Evolutionary and Noise-aware Data Gathering
for Wireless Sensor Networks

Bingchun Zhu[1], Junichi Suzuki[1], and Pruet Boonma[2]

[1] Department of Computer Science
University of Massachusetts, Boston, USA
`{numchun,jxs}@cs.umb.edu`
[2] Department of Computer Engineering
Chiang Mai University, Thailand
`pruet@eng.cmu.ac.th`

Abstract. This paper formulates a prioritized data gathering problem in noisy wireless sensor networks (WSNs) and solves the problem with a noise-aware evolutionary multiobjective optimization algorithm (EMOA). Unlike existing local search heuristics, the proposed algorithm can seek the Pareto-optimal routing structures with respect to conflicting optimization objectives. Simulation results demonstrate that the proposed algorithm outperforms a traditional EMOA in a noisy WSN.

Key words: Wireless sensor networks, Data gathering protocol, Noisy multiobjective optimization problem, Genetic algorithm

1 Introduction

Data gathering is one of the most fundamental tasks in wireless sensor networks (WSNs): collecting sensor data from all of, or a subset of, sensor nodes in the network on a periodic or on-demand basis [1, 2]. It it known that a problem of finding the optimal data gathering solution(s) is NP-complete [2]. This paper focuses on a prioritized data gathering problem in multi-hop and semi-static WSNs. The problem is to construct the routing structures for fixed-sized query packets to visit a subset of nodes in the network, starting and ending at the base station, transmit query packets through the constructed routing structures to collect sensor data from visited nodes, and deliver collected data to the base station. In this problem, different priorities are given to different nodes; higher priorities indicate higher requirements for query packets to visit.

This paper describes a data gathering protocol, called DGP Boston (DGPB), which uses source routing to seek the optimal routing structures for prioritized data gathering in WSNs. DGPB is designed to make two key contributions. First, DGPB seeks the Pareto-optimal routing structures with respect to multiple conflicting optimization objectives such as data yield, latency and energy consumption. Since optimization objectives often conflict with each other in WSNs [3], it is important to examine the optimal trade-offs (i.e., Pareto optima) among those objectives. However, as discussed in Section 5, no existing data gathering protocols do not study that. The second contribution of DGPB is that

it considers noise in environmental and protocol parameters such as packet loss rate and node priority. No existing data gathering protocols assume noise in their parameters. (See Section 5.) This paper describes and evaluates an evolutionary and noise-aware multiobjective optimization algorithm in DGPB.

2 Problem Statement

This paper uses the following notations to state a data gathering problem in question. A WSN is considered as a graph $G(N, L)$.

- $N = \{n_0, n_1, ..., n_m\}$ is a set of nodes in G, where n_0 is the base station. $N' = N - \{n_0\}$ is a set of m sensor nodes. This paper assumes semi-static WSNs [4,5]; nodes are stationary and $|N|$ does not change dynamically.
- $M \subseteq N'$ is a set of nodes that query packets gather data from. It is referred as a *measurement set*. M is chosen manually or via techniques such as [6,7].
- $L = \{(n_i, n_j)|n_i, n_j \in N; i \neq j\}$ is a set of links in G. A link is established from n_i to n_j if n_i can directly transmit a packet to n_j without intermediate nodes. Thus, nodes in G are not fully connected with L.
- r_{ij} is the packet loss rate on a link (n_i, n_j). Due to the asymmetric nature of link quality, $r_{ij} = p_{ij} \times p_{ji}$, where p_{ij} denotes the loss rate to transmit a packet from n_i to n_j. $t_{ij} = \frac{1}{(1-p_{ij}) \times (1-p_{ji})}$ is the expected number of transmissions, including transmission retries, to successfully deliver a packet on (n_i, n_j).
- $G_M(M, P) \subseteq G$ is a graph that consists of M. $P = \{(n_i, n_j)|n_i, n_j \in M; i \neq j\}$ is a set of paths in G_M. A path is a sequence of one or more links in L. It is established as the minimum loss rate path from n_i to n_j if n_i can transmit a packet to n_j through a single-hop or multi-hop path. Thus, nodes in G_M are fully connected with P.
- d_i is a demand (or priority) assigned to $n_i \in M$.
- g is the number of query packets used for data gathering. Each packet can carry a limited size S of data due to the limitation of packet size. This is a constraint on how many nodes a packet can collect data from.
- R_q is a route that a query packet q follows to gather data through the paths in P. It is a sequence of nodes in M, starting and ending at n_0. l_{R_q} is the number of links in R_q. $T_{R_q} = \sum_{n,n' \in R_q} t_{nn'}$ is the expected total number of transmissions to route a packet in R_q. (n' is the next hop node of the node n in R_q.) $R = \{R_1, R_2, ..., R_q, ..., R_g\}$ is a route set.

A data gathering problem this paper addresses is to find a set of Pareto-optimal route sets that minimize the following objective functions:

1. *Latency*: indicates the time required for data gathering. It is computed as $\max\{T_{R_1}, ..., T_{R_q}, ..., T_{R_g}\}$.
2. *Per-demand energy consumption*: indicates the amount of energy consumed to collect a unit demand. It is computed as $\sum_{R_i \in R} T_{R_i} / \sum_{R_i \in R} \sum_{j \in R_i} d_j$.
3. *Per-hop packet loss rate*: indicates the average packet loss rate in a route set. It is computed as $\sum_{R_i \in R} \sum_{n,n' \in R_i} r_{n,n'} / \sum_{i=1}^{g} l_i$, where n' is the next hop node of the node n in R_i.

3 DGP Boston

DGPB's algorithmic structure follows NSGA-II, an existing evolutionary multi-objective optimization algorithm [8]. Due to space limitation, this section focuses on a set of extensions that DGPB makes on NSGA-II.

3.1 Individuals

In DGPB, each individual is a variable-length representation of packet routes as shown in Fig. 1. It encodes the number of routes (i.e., the number of packets) and the order of nodes visited by each packet.
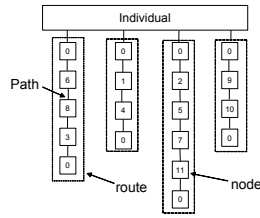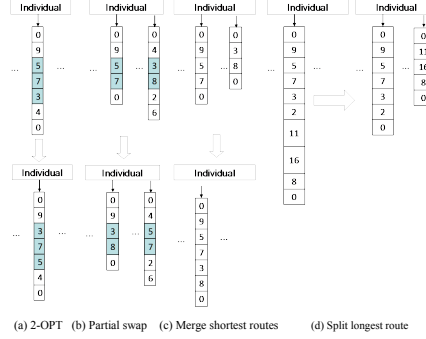

Fig. 1. An Example Individual

(a) 2-OPT   (b) Partial swap   (c) Merge shortest routes   (d) Split longest route

Fig. 2. Mutation Operators

3.2 Crossover and Mutation Operators

BGPB adopts partially-mapped crossover (PMX) [9] as its crossover operator. It also uses the following eight mutation operators. (See Fig. 2 for four of them.):

- *Add*: randomly chooses a node from the measurement set and inserts it to a randomly-selected place in a randomly-selected route. This operator ensures that the inserted node is not redundant with the other nodes in the route.
- *Delete*: removes a randomly-selected node from a randomly-selected route.
- *Exchange*: randomly chooses a node in a randomly-selected route and replaces it with a node selected from the measurement set randomly. This operator ensures that the new node is not redundant with the other nodes in the route.
- *Swap*: exchanges the positions of two randomly-selected nodes in a route, which is also selected randomly.
- *Inversion (2-OPT)*: randomly chooses two cut points in a randomly-selected route and reverses the the order of nodes between the cut points.
- *Partial swap*: randomly selects a subsequence of nodes in each of randomly-selected two routes, and swaps two subsequences between the two routes.
- *Merge the shortest routes*: identifies the two shortest routes and appends one of them to the other.

– *Split the longest route*: identifies the longest route and splits it to two routes at a randomly-chosen point.

DGPB classifies these eight mutation operators to three categories. The first category consists of *Add*, *Delete* and *Exchange*. They alter the nodes an individual visit. The second category consists of *Swap*, *Partial swap* and *2-OPT*, which alter the order of nodes that an individual visits. The remaining two operators are in the third category. They alter the number of routes that an individual have. These three categories have the same probably to be used. In each category, mutation operators are selected randomly.

3.3 The α-Dominance Operator

As discussed in Section 2, BGPB considers noise in its objective functions. The noise interferes with a dominance operator, which determines dominance relationships among individuals. For example, the operator may mistakenly judge that an inferior individual dominates an superior one. Defects in a dominance operator significantly degrades optimization performance [10].

To address this issue, DGPB replaces NSGA-II's classical dominance operator with a new noise-aware operator, called the α-dominance operator. Given objective value samples of two individuals, the α-dominance operator estimates the impacts of noise on objective values and determines whether it is statistically confident enough to judge a dominance relationship between the two individuals.

Individual A is said to α-*dominate* individual B (i.e., $A \succ_\alpha B$) iif:

1. A's and B's samples are classifiable with the confidence level of α, and
2. $\mathcal{C}(A,B) = 1 \wedge \mathcal{C}(B,A) < 1$.

In order to examine the first condition, the α-dominance operator classifies A's and B's objective value samples with Support Vector Machine, and measures a classification error (Step 1 in an example in Fig. 3). The error (e) is computed as the ratio of the number of miss-classified samples to the total number of samples. For evaluating confidence level (α) in the classification error, the α-dominance operator computes the classification error's confidence interval: $e_{int} = e \pm t_{\alpha,n-1}\sigma$ where $t_{\alpha,n-1}$ denotes a single-tail t-distribution with α confidence level and $n-1$ degrees of freedom. n denotes the total number of samples. σ is the standard deviation of e, which is approximated as $\sigma \cong \sqrt{\frac{e}{n}}$. If e_{int} is significant (i.e., if e_{int} does not span zero), the α-dominance operator cannot classify A's and B's samples with the confidence level of α. Thus, the operator determines that A and B do not α-dominate with each other (Step 2 in Fig. 3).

If e_{int} is not significant (i.e., if e_{int} spans zero), the α-dominance operator can classify A's and B's samples with the confidence level of α. Thus, the operator examines the aforementioned second condition with $\mathcal{C}$-metric [11]: $\mathcal{C}(A,B) = |\{b \in B \mid \exists a \in A : a \succ b\}|/|B|$ where $\succ$ denotes a classical notion of dominance [8]. A sample $a \in A$ is said to dominate a sample $b \in B$ (i.e., $a \succ b$) iff a's objective values are superior than, or equal to, b's in all objectives, and a's objective values are superior than b's in at least one objective. $\mathcal{C}(A,B)$ denotes the fraction of

B's samples that at least one sample of A dominates. If $\mathcal{C}(A, B) = 1$, all of B's samples are dominated by at least one sample of A. The α-dominance operator determines $A \succ_\alpha B$ if $\mathcal{C}(A, B) = 1$ and $\mathcal{C}(B, A) < 1$. If $\mathcal{C}(A, B) < 1$ and $\mathcal{C}(B, A) < 1$, it determines neither $A \succ_\alpha B$ nor $B \succ_\alpha A$. See Step 2 in Fig. 3.

Fig. 3 shows an example that determines the α-dominance relationship between two individuals, A and B, with two objectives, f_1 and f_2, to be minimized. A and B have seven samples each. The first step is to classify these 14 samples with SVM and compute e_{int}. Suppose SVM produces a classification vector as shown in Fig. 3. Two samples of B are miss-classified; $e = \frac{2}{14}$ (0.143). Thus, $\sigma \cong \sqrt{\frac{0.143}{14}} = 0.1$. Assuming $\alpha = 0.95$, $e_{int} = 0.143 \pm 1.771*0.1 = 0.143 \pm 0.1771$. Since e_{int} spans zero, A's and B's samples are classifiable with the confidence level of 95%. The second step is to compute $\mathcal{C}$-metric: $\mathcal{C}(A, B) = 1$ and $\mathcal{C}(B, A) = 2/14 < 1$ Therefore, the α-dominance operator determines $A \succ_\alpha B$.
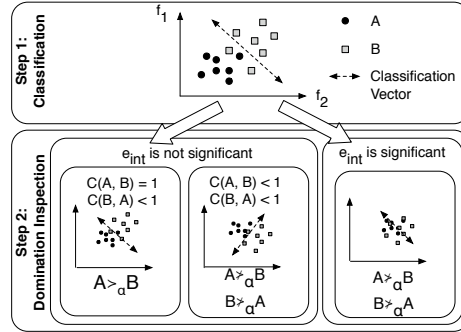

Fig. 3. An Example α-Dominance Operation

4 Preliminary Simulation Evaluation

This section evaluates DGPB, through simulations, in a WSN that have noise in packet loss rate and demand. The highest demand (100) is given to an event reporting node. Lower demands (85, 70 and 55) are given to the nodes that are 1-hop, 2-hops and 3-hopes away from the event reporting node. These nodes and the event reporting node are included in the measurement set M ($|M| = 79$). Table 1 shows a set of parameter values used in simulations. The α-dominance operator performs C-support vector classification with a linear kernel in its SVM.

The noise on demands follows a uniform distribution [-50,50]. The noise on packet loss rate is given based on an empirical experimental result in [4].

Table 2 compares DGPB with NSGA-II and four combinations of existing heuristics: nearest neighbor route construction (NN), 2-OPT local search and Clarke-Wright Savings algorithm (CWS). Performance metrics are the total number of data transmissions as well as the three objectives described in Section 2. The average results indicate the average of 20 simulation runs. The values in parentheses show standard deviations. Table 2 illustrates that DGPB outperforms NSGA-II except for per-hop packet loss rate. It outperforms four combinations of exiting heuristics in all performance metrics.

Table 1. Parameter Values used in Simulations

Parameter	Value	Parameter	Value
Observation area	200m×200m	Population size	100
The number of nodes	150	Max # of generations	500
Communication range	30m	Crossover rate	0.9
Base station	(0,100)	Mutation rate	0.4
Event reporting node	(100, 100)	Stopping criteria in SVM	$1e^{-3}$
Max # of data stored in a packet	10	C in SVM	1
Packet loss rate on each link	0.05	# of samples in α-dominance	20

Table 2. Objective Values

	Average				Maximum				Minimum			
	#Trans	Obj1	Obj2	Obj3	#Trans	Obj1	Obj2	Obj3	#Trans	Obj1	Obj2	Obj3
NN	$58.7_{(31.1)}$	$12.71_{(2.01)}$	$0.018_{(0.002)}$	$0.371_{(0.017)}$	104.4	22.14	0.083	0.71	4.5	5.30	0.007	0.16
NN+2OPT	$48.7_{(28.2)}$	$12.19_{(1.09)}$	$0.017_{(0.001)}$	$0.277_{(0.016)}$	93.0	25.23	0.079	0.71	4.2	5.62	0.011	0.15
CWS+NN	$64.2_{(32.4)}$	$14.79_{(2.66)}$	$0.031_{(0.002)}$	$0.404_{(0.021)}$	108.8	24.65	0.084	0.83	10.5	4.36	0.014	0.20
CWS+NN +2OPT	$52.4_{(27.1)}$	$13.85_{(2.31)}$	$0.020_{(0.001)}$	$0.375_{(0.018)}$	97.7	24.61	0.079	0.79	9.2	4.03	0.013	0.18
NSGA-II	$5.18_{(1.54)}$	$4.87_{(1.42)}$	$0.019_{(0.008)}$	$0.016_{(0.006)}$	7.92	7.92	0.004	0.057	2.26	2.26	0.013	0.001
DGPB	$4.26_{(1.56)}$	$3.75_{(1.09)}$	$0.017_{(0.008)}$	$0.052_{(0.002)}$	7.88	7.88	0.022	0.079	3.11	3.11	0.013	0.03

Obj1: Latency; Obj2: Per-demand energy consumption; Obj3: Per-hop packet loss rate

Table 3. Quality Indicators

	Hyper Volume	Scott's Spacing	U-Measure	# of Non-dominated Individuals
NSGA-II	0.53	0.381	0.72	49.2
DGPB	0.75	0.207	0.61	100

Table 3 compares the non-dominated individuals produced by NSGA-II and DGPB. Hypervolume [11] indicates the optimality and spread of individuals in the objective space. DGPB yields 42% greater volume than NSGA-II. Scott's Spacing [12] and U-measure [13] indicate the distribution of individuals in the objective space. (A smaller value means that individuals are distributed more evenly.) DGPB evolves individuals more evenly in the objective space than NSGA-II. In 500 generations, DGPB evolves all 100 individuals to be non-dominated, while NSGA-II evolves less than half of it. This means that DGPB maintains a stronger evolution pressure to converge individuals. As demonstrated in Table 3, DGPB outperforms NSGA-II in all comparison metrics.

5 Related Work

This work is an extension to [14], which formulates data gathering in WSNs as a VRP and solves it with an existing heuristic algorithm (CWS). As shown in Table 2, DGPB outperforms CWS-based algorithms. Moreover, DGPB considers three conflicting optimization objectives, while [14] considers energy consumption as an objective and latency as a constraint. [14] does not consider prioritized data gathering in noisy WSNs.

Meliou el al. [2] formulate data gathering as a TSP and solve it with a TSP approximation method. They consider a single optimization objective: energy consumption. Prioritized data gathering in noisy WSNs is out of their scope.

Several evolutionary multiobjective optimization algorithms (EMOAs) exist to solve VRPs. Ombuki et al. [15] study an EMOA for a VRP, and Tan et al. [16] study an EMOA for a VRP with stochastic demands. Both work assume fully-connected graphs. Since nodes in a WSN are often not fully-connected, DGPB reduces a WSN to a fully-connected graph with the notion of shortest paths. DGPB considers an extra stochastic (or noisy) parameter, packet loss rate, which Tan et al. do not.

In the field of EMOAs, several existing dominance operators consider noise in objective functions [17,18]; however, all of them assume particular noise distributions. For example, [19–21] assume normal distributions. [22] assumes a uniform distribution. [23, 24] assume Poisson distributions. Given a noise distribution, each of existing noise-aware dominance operators statistically estimates each individual's objective value by collecting its samples. In contrast, the α-dominance operator assumes no noise distributions a priori because it is not realistic to predict and model them in WSNs. Instead of estimating each individual's objective values, the α-dominance operator measures the effect of noise on objective value samples and determines whether it is confident enough to compare individuals.

6 Conclusion

This paper proposes considers WSN application required to simultaneously minimize latency, energy consumption and per-hop packet loss rate in a noisy environment. A noisy-aware multiobjective evolutionary algorithm, NSGA-IIA, is proposed and applied to the problem. Evolution results show that NSGA-IIA can find better nondominated individuals in terms of convergence and diversity.

References

1. B. Krishnamachari, "Modeling data gathering in wireless sensor networks," in *Wireless Sensor Networks and Applications*.   Springer, 2007.
2. A. Meliou, D. Chu, J. Hellerstein, C. Guestrin, and W. Hong, "Data gathering tours in sensor networks," in *Proc. of ACM/IEEE IPSN*, 2006.

3. Q. Han, D. Hakarrinen, P. Boonma, and J. Suzuki, "Quality-aware sensor data collection," *Int'l Journal of Sensor Networks*, vol. 7, no. 3, pp. 127–140, 2010.
4. A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *Proc. SenSys*, 2003.
5. J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proc. SenSys*, 2003.
6. A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proc. VLDB*, 2004.
7. H. Wada, P. Boonma, and J. Suzuki, "Chronus: A spatiotemporal macroprogramming language for autonomic wireless sensor networks," in *Autonomic Network Management Principles: From Concepts to Applications*. Elsevier, in press.
8. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6(2), 2002.
9. D. Goldberg and R. Lingle, "Alleles, loci and the traveling salesman problem," in *Proc. 1st Int. Conf. On Genetic Algorithms*, 1985, pp. 154–159.
10. L. Bianchi, M. Dorigo, L. Gambardella, and W. Gutjahr, "A survey on metaheuristics for stochastic combinatorial optimization," *Natural Computing*, vol. 8(2), 2009.
11. E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," *IEEE Trans. Evol. Comput.*, 3(4).
12. J. Knowles and D. Corne, "On metrics for comparing nondominated sets," in *Proc. World on Congress on Computational Intelligence*, 2002.
13. Y.-P. Wang, Y. wing Leung, Y. ping Wang, and Y. ping Wang, "U-measure: A quality measure for multiobjective programming," Technical Report, Hong kong Baptist University, 2003.
14. P. Boonma, Q. Han, and J. Suzuki, "Leveraging biologically-inspired mobile agents supporting composite needs of reliability and timeliness in sensor applications," in *Proc. IEEE FBIT*, 2007.
15. B. Ombuki, B. J. Ross, and F. Hanshar, "Multi-objective genetic algorithms for vehicle routing problem with time windows," *Applied Intelligence*, vol. 24, 2006.
16. K. C. Tan, C. Y. Cheong, and C. K. Goh, "Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation," *European Journal of Operational Research*, vol. 177, no. 2, 2007.
17. H.-G. Beyer, "Evolutionary algorithms in noisy environments: Theoretical issues and guidelines for practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 186(2–4), 2000.
18. Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments: a survey," *IEEE Trans. Evol. Comput.*, vol. 9(3), 2005.
19. C. K. Goh and K. C. Tan, "Noise handling in evolutionary multi-objective optimization," in *Proc. of IEEE CEC*, 2006.
20. H. Eskandari, C. D. Geiger, and R. Bird, "Handling uncertainty in evolutionary multiobjective optimization: SPGA," in *Proc. of IEEE CEC*, 2007.
21. M. Babbar, A. Lakshmikantha, and D. E. Goldberg, "A modified NSGA-II to solve noisy multiobjective problems," in *Proc. of ACM GECCO*, 2003.
22. J. Teich, "Pareto-front exploration with uncertain objectives," in *Proc. of Int'l Conf. on Evol. Multi-Criterion Optimization*, 2001.
23. M. Wormington, C. Panaccione, K. M. Matney, and D. K. Bowen, "Characterization of structures from x-ray scattering data using genetic algorithms," *JSTOR Philosophical Transactions*, vol. 357, no. 1761, pp. 2827–2848, 1999.
24. K. Delibrasis, P. Undrill, and G. Cameron, "Genetic algorithm implementation of stack filter design for image restoration," *IEE Proc. VISP*, vol. 143(3), 1996.