	0.87	0.75	0.01	0.13	0.18	0.75	0.75	0.008	0.09
3,20,25	0.08	0.83	0.72	0.02	0.38	0.13	0.82	0.77	0.035	0.18
4,30,20	0.11	0.77	0.77	0.06	0.10	0.21	0.86	0.81	0.05	0.13
5,20,35	0.09	0.83	0.84	0.03	0.26	0.19	0.84	0.74	0.03	0.55
5,20,40	0.15	0.86	0.79	0.02	0.27	0.06	0.81	0.79	0.05	0.28
5,30,30	0.28	0.74	0.68	0.05	0.14	0.23	0.82	0.72	0.06	0.25

Table 3: EMOCA (*E*) versus weighted genetic algorithm (*WGA*) using *C* and *Dom* metrics: All results are averages over 30 trials.

Problem parameters : (no of targets, clusters and sensors/cluster)	Randomized median filtering			Randomized censored averaging		
	C(WGA,E)	*C(E,WGA)*	Dom*(E,WGA)*	*C(WGA,E)*	*C(E,WGA)*	*Dom(E,WGA)*
1,5,20	0	0.79	1	0	0.87	1
2,10,20	0	0.84	1	0	0.93	1
2,10,30	0	0.89	1	0	0.89	1
3,10,40	0	0.96	1	0	0.81	1
3,20,25	0	0.96	1	0	0.67	1
4,30,20	0	0.97	1	0	0.92	1
5,20,35	0	0.99	1	0	0.97	0.97
5,20,40	0	0.98	1	0	0.89	1
5,30,30	0	0.95	1	0	0.86	1

Table 4: RCA versus RMF approaches using *C* , *Dom* and *S* metrics: All results are averages over 30 trials.

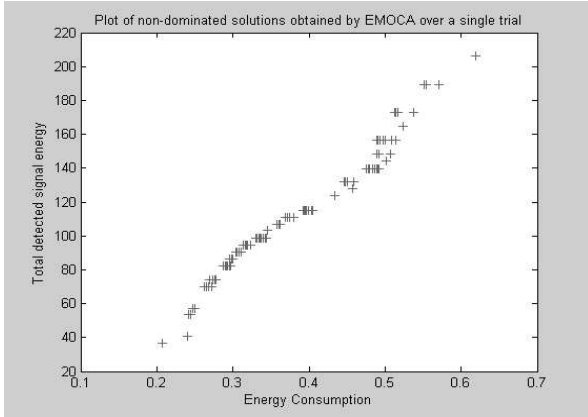| Problem parameters : (no of targets, clusters and sensors/cluster	*C(RMF,RCA)*	*C(RCA,RMF)*	*Dom(RCA, RMF)*	*S(RCA)*	*S(RMF)*
1,5,20	0	0.96	1	0.008	0.023
2,10,20	0.34	0.82	0.66	0.011	0.087
2,10,30	0.17	0.90	0.84	0.005	0.054
3,10,40	0.01	0.80	0.98	0.016	0.097
3,20,25	0.12	0.68	0.68	0.071	0.124
4,30,20	0	0.92	1	0.016	0.942
5,20,35	0.23	0.73	0.58	0.034	0.087
5,20,40	0	0.92	1	0.064	0.091
5,30,30	0.23	0.73	0.58	0.013	0.064

Figure 4: Projections showing the non-dominated solutions obtained by EMOCA along two of the three objectives (energy consumption and detected signal energy) for a 200 node MADSN in one trial
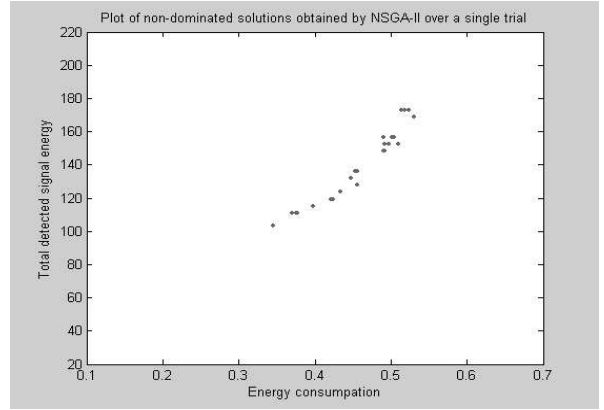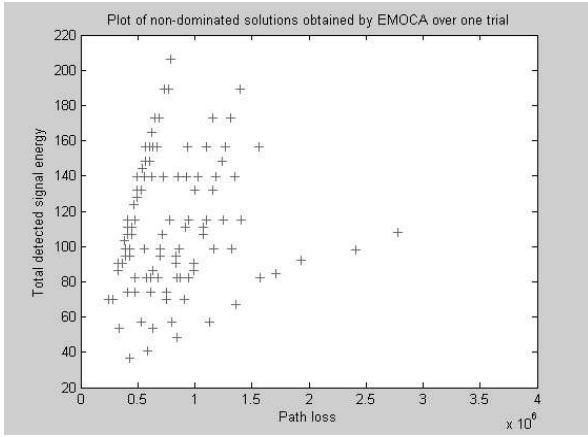

Figure 5: Projections showing the non-dominated solutions obtained by NSGA-II along two of the three objectives (energy consumption and detected signal energy) for a 200 node MADSN in one trial

Figure 6: Projections showing non-dominated solutions obtained by EMOCA for two dimensions (detected signal energy and path loss) for a 200 node MADSN in one trial
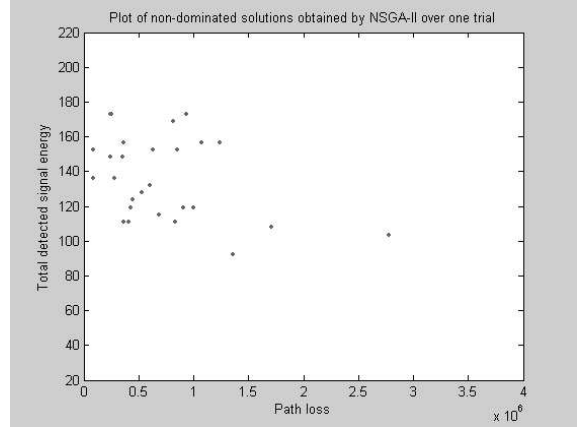

Figure 7: Projections showing non-dominated solutions obtained by EMOCA for two dimensions (detected signal energy and path loss) for a 200 node MADSN in one trial

7. Conclusions

We have developed a multi-objective optimization framework for mobile agent routing in wireless sensor networks. EMOCA and NSGA-II, two recently developed multi-objective evolutionary optimization algorithms, were used to obtain mobile agent routes. Our comparisons with a GA using weighted objectives showed conclusively that EMOCA is much more successful for this problem.

Although comparing MOO algorithms is difficult, our simulation results indicate that EMOCA obtains sets of candidate solutions that appear to be better with respect to quality (relative non-domination) and diversity (spacing of the solutions). The non-dominated solutions obtained illustrate that EMOCA is able to discover satisfying routes with high detected signal energy in the presence of faulty sensors.

We have also proposed two approaches (randomized median filtering and randomized censored averaging) for fault tolerance in MADSNs, and compared their results using EMOCA. The results indicate that the randomized censored averaging approach outperforms the randomized median filtering approach.

Bibliography

[Chan04] H. Chan and A. Perrig, "ACE: An emergent algorithm for highly uniform cluster formation," 2004 European workshop on sensor networks, pp. 154-171.

[Clouqueur01] T. Clouqueur, P. Ramanathan , K.K Saluja and K.C. Wang, "Value fusion versus decision fusion for fault tolerance in collaborative target detection in sensor networks," *Proc. Fusion 2001 conference,* 2001.

[Coello00] Carlos A. Coello Coello, "An updated survey of GA-based multi-objective optimization techniques," ACM computing surveys, 32(2), pp. 109-143, June 2000.

[Coello02] Carlos A. Coello Coello, D. A.Van Veldhuizen and G. B. Lamont, "Evolutionary algorithms for solving multi-objective optimization problems," Kluwer Academic Publishers, New York, March 2002.

[Deb00] K. Deb, S. Agarwal, A. Pratap, and T.Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II", *Proc. Parallel Problem Solving from Nature VI*, pp. 849-858, 2000.

[Friss46] H.T. Friss , "A note on a simple transmission formula," Proc. IRE, 1946.

[Knowles00] Joshua D. Knowles and David W. Corne. "Approximating the nondominated front using the Pareto Archived Evolution Strategy," *Evol. Computation* 8(2):149-172, 2000.

[Knowles02] Joshua D. Knowles and David W. Corne. "On Metrics for comparing non dominated sets," In *Proceedings IEEE Congress on Evolutionary Computation* (CEC02)*,* pages 711-716, 2002.

[Lamport82] L. Lamport, R. Shostak and M. Pease, "The Byzantine generals problem," *ACM Trans. Programming languages and Sytems*, vol. 4, no. 3, pp. 382-401, July 1982.

[Qi01] H. Qi, S.S. Iyengar, and K. Chakrabarty, "Multi-resolution data integration using mobile agents in distributed sensor networks," *IEEE Trans. Systems, Man and Cybernetics Part C: Applications and Rev.,* vol. 31, no. 3, pp. 383-391, Aug. 2001.

[Rajagopalan04a] R. Rajagopalan, "Path planning with evolutionary algorithms," M.S. Thesis, Department of Electrical Engineering & Computer Science, Syracuse University, July 2004.

[Rajagopalan04b] R. Rajagopalan, C. K. Mohan, K. G. Mehrotra and P. K. Varshney, "Evolutionary Multi-objective Crowding Algorithm for Path Computations," *Proc. Fifth International Conference on Knowledge Based Computer Systems*, pp. 46-65, Dec. 2004.

[Rappaport01] T.S. Rappaport, *Wireless Communications: Principles and Practice,* Prentice-Hall , second edition, Dec. 2001.

[Schott95] J. R. Schott, "Fault tolerant design using single and multi-criteria genetic algorithm optimization," Master's Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, May 1995.

[Wu04] Q. Wu, N.S.V Rao, J. Barhen, S.S. Iyengar, V.K. Vaishnavi, H. Qi and K. Chakrabarty, "On computing mobile agent routes for data fusion in distributed sensor networks," *IEEE Trans. Knowledge and Data Engineering,* vol. 16, no. 6, pp. 740-753, June 2004.

[Zitzler99] E. Zitzler, "Evolutionary Algorithms for multiobjective optimization: Methods and applications," Ph.D. Thesis, Zurich, Switzerland: Swiss Federal Institute of Technology (Dissertation ETH No:13398), 1999.

[Zitzler03] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V.G.da Fonseca, "Performance Assessment of Multiobjective Optimizers: An Analysis and Review," *IEEE Trans. Evolutionary Computation,* 7(2):117-132, Apr. 2003.