PDF Download
2576768.2598384.pdf
29 January 2026
Total Citations: 7
Total Downloads: 149

Published: 12 July 2014

Citation in BibTeX format

GECCO '14: Genetic and Evolutionary Computation Conference
July 12 - 16, 2014
BC, Vancouver, Canada

Conference Sponsors:
SIGEVO

RESEARCH-ARTICLE

The tradeoffs between data delivery ratio and energy costs in wireless sensor networks: a multi-objectiveevolutionary framework for protocol analysis

DOINA BUCUR, University of Groningen, Groningen, Groningen, Netherlands

GIOVANNI IACCA

GIOVANNI SQUILLERO, Polytechnic of Turin, Turin, TO, Italy

ALBERTO PAOLO TONDA, Paris-Saclay Food and Bioproduct Engineering, Massy, Ile-de-France, France

GECCO '14: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation (July 2014)
https://doi.org/10.1145/2576768.2598384
ISBN: 9781450326629

The Tradeoffs between Data Delivery Ratio and Energy Costs in Wireless Sensor Networks: A Multi-Objective Evolutionary Framework for Protocol Analysis*

Doina Bucur
University of Groningen
Nijenborgh 9, 9747 AG
Groningen, The Netherlands
d.bucur@rug.nl

Giovanni Iacca
INCAS^3
Dr. Nassaulaan 9, 9401 HJ
Assen, The Netherlands
giovanniiacca@incas3.eu

Giovanni Squillero
Politecnico di Torino
C.so D. d. Abruzzi 24, 10129
Torino, Italy
giovanni.squillero@polito.it

Alberto Tonda
INRA UMR 782 GMPA
1 Av. L. Brétignières, 78850
Thiverval-Grignon, France
alberto.tonda@grignon.inra.fr

ABSTRACT

Wireless sensor network (WSN) routing protocols, e.g., the Collection Tree Protocol (CTP), are designed to adapt in an ad-hoc fashion to the quality of the environment. WSNs thus have high internal dynamics and complex global behavior. Classical techniques for performance evaluation (such as testing or verification) fail to uncover the cases of extreme behavior which are most interesting to designers. We contribute a practical framework for performance evaluation of WSN protocols. The framework is based on multi-objective optimization, coupled with protocol simulation and evaluation of performance factors. For evaluation, we consider the two crucial functional and non-functional performance factors of a WSN, respectively: the ratio of data delivery from the network (DDR), and the total energy expenditure of the network (COST). We are able to discover network topological configurations over which CTP has unexpectedly low DDR and/or high COST performance, and expose full Pareto fronts which show what the possible performance tradeoffs for CTP are in terms of these two performance factors. Eventually, Pareto fronts allow us to bound the state space of the WSN, a fact which provides essential knowledge to WSN protocol designers.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Protocol verification, Routing protocols*; C.4 [**Performance of Systems**]: [Design studies]; G.1.6 [**Numerical Analysis**]: Optimization—*Global optimization*

General Terms

Performance, Algorithms

Keywords

Evolutionary Algorithms; Multi-objective Optimization; Wireless Sensor Networks; Collection Tree Protocol; Performance Evaluation; Ad-hoc Routing; Data Delivery Ratio; Energy Consumption

1. INTRODUCTION

Wireless sensor networks (WNSs) are distributed, energy-constrained – often battery powered – self-organizing systems most often employed for autonomous data collection in challenging communication environments, as part of monitoring applications in nature or industry. WSNs are paradigmatic examples of complex systems embedded in the environment: while each node in the network is a constrained embedded system, the global behavior of the network depends upon the physical *topology* of the network (i.e., the set of viable wireless communication links) and upon the *environmental conditions* (e.g., the pattern of added environmental noise which can impede wireless communication between nodes). It is important to understand precisely which type of topology combined with which type of environmental conditions may cause a WSN infrastructure to fail at its task.

The performance evaluation of WSN routing protocols is extremely difficult, since protocol designers have very limited information about the cause of any malfunction which occurred in a remote deployment: "We frequently failed to understand performance results and could not determine who was to blame (i.e., the testbed characteristics, or the routing layer?)" [17]. Some WSN protocols which performed well in controlled environments had as low as 2% data delivery in the field [9, 26]. An inability to reproduce faulty scenarios was also stated: "it is unclear why collection per-

*All authors contributed equally and their names are presented in alphabetical order.

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
Copyright 2014 ACM 978-1-4503-2662-9/14/07 ...$15.00.
http://dx.doi.org/10.1145/2576768.2598384.

forms well in controlled situations yet poorly in practice, even at low data rates" [14].

This lack of understanding is not surprising. WSN testing is often done on indoor WSN testbeds [4, 14], which form well-connected networks unlikely to reproduce the type of communication interruptions encountered later in environmental deployments. Furthermore, "experimental results obtained on a single testbed are very difficult to generalize" [17]. Since worst-case scenarios are statistically rare events in the state-space of the problem, non-exhaustive methods, such as testbed analysis [14, 22] or random testing, are likely to miss them.

A semi-automatic method to generate worst-case data-throughput scenarios for a *model* of the IEEE 802.11 MAC protocol was proposed in [3]. The approach is based on a search algorithm through the state-space of an abstract model in state-machine syntax; the modeling makes some simplifying assumptions for some system features, which is a disadvantage of the method. On the other hand, using formal verification for analyzing full *implementations* of protocols is computationally prohibitive, and has met with limited success [21, 27]. At best, it succeeds in identifying unsafe behavior of a WSN protocol in a few concrete, small-size WSN topologies, which makes it impossible to generalize the cause of the behavior.

Recently, Evolutionary Algorithms (EAs) were proven to be an effective technique for protocol analysis in WSNs, as a means to create the network configurations required for exploiting simulation-based techniques [1, 2, 5, 6]. In [5, 6], WSNs were analyzed against a single performance metric: the lifetime of a WSN routing protocol. This metric was used as evaluator for the "interestingness" of network configurations, and served to generate sets of interesting examples of critical WSN topologies with unexpectedly low lifetime. Earlier, [1] presented a seminal work on the use of an evolutionary algorithm for generating a critical test case for a simplified TCP/IP network. In [2], the authors do a search of the state space of a particular WSN application for fire detection, using a fitness function designed such that it guides the search algorithm towards WSN situations in which not only the system fails to detect a fire, but does so because of a minimal number of node failures caused by the environment.

Applying evolutionary computation to verification and testing has also been explored in other domains. Preliminary experiments in [25] suggest that stochastic meta-heuristics are effective in locating the most promising parts of the search space to test complex software modules. A flight system software is verified in [23], where a genetic algorithm outperforms classical random testing. In [13], the operating system of a mobile phone prototype is tested with evolved keyboard inputs, uncovering several power-related bugs.

This work extends the scope of EA-guided WSN performance evaluation by considering multiple fitness values to uncover multi-dimensional performance bounds, and using a *Multi-Objective Evolutionary Algorithm* (MOEA) to solve the associated optimization problem. Such analysis deepens the practical lessons to be learned by WSN designers compared to the single-metric cases previously found in the literature, by providing a complex, heterogeneous set of misbehaving network topologies.

Our experiments focus on the Collection Tree Protocol (CTP) [14] and its mainstream full implementation in the TinyOS [19] embedded operating system; while analyzing the implementation of the protocol, rather than an abstract model of it, adds to the complexity of the problem, it has the great advantage that the results of the analysis can immediately be translated into practical knowledge. To evaluate the performance of CTP, we consider two fitness functions, namely (1) the ratio of data packets delivered successfully by the network (i.e., the most important functional requirement of a WSN), and (2) the energy consumed from the nodes' batteries in the process (i.e., the most important non-functional requirement). By obtaining WSN configurations which maximize or minimize both metrics at the same time, we can draw practical conclusions about the *performance tradeoffs* of the network.

The remaining of this paper is organized as follows: Section 2 summarizes WSN concepts, Section 3 presents the design of our multi-objective search framework, and experimentation is reported in Section 4. Conclusions and future work are outlined in Section 5.

2. WSN DESIGN AND ANALYSIS

This section gives an overview of the design, complexity, and performance factors of the WSN data collection protocol under study.

2.1 Data collection with WSNs

In a WSN, each node is a small, wireless embedded system with limited memory, computation power, communication bandwidth, and available energy. The physical topology of the wireless network typically exhibits heavy link dynamics due to various changes in the environmental conditions. The nodes are deployed at locations of interest and will sense, store and forward data wirelessly to a *sink* node. The sink node has a more reliable connection to data storage, and will forward all collected data.

For successful data collection, WSNs employ ad-hoc collection routing protocols, which form the basic underlying infrastructure to many WSN applications. WSN collection routing aims at adaptively organizing the nodes for data collection into a multi-hop, cycle-free spanning tree rooted in the sink node. In order to design adaptive routing, all nodes broadcast to their immediate neighbors *beacon* messages containing the nodes' local routing knowledge, thus allowing these neighbors to (i) update their estimation of the quality of communication links in the neighborhood, and (ii) extend their routing knowledge with that of the neighbors. On the basis of this, a protocol will then do adaptive route selection, i.e., will choose the route to forward packets based on the current quality of known routes to sink.

Among the distance-vector collection routing mechanisms applied to WSNs, CTP [14] is the de facto standard: it builds on the concept of combining the basic distance-vector, beacon-based mechanism by adding adaptive beacon intervals, designed to broadcast beacons more often in order to quickly reconstruct the routing tree when the network is changing. The combination of heavy link dynamics, continuous link-quality estimation, adaptive beaconing, and adaptive route selection makes CTP a quite complex subject suitable for our analysis. Given that CTP is widely deployed in battery-powered testbeds for environmental, medical or infrastructure monitoring [7, 16, 26], it is also of practical importance that its performance tradeoffs be known.

2.2 Quality metrics for WSN data collection

A number of performance metrics are crucial when analyzing a WSN collection protocol in any environmental configurations. *Functional* performance metrics assess the overall ability of the protocol to deliver all the data packets to the sink node; *non-functional* metrics evaluate how efficiently (in terms of, e.g., delay or energy consumption) data collection is done. Here, we select two critical quality metrics:

1. The *data delivery ratio* (DDR) is the main functional performance metric. This metric is calculated as the fraction of unique data packets sent by all nodes which are received successfully at the sink node.

2. The *total energy cost* (COST) spent in (a) organizing and maintaining a routing tree using beacon messages, and (b) forwarding data packets towards the sink. This is the basic non-functional metric, and is evaluated here via a roughly equivalent heuristic: the total number of network messages sent and received on the network during the interval of experimentation.

In order to uncover novel worst cases for the performance of CTP, we aim to find not only the single worst possible values for DDR or COST, but also more elusive information about the interplay between these performance factors:

- Does DDR generally correlate positively with COST, as intuitively expected from routing protocols, where heavier network communication is expected to lead to better data delivery?

- For an observed value of DDR, what are the upper and lower bounds of COST values over all possible real-world network situations?

To answer these questions, we need to search for those extreme network topologies and environmental conditions which give a performance tradeoff between the network energy consumption and the data delivery ratio. In effect, we need to obtain the *bounds* of the *two-dimensional state space* created by these performance metrics. As we will show in Section 3, this problem can be intuitively formulated in terms of multi-objective optimization, thus representing a natural, yet unconventional, application of MOEAs.

2.3 WSN simulation: stochasticity, topology, and noise injection

We use the open-source simulator TOSSIM [20] to evaluate CTP over different WSNs. TOSSIM is a discrete-event simulator which allows the definition of a controlled WSN environment. It guarantees high-fidelity simulations, from the level of hardware interrupts up to application-level events. Also, it provides a complete Python API which allows full control over the simulation, as well as the collection of the statistics needed for our performance analysis.

In Table 1, we summarize the configuration we use for simulation. We consider dense topologies of 20 nodes booting at simulation time 0; there is randomness in the network stack purposely designed for concurrent boots not to cause network collisions due to synchronicity. Each topology is simulated for 200 seconds, i.e., much longer than the time needed for CTP to form a routing tree (which is less than 1 second [14]), and data packets to be routed to sink. In particular, each node samples an on-board sensor every second, and after bundling 5 readings, forwards them to the sink.

A viable link between two nodes is that which assumes any value above -110 dBm (a threshold particular to the modelling of radio-frequency communication in TOSSIM). We thus have two intervals of interest for the quality (i.e., signal gain) of any network link:

- *strong link*, with a signal gain between -100 and 0 dBm;
- *no link*, modeled with a fixed gain of -200 dBm.

WSNs are affected by a number of stochastic effects, including *noise* on the radio frequencies used for communication, interference among the nodes themselves, packet collision, etc. TOSSIM provides an accurate model of the mainstream radio stack. In addition to that, to further improve the realism of the simulation (e.g., taking into account bursts of interference) it is possible to add a statistical *noise model* over the original links' signal gains. This model is generated automatically from a trace measured experimentally in real-world testbeds, with the generation algorithm based on Closest Pattern Matching [18]. In our experiments, we used two noise traces: one of *light*, and one of *heavy* noise (the `meyer-short` trace available with TOSSIM).

Table 1: Configuration of simulation variables.

Simulation setting	Value
Network size, density	20 nodes, 50%
Sink node	node 0
Simulation time	200 seconds
Rate of data packets	every 5 seconds

A noise model is injected into a simulation by TOSSIM as described in Fig. 1: while a link between two given nodes will be modelled with a fixed signal gain value, a given noise pattern is superimposed to all the links in the network. If the amplitude of the noise gain is higher than that of the signal gain for any time interval, then for that interval the data link becomes unusable. The heavy noise model that we use has peaks of amplitude reaching -40 dBm.

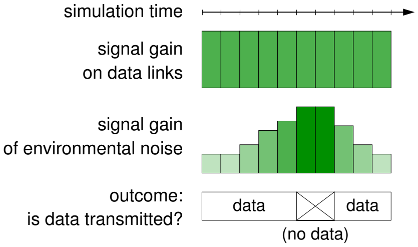

Figure 1: TOSSIM technique for simulating noise on radio frequencies.

Due to these stochastic effects, in order to compute a statistically meaningful value for the WSN performance factors, each evaluated WSN topology is simulated multiple times with different random seeds. Since the standard error of the mean of an n-dimensional sample whose variance is σ is $\sigma/\sqrt{n}$, and applying the central limit theorem to approximate the sample mean with a normal distribution, a sample size $n = 16\sigma^2/W^2$ guarantees a 95% confidence interval of width W. Thus, in order to guarantee confidence interval $W = \sigma$ regardless of the actual value of σ (which is not constant in the search space) we always run $n = 16$ simulations per each WSN evaluation.

2.4 The size of the search space

For WSNs of 20 nodes with a 50% density of strong links, the search space of our problem is defined by the variables listed in Table 2.

Table 2: Variables used and their definition domain.

Variables	Domain of definition for each
380 links	{strong links, no links}
190 strong links	{0} or {-100, -80, -60, -40, -20, 0} dBm
noise model	{light, heavy}

We thus estimate the size of the search space as follows: 10^{113} different combinations exist of choosing 190 strong links out of the total 380 possible links in the network; this is thus the size of the state space for networks with a single gain value for strong links, and with light noise. In the experiments where we inject heavy communication noise, we need to multiply this number by the 6 different gain values which may be given to any strong link (see subsection 3.1).

3. A MULTI-OBJECTIVE PERSPECTIVE

Many real-life optimization problems have more than one aspect to be considered at the same time, i.e., more than one objective to be minimized or maximized. A possible solution (the *scalarization* approach) is to aggregate all objectives into a single function, e.g., through a weighted sum. As an alternative (the *lexicographical* approach), a domain expert might prioritize the objectives according to some order of preference, so that these objectives are evaluated following that predefined order. In both cases, however, an implicit bias is introduced in the problem, that might be harmful, especially if the objectives are conflicting. Moreover, such approaches are not usually able to provide a full set of tradeoff solutions, but rather they tend to offer a few solutions (perhaps only a single solution), which are often not homogeneously distributed in the objective space.

Unlike the fitness scalarization and lexicographical approaches, Multi-Objective Evolutionary Algorithms [10] are based on the assumption that all the objectives of a problem should be considered at the same time. In other words, in order to capture the complexity of the search space of two conflicting objectives, an algorithm cannot provide a single optimal solution, but rather should generate a *Pareto front*, that is, a set of *non-dominated* solutions such that none of the objective functions can be further improved without degrading some other objective value. The user is thus presented a set of optimal solutions, each one corresponding to a different tradeoff between different, conflicting goals. Due to their versatility, EAs have proven a valid tool for solving many hard, real-world multi-objective problems [8].

3.1 Individual description

In our problem, an individual (depicted in Fig. 2) represents a candidate topology of the WSN, encoded as a square matrix $N \times N$, with $N = 20$ the number of nodes in the network. Each position i, j in the matrix holds the gain of the signal between nodes i and j, expressed in dBm units. It is to note that, given the nature of wireless transmitters and receivers, the matrix should not be symmetrical in the general case. In Fig. 2, in the noiseless case, strong links have 0 dBm gain (green cells, gray in print); absent links are assigned -200 dBm (white cells). In the noisy case, strong links take a gain value from a set (green scale, gray scale in print, with darker cells being higher gains). The diagonal elements (node self-connectivity) are not part of the model.

Since we are also interested into investigating the effect of noise on the link performance, we further discretize the sets of possible gain values, depending on the presence of noise in the network. More specifically, in absence of noise, we assign to strong links a fixed, ideal signal gain of 0 dBm: this can be explained considering that links, in the absence of noise, will have the same relative performance regardless of their actual gain value. On the other hand, in the noisy case, we discretize the set of possible signal gains as {-100, -80, -60, -40, -20, 0} dBm: in this case, depending on the link noise generated by the noise model (as described in subsection 2.3) at a point in time, the link performance will strictly correlate to its gain.

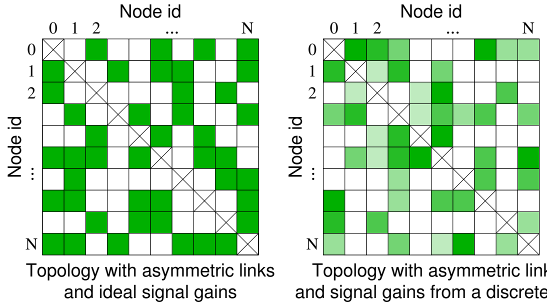
Node id
0 1 2 ... N
0
1
2
Node id
...
N
Topology with asymmetric links and ideal signal gains
Node id
0 1 2 ... N
0
1
2
Node id
...
N
Topology with asymmetric links and signal gains from a discrete set

Figure 2: Visual representation of individual encoding. Noiseless case (left); noisy case (right). Shades of green (grey in print) model signal gain values of strong links; white models the absence of links.

From an evolutionary perspective, this kind of encoding translates into the fact that individuals (i.e., candidate network topologies) present two different *alleles* for each *gene* in their genome, i.e., two possible (sets of) values for each link gain value, corresponding respectively to either strong or no links. Mutation operators (see subsection 3.3) can either (a) switch a gene from one allele to the other or, (b) in the case of a finite set of possible gain values (for noisy strong links), tune the allele selecting one of those values. Thus, an individual with a strong link at a given matrix position is *close* (in the parameter search space) to an individual without a link at the same position, regardless of the actual gain of the link. Moreover, an individual without a link contains in its genome the value of the strong link as non-coding material, thus *remembering* it.

Additionally, a user-defined occurrence probability is associated to each allele, so that it is possible to design an initial population where each individual is characterized by a predetermined percentage of strong links.

3.2 Experimental framework

The framework used for the experiments is summed up in Fig. 3: the evolutionary core creates candidate network topologies for TOSSIM. All events of interest on each node (i.e., the sending and receiving of beacons and data packets at each node), are logged during the simulation, and the logs are then processed to obtain the COST and DDR values.

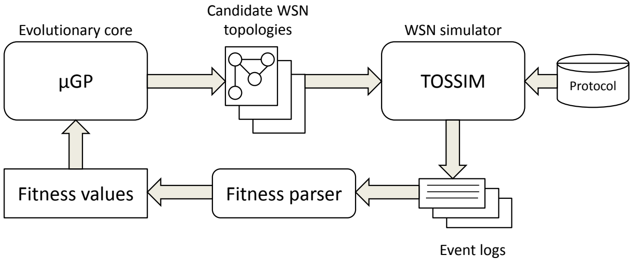

Figure 3: Conceptual scheme of the proposed evolutionary approach.

For our experiments, we specialize a general-purpose MOEA, μGP [24]. Three interesting properties influence our choice: first, since μGP's individuals are represented as a succession of macros with a tunable probability of appearance, it is possible to describe an individual structure such as the one presented in subsection 3.1[1]; second, this EA is capable of managing complex genetic structures, with constant values, integers and reals; and third, the μGP framework can be easily coupled with an external evaluator, which makes the integration with a WSN simulator straightforward.

μGP features built-in support for two or more fitness values, which can be evaluated either in lexicographical order or with a multi-objective approach. The MOEA algorithm makes use of a *crowding distance* metric similar to NSGA-II [12] in order to push the exploration of the Pareto front: for each individual, the closest neighbours are located on the same front in the fitness space; such neighbours are then used as vertices to compute the hypervolume of the resulting cuboid; individuals associated with larger hypervolumes will be favored for reproduction. The underlying idea is that individuals that are the farthest from others are positioned on segments of the Pareto front not yet well explored. By favoring such individuals, an even exploration of the Pareto front is promoted.

3.3 Evolutionary process

The MOEA is executed with the parameter settings shown in Table 3, and the genetic operators listed in Table 4, namely two standard crossover operators (`one-point` and `two-point`), and two mutation operators that act at the level of a single gene. As explained in subsection 3.1, one mutation operator, namely `replacement mutation`, switches one or more genes from one allele to the other (so that an absent link becomes a strong link, and vice versa). On the other hand, `alteration mutation` perturbs on or more genes corresponding to strong links modifying their gains, i.e. selecting one of the other possible values in the predefined range. The latter mutation is effective only in noisy cases, where strong links are associated to a range of 6 possible values (see Table 2).

In addition to that, the algorithm uses a self-adaptation mechanism able to tune the activation of the different genetic operators, and to determine the strength of a mutation operator, thus effectively balancing exploration and exploitation during the evolution. The evolutionary algorithm is configured in such a way that a $\mu+\lambda$ evolution strategy is used. Selection is performed resorting to tournament selection with tournaments of size τ. Each evolutionary process is allotted a computational budget of $1,000$ generations.

Table 3: MOEA parameters used.

Param.	Description	Value
μ	population size	40
λ	no. of operators applied at every step	5
τ	size of tournament selection	2
stop	no. of generations allotted	1000

Table 4: Genetic operators used.

Operators	Description
`alteration mutation`	Change one or more integer values to different integer values (strong links, noisy case)
`replacement mutation`	Substitute one or more integer values (strong links) with the constant value modeling an absent link, or vice versa
`one-point crossover`	Standard one-point crossover
`two-point crossover`	Standard two-point crossover

4. EXPERIMENTAL EVALUATION

In order to obtain the bounds of the state space of CTP with regard to the two performance factors, we laid down the possible configurations for the MOEA: since both the DDR and the COST objectives may be either maximized (denoted ↑) and minimized (denoted ↓) in an experiment, four experimental configurations are possible. Two of these configurations uncovered rich, informative Pareto fronts, while the other two yielded legitimate, but trivial Pareto fronts of few distinct points, due to the shape of the state space. In what follows, we report on the informative experiments.

For each of the two informative configurations, we set up two experiments: without and with heavy radio-frequency noise injection. The resulting list of experiments performed is shown in Table 5.

4.1 Parallelization and runtimes

All experiments have been performed on a 32-core Linux machine (Intel(R) Xeon(R) CPUs, 2.00GHz, 128GB RAM), running Ubuntu 12.04.1 LTS. In order to exploit the inherent parallelism of the objective evaluation (for which every simulation is repeated 16 times with different random seeds, as motivated in Section 2.3), we ran pairs of experiments in parallel, each one using 16 cores, so that each individual evaluation was parallelized. The values for COST and DDR obtained from the 16 simulation repetitions were then averaged and passed to the MOEA.

Furthermore, we assessed the repeatability of results by rerunning each experiment configuration (not reported here

[1] It should be noted that such a complex individual structure (characterized by predefined discrete sets for each allele, each one with a user-defined occurrence probability) makes difficult the application of general-purpose state-of-the-art MOEAs, such as NSGA-II [12] or its recent variant NSGA-III [11, 15]. Indeed, although it would be possible to implement such individual structure within NSGA-II/III, these algorithms do not provide natively this kind of flexible encoding, that is crucial to this problem. On the other hand, μGP offers off-the-shelf various encoding schemes, thus requiring a much smaller coding effort on the algorithm implementation, and allowing users to focus on applications rather than algorithmic details.

Table 5: Configuration of the experiments considered in this study.

Fitness functions maximized (↑) or minimized (↓)	Noise model	Network size	Model for strong links: discrete set of gains (dBm)	% of strong links	No of generations	No of evaluations	Runtime
COST ↑ – DDR ↑	light	20	{0}	50%	1000	5123	∼ 18h
COST ↓ – DDR ↓	light	20	{0}	50%	1000	5164	∼ 18h
COST ↑ – DDR ↑	heavy	20	{-100, -80, -60, -40, -20, 0}	50%	1000	6652	∼ 33h
COST ↓ – DDR ↓	heavy	20	{-100, -80, -60, -40, -20, 0}	50%	1000	6551	∼ 20h

due to lack of space) and observing the statistical equivalence of the Pareto fronts obtained on repeated experiments.

With regard to the runtime of the experiments, it should be noted that the runtime of each experiment depends on (a) the signal gain model for strong links, (b) the noise model, (c) the number of topologies thus evaluated in an experiment, and, significantly, (d) it is higher for topologies with high final COST and DDR values, as these are the cases in which the network will generate large numbers of logged events. Hence, COST ↓ – DDR ↓ experiments show shorter runtimes than COST ↑ – DDR ↑, due to the lower number of packets generated by each simulation. On the other hand, experiments with heavy injected noise have longer runtimes than the corresponding noiseless ones, because of the additional overhead introduced by the noise generation. This can be seen also at individual level, where the duration of each single simulation ranges from approximately 18 seconds (heavy noise, COST ↑ – DDR ↑) to $11 - 12$ seconds (remaining cases). Obviously, the individual runtime scales up with the network size: however, since in this study we focused only on 20-nodes networks, we were not interested in studying simulation runtime scalability.

Another important aspect that should be noted is that, in the experiments with heavy injected noise, a higher number of topologies was evaluated. This is due to the stochastic activation of the genetic operators from Table 4 during each generation: indeed, while the number of operators activated per generation is fixed, the type of operators is variable, so that a higher number of crossover activations will generate a higher number of individuals. Moreover, due to the self-adaptation scheme embedded within μGP, the activation probability changes across generations. The MOEA tends to use the crossover operators more intensively in the noisy experiments than the corresponding noiseless experiments, thus generating more individuals. This can be easily explained since the fitness landscape is likely more rugged in the presence of noise, and the fine-tuning caused by mutations may be less effective.

4.2 Pareto fronts for collection routing

The Pareto fronts obtained, for each experimental configuration (after various numbers of generations, including the final count of 1000 generations) are reported in Figures 4 and 5. To be able to answer the question whether a stopping condition at 1000 generations is sufficient to find Pareto fronts, we illustrate not only the final front, but also the first, randomly selected generation, and also the Pareto fronts found after every 100th generation in the experiment.

The figures also show the combined Pareto fronts obtained in the COST ↑ – DDR ↑ and COST ↓ – DDR ↓ experiments for both light (Fig. 4) and heavy injected noise (Fig. 5), so as to give a visual depiction of how all tradeoff topologies are spread over the two search spaces, and thus delineate the search spaces features. It can be seen, for example, that noisy links have a stronger influence on the upper bounds of COST and DDR, while the lower-bound Pareto front for COST and DDR is more robust against noise.

4.3 Practical implications of this study

The importance of our results for WSN protocol designers is manifold. By finding bounds to the two-dimensional performance of CTP, we are able to prove that – surprisingly – this protocol for collection routing has strongly unintuitive performance trends, which may be summarized and used by practitioners as follows:

1. COST and DDR will, in general, *correlate negatively* for CTP.

2. The upper bound of the energy COST in WSN situations with *low* DDR is one order of magnitude *higher* (rather than lower, as expected) than that of WSN situations with near-perfect data delivery ratios. This statement is true regardless of the amplitude of the injected noise.

3. WSN situations which allow for high DDR can only exhibit few COST values from a small interval. On the other hand, WSN situations where DDR is low can exhibit a staggering amount of COST variation.

These results can be an important tool for WSN practitioners: by having this quantitative *model* of the performance bounds for the protocol across the entire space of possible topologies and communication conditions, practitioners can:

- Predict the worst possible performance for a completely uncontrolled WSN testbed; not only can DDR fall to zero, but also the total energy COST of the testbed can rise to an average of 30 radio events per node per second, a number on the basis of which a worst-case lifetime can be predicted for the network.

- Gain visibility into the effect of a network problem in the field. E.g., a practitioner who sees a current global DDR over 95% calculated at the sink can be certain that the current energy COST of the network is under a relatively low average of 7.5 radio events per node per second; on the other hand, if the DDR at the sink is close to zero, there is a possibility that the total COST has currently risen to the upper bound of 30 radio events per node per second, which may require corrective intervention over the testbed.

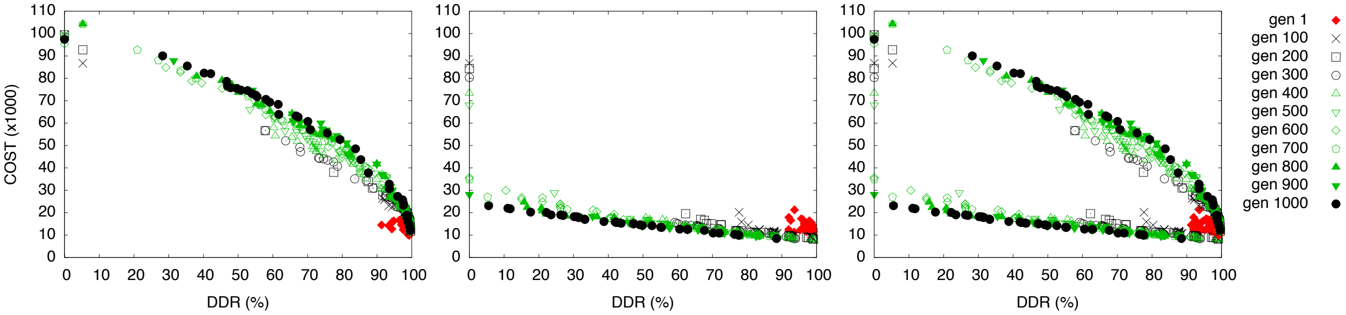

Figure 4: Pareto fronts obtained for CTP with light noise, for the configurations COST ↑ – DDR ↑ (left), COST ↓ – DDR ↓ (center), and the combination of the two fronts (right).

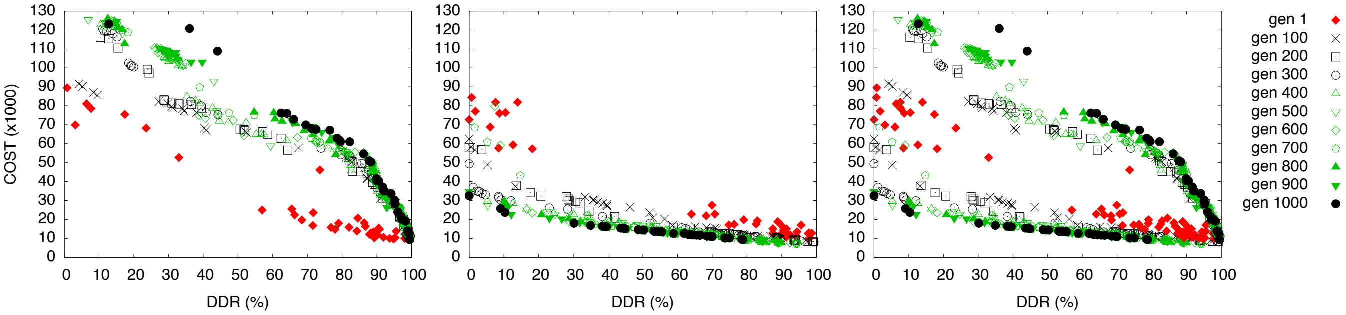

Figure 5: Pareto fronts obtained for CTP with heavy noise, for the configurations COST ↑ – DDR ↑ (left), COST ↓ – DDR ↓ (center), and the combination of the two fronts (right).

Furthermore, the experiments yielded a large number of WSN situations which exhibit interesting performance; a detailed analysis of topologies was partially done in [6] (only for topologies of high COST), partly left as future work. In Fig. 6, we show two such top individuals; dashed links are links which are strong in only one direction.

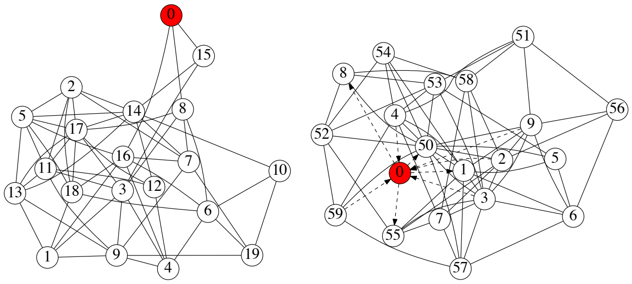

Figure 6: (Left) topology with DDR=0.99%, COST=12000; (right) topology with DDR=0%, COST=97000; both with light noise.

The WSN topology in Fig. 6 (left) (shown as an undirected topology, i.e., disregarding, for readability, all links which are not strong in both directions) is an extremely well-behaved topology, with near-perfect values for both DDR and COST under noiseless conditions. This shows that bidirectional connectivity of the topology graph can be crucial to good performance. On the other hand, the topology of the top individual in Fig. 6 (right) is such that the topology is still connected, yet partially through directional links. While the protocol is still expected to function using these directional links, even if no node is bidirectionally connected to the sink node, the protocol delivers an unexpected DDR value of zero, and also causes a one order of magnitude increase in the COST. This energy is spent by the protocol in an effort to, unsuccessfully, build the routing tree, even in the presence of only light noise. The low DDR demonstrates a vulnerability of the CTP design or configuration, complicated further by the adaptive beacon intervals, which also add an excessive amount of network traffic to the problem.

5. CONCLUSIONS

In this paper, we presented recent developments on an evolutionary methodology to analyze routing protocols in WSNs. The underlying idea is to use an evolutionary core to generate individuals, i.e. candidate WSN topologies, showing unexpected network performances. We analyzed the behavior of a target protocol, CTP, on the generated topologies evaluating two metrics that relate with undesired protocol behaviors: data delivery ratio, and total energy cost. With the proposed approach, we were able to explore in feasible time the highly multidimensional network state space, uncovering full sets of Pareto-efficient topologies characterized by tradeoff values of the two metrics. Essential to the success of the approach was the use of a MOEA toolkit able to handle the complex description of the individuals.

Future works will focus on performing additional experiments with more configurations and on other protocols, in order to further explore the capabilities of the proposed evolutionary framework. From an algorithmic point of view, we are also interested in extending the capabilities of μGP, enriching it with more sophisticated multi-objective search operators and archiving methods.

6. ACKNOWLEDGMENTS

INCAS^3 is co-funded by the Province of Drenthe, the Municipality of Assen, the European Fund for Regional Development and the Ministry of Economic Affairs, Peaks in the Delta.

7. REFERENCES

[1] M. Baldi, F. Corno, M. Rebaudengo, and G. Squillero. GA-based performance analysis of network protocols. In *Proc. Ninth IEEE International Conference on Tools with Artificial Intelligence, 1997*, pages 118–124, 1997.

[2] I. Bate and M. Fairbairn. Searching for the minimum failures that can cause a hazard in a wireless sensor network. In *Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference*, GECCO '13, pages 1213–1220, New York, NY, USA, 2013. ACM.

[3] S. Begum, A. Helmy, and S. Gupta. Modeling and test generation for worst-case performance evaluation of mac protocols for wireless ad hoc networks. In *IEEE International Symposium on Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 1–10, 2009.

[4] A. Boukerche. Performance evaluation of routing protocols for ad hoc wireless networks. *Mob. Netw. Appl.*, 9(4):333–342, Aug. 2004.

[5] D. Bucur, G. Iacca, G. Squillero, and A. Tonda. An evolutionary framework for routing protocol analysis in wireless sensor networks. In A. I. Esparcia-Alcázar, editor, *Applications of Evolutionary Computation*, volume 7835 of *Lecture Notes in Computer Science*, pages 1–11. Springer Berlin Heidelberg, 2013.

[6] D. Bucur, G. Iacca, G. Squillero, and A. Tonda. The impact of topology on energy consumption for collection tree protocols: An experimental assessment through evolutionary computation. *Applied Soft Computing*, 16(0):210–222, 2014.

[7] O. Chipara, C. Lu, T. C. Bailey, and G.-C. Roman. Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 155–168, New York, NY, USA, 2010. ACM.

[8] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuisen. *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.

[9] P. Corke, T. Wark, R. Jurdak, W. Hu, P. Valencia, and D. Moore. Environmental wireless sensor networks. *Proceedings of the IEEE*, 98(11):1903 –1917, nov. 2010.

[10] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 2001.

[11] K. Deb and H. Jain. An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part I: Solving Problems with Box Constraints. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1, 2013.

[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[13] S. Gandini, W. Ruzzarin, E. Sanchez, G. Squillero, and A. Tonda. A framework for automated detection of power-related software errors in industrial verification processes. *Journal of Electronic Testing*, 26(6):689–697, 2010.

[14] O. Gnawali, R. Fonseca, K. Jamieson, M. Kazandjieva, D. Moss, and P. Levis. CTP: An Efficient, Robust, and Reliable Collection Tree Protocol for Wireless Sensor Networks. *ACM Transactions on Sensor Networks (TOSN)*, 10(3), 2014.

[15] H. Jain and K. Deb. An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1, 2013.

[16] J. G. Ko, T. Gao, and A. Terzis. Empirical study of a medical sensor application in an urban emergency department. In *Proceedings of the Fourth International Conference on Body Area Networks*, BodyNets '09, pages 10:1–10:8, ICST, Brussels, Belgium, Belgium, 2009.

[17] K. Langendoen. Apples, oranges, and testbeds. In *In Proc. IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS '06)*, pages 387–396, 2006.

[18] H. J. Lee, A. Cerpa, and P. Levis. Improving wireless simulation through noise modeling. In *Proceedings of the 6th international conference on Information processing in sensor networks*, IPSN '07, pages 21–30, New York, NY, USA, 2007. ACM.

[19] P. Levis, D. Gay, V. Handziski, J.-H. Hauer, B. Greenstein, M. Turon, J. Hui, K. Klues, C. Sharp, R. Szewczyk, J. Polastre, P. Buonadonna, L. Nachman, G. Tolle, D. Culler, and A. Wolisz. T2: A second generation OS for embedded sensor networks. Technical Report TKN-05-007, Technische Universität Berlin, 2005.

[20] P. Levis, N. Lee, M. Welsh, and D. E. Culler. TOSSIM: Accurate and scalable simulation of entire TinyOS applications. In *Proceedings of the ACM conference on Embedded Networked Sensor Systems (SenSys)*, pages 126–137, 2003.

[21] L. Mottola, T. Voigt, F. Österlind, J. Eriksson, L. Baresi, and C. Ghezzi. Anquiro: Enabling efficient static verification of sensor network software. In *Workshop on Software Engineering for Sensor Network Applications (SESENA) ICSE(2)*, 2010.

[22] D. Puccinelli, O. Gnawali, S. Yoon, S. Santini, U. Colesanti, S. Giordano, and L. Guibas. The impact of network topology on collection performance. In *Proc. 8th European Conference on Wireless Sensor Networks (EWSN)*, pages 17–32, 2011.

[23] G. Sacco, K. Barltrop, C. Lee, G. Horvath, R. Terrile, and S. Lee. Application of genetic algorithm for flight system verification and validation. In *Aerospace conference*, pages 1–7. IEEE, 2009.

[24] E. Sanchez, M. Schillaci, and G. Squillero. *Evolutionary Optimization: the μGP toolkit*. Springer Publishing Company, Incorporated, 1st edition, 2011.

[25] W. Shyang, C. Lakos, Z. Michalewicz, and S. Schellenberg. Experiments in applying evolutionary algorithms to software verification. In *IEEE World Congress on Computational Intelligence (CEC)*, pages 3531–3536. IEEE, 2008.

[26] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh. Fidelity and yield in a volcano monitoring sensor network. In *Proceedings of the 7th symposium on Operating systems design and implementation*, OSDI '06, pages 381–396. USENIX Association, 2006.

[27] M. Zheng, J. Sun, Y. Liu, J. Dong, and Y. Gu. Towards a model checker for nesc and wireless sensor networks. In S. Qin and Z. Qiu, editors, *Formal Methods and Software Engineering*, volume 6991 of *Lecture Notes in Computer Science*, pages 372–387. Springer Berlin Heidelberg, 2011.