Evolutionary and Principled Search Strategies for Sensornet Protocol Optimization

Jonathan Tate, *Member, IEEE*, Benjamin Woolford-Lim, *Student Member, IEEE*,
Iain Bate, and Xin Yao, *Fellow, IEEE*

***Abstract*—Interactions between multiple tunable protocol parameters and multiple performance metrics are generally complex and unknown; finding optimal solutions is generally difficult. However, protocol tuning can yield significant gains in energy efficiency and resource requirements, which is of particular importance for sensornet systems in which resource availability is severely restricted. We address this multi-objective optimization problem for two dissimilar routing protocols and by two distinct approaches. First, we apply factorial design and statistical model fitting methods to reject insignificant factors and locate regions of the problem space containing near-optimal solutions by principled search. Second, we apply the Strength Pareto Evolutionary Algorithm 2 and Two-Archive evolutionary algorithms to explore the problem space, with each iteration potentially yielding solutions of higher quality and diversity than the preceding iteration. Whereas a principled search methodology yields a generally applicable survey of the problem space and enables performance prediction, the evolutionary approach yields viable solutions of higher quality and at lower experimental cost. This is the first study in which sensornet protocol optimization has been explicitly formulated as a multi-objective problem and solved with state-of-the-art multi-objective evolutionary algorithms.**

***Index Terms*—Evolutionary Algorithms (EAs), experiment design, multi-objective optimization, protocols, sensornets.**

I. Introduction

SENSORNETS compose many autonomous *motes* into ad-hoc networks for distributed sensing and processing applications. Motes are small, cheap computers equipped with independent power supplies, wireless communication capability, and sensors with which to passively monitor their environment. Sensor-actuator networks also interface some motes with actuators to actively influence the environment, completing the control feedback loop. Interaction with the physical environment implies that sensornets have real-time requirements in addition to functional requirements. Typical applications include environmental monitoring or surveillance.

An important feature of sensornets is that motes have highly restricted resource availability, owing to the requirements for small physical size and low unit cost. Limited availability of energy resources are particularly significant. When a mote runs out of energy it cannot participate in the sensornet. Eventually, a state is reached in which there are insufficient active motes for the distributed application to function correctly; at this stage, the entire sensornet is effectively dead [2]. Sensornet designs must therefore be sufficiently optimized to ensure correct operation for at least the specified operational lifetime.

Wireless communication is generally the most energy-hungry aspect of sensornet operation [3], and as such represents an obvious target for optimization. Energy efficiency improvements might be achieved by improvements in software or hardware, but the latter is often impractical; sensornet designers typically select standard COTS hardware which cannot be changed or improved *post hoc* in the field [4]. In this paper, we consider only software-driven improvements at the level of network middleware, assuming a fixed hardware platform and typical distributed sensing application. Our results are therefore portable across hardware platforms and high-level software applications in typical sensornet scenarios.

The application of search-based protocol tuning methods offers the sensornet designer an excellent tool with which to optimize network performance for a given usage context. It is surprising, therefore, that this approach has received such little attention in the literature to date. Possible explanations include the large number of controllable factors, measurable responses, and test cases required for meaningful coverage. We address these problems in this paper.

We demonstrate that careful tuning of networking protocols can yield substantial improvements in energy efficiency without compromising performance, elaborating a mechanism with which to address this multi-objective optimization problem. We compare two fundamentally different approaches; a broad-but-shallow design of experiments (DOE) approach based on factorial design (FD), and a narrow-but-deep evolutionary algorithm (EA) approach employing the *Strength Pareto Evolutionary Algorithm 2* (SPEA2) and Two-Archive (TA) algorithms. We address the relationship between analytical cost and solution quality, demonstrating that a more sophisticated experimental technique can yield high quality solutions at greatly reduced overhead. To the best of our knowledge, our work is the first to apply EA techniques to the sensornet protocol optimization problem.

To demonstrate the versatility and generality of our proposed method we consider its application to the optimization of two

Manuscript received December 30, 2009; revised May 20, 2010, September 27, 2010, and January 24, 2011; accepted June 23, 2011. A preliminary and much shorter version of this paper was published at CEC'09 [1]. This work was supported in part by EPSRC Grant EP/D052785/1 on "SEBASE: Software Engineering By Automated SEarch," and BAE Systems plc by the Grant "Hierarchical System Management for Integrated Modular Systems." This paper was recommended by Associate Editor E. Santos, Jr.

J. Tate and I. Bate are with the Department of Computer Science, University of York, YO10 5DD York, U.K. (e-mail: jt@cs.york.ac.uk; iain.bate@cs.york.ac.uk).

B. Woolford-Lim and X. Yao are with the School of Computer Science, University of Birmingham, B15 2TT Birmingham, U.K. (e-mail: b.j.woolford-lim@cs.bham.ac.uk; x.yao@cs.bham.ac.uk).

Digital Object Identifier 10.1109/TSMCB.2011.2161466

1083-4419/$26.00 © 2011 IEEE

different network routing protocols; the *TTL-bounded gossip* (TBG) protocol [5], and the *implicit geographic forwarding* (IGF) protocol [6]. We provide a detailed comparison of single- and multi-objective optimization strategies. We compare a DOE approach based on FD [7] to two different EAs; *SPEA2* [8] and *TA* [9]. These EAs were selected as they perform well in optimization problems with many objectives. Our method can be implemented using other EAs and can be applied to other protocol optimization problems.

In this paper, we describe protocol tuning techniques and do not seek to define or impose any specific system design process or philosophy; it is for the system designer to select an appropriate development process. In general, these techniques might be implemented to obtain near-optimal parametric values before sensornet deployment. The techniques could also be applied at a later stage, when a sensornet has already been deployed, so as to improve performance in the field. For example, a sensornet might be deployed hastily, with suboptimal configuration, in response to an emergency situation. While the sensornet collects valuable data in the field, the designer can tune and optimize the configuration in the laboratory, pushing out the optimized network configuration at a later time.

This work addresses interesting challenges. The sensornet protocol tuning problem is not simple or idealized; it is a real-world problem with multiple inputs, multiple outputs, and multiple objectives. Complex interrelationships between factors were unknown *a priori* and could not be targeted during experiment design. Fitness function evaluation incurred an unusually high cost penalty; each evaluation implied execution of a substantial simulation instance. Furthermore, our analysis accounts for noise present in the experimental data.

This article addresses the following research objectives:

Obj 1: Obtain near-optimal solutions to the sensornet protocol tuning problem using evolutionary and principled search approaches.

Obj 2: Compare solution quality attainable by evolutionary and principled search approaches.

Obj 3: Consider the relationship between solution quality and experimental overhead for the optimization problem.

The remainder of the paper is organized as follows. Section II sets the content of this paper in the context of related work. Section III defines the protocol optimization problem and the experiment design with which it is to be addressed. Section IV describes the FD principled search approach. Sections V and VI describe the SPEA2 and TA evolutionary search approaches, respectively. Section VII presents experimental results from which conclusions are drawn in Section VIII.

II. Related Work

A. Case for Protocol Tuning

Sensornets are similar to *Mobile Ad-hoc Networks* (MANETs) in that they are composed of small, low-resource computer nodes connected through a wireless network. Akyildiz *et al.* [10] consider MANETs to be *the closest peers to sensor networks*, but note that MANETs place far greater importance on mobility, and MANETs may be composed of far fewer nodes. More significantly, sensornets are *data-centric* rather than *application-centric* [11] and are usually composed of nonmobile nodes with environmental sensors [3].

Many sensornet communication protocols are described in the literature, many of which accept a range of general and/or protocol-specific parameters to fine-tune performance in particular cases [12]. It appears that rather more research effort has been devoted to creating new protocols than optimizing the performance of existing protocols. Although these new protocols are valuable contributions to the field, a sensornet designer must balance the demands of multiple performance objectives, and as such it is rarely appropriate to create a new protocol to address a single factor in isolation [13].

Even small improvements in network efficiency can yield large gains in network lifetime. Sending a single bit of information 100 m may consume more energy than executing 3000 CPU instructions [14]. Surprisingly, network radio hardware operating in the idle listening mode offers little power advantage over actively sending or receiving, and receiving data often consumes more energy than transmitting [15]. These costs are incurred by the sender, the receiver, and any intermediate nodes along multihop paths, which may grow as network node count increases.

To maximize the overall effectiveness and efficiency of complex systems, it is insufficient to consider each influential aspect in isolation. Owing to the nonlinear relationship between distance and received signal power in wireless communications [3], the total energy consumed in delivering a given packet is sometimes lower if the route contains many short hops rather than few long hops, despite the higher number of participating nodes and transmissions [16].

The POCSAG protocol [17] defines the minimal set of periods during which devices must listen for broadcasts of relevant messages on a shared channel, the inverse of which being the much larger set of periods during which a device can switch off to conserve energy reserves. This implicitly defines a nonzero lower bound on message latency, typically orders of magnitude greater than actual message transmission times. Designers are forced to define these tradeoffs and design compromises if a minimum QoS is to be guaranteed [18].

Sensornet designers must identify the most significant factors to avoid being swamped by unnecessary detail. Unfortunately, even identifying the relative importance of factors and their interactions is rarely trivial [19]. Discovering the best values to assign to these factors and understanding their impact on network behavior tradeoffs is harder still [20]. Tunable parameters are often defined without clear default values and may be defined over an infinite range.

B. Approaches to Protocol Tuning

When exploring the behavior of a system, it is possible to apply either *analytical techniques* to a formal model of the system, or *simulative analysis* in which experiments are performed with a simulation model of the system, or some combination of the two. Ploennigs *et al.* [21] support the latter hybrid approach in which a formal model of the system is defined for *analytical techniques* and from which simulation models are automatically generated for *simulative analysis*. However, this assumes that a reasonable formal model of the system can be constructed. There is some debate as to whether this is [22] or is not [23] feasible for typical sensornets.

Mohan *et al.* [24] observe that little work exists on evaluating and maximizing end-to-end performance of large, dense sensornets. A number of candidate protocol stacks were measured for a fixed configuration of sensornet and application. Rather than considering several tunings of a single protocol, this experimental work considered a number of protocols, but only one configuration of each. A significant diversity of network performance was observed, though it was not always possible to determine why a given protocol behaved as it did.

With many controlled factors and measured responses, it is generally difficult to understand the resulting complex interrelationships. Totaro and Perkins [25] apply a *systematic statistical DOE* approach to evaluate and model the complex tradeoffs in designing MANETs. This work considers the impact of controlling network composition with a fixed network application and environment. In contrast, we assume a fixed network design and a distributed application built on a tunable networking infrastructure. We assume no prior knowledge of physical topology, which strongly influences energy consumption [26].

Alba *et al.* [27] describe a process to optimize MANET broadcast strategies. Five tunable parameters were defined which characterize the search space, and three metrics of network performance were defined which characterize the solution space. Two problem formulations were considered; the first optimized against three objectives, and the second optimized against two objectives with a user-supplied constraint on acceptable solutions. A cellular multi-objective genetic algorithm called *cMOGA* was employed to generate a Pareto front of good candidate solutions from which a human designer selects. Although the technique is generally effective, the Pareto front is *very reduced* in the presence of constraints, and there is little guidance for designers on manually selecting the single best solution. This limits the extent to which the technique is useful for the problem addressed in this paper.

C. Multi-Objective Optimization

A significant issue with multi-objective problems is the difficulty in determining a well-defined ordering of solution quality for a given set of candidate solutions [28]. In single-objective problems, the ordering of solution quality is given simply by the ordering of the associated fitness values. However, with multi-objective problems, there are multiple fitness values to consider. A solution with every fitness value lower than another is clearly superior and is said to *dominate* the other solution. However, solutions may be lower in one fitness value but higher in another; this kind of solution is termed *nondominated* and is harder to classify. In place of a single optimum value, multi-objective problems generally have a *Pareto-optimal front* along which all solutions are mutually *nondominated*, and all other valid solutions are dominated by the Pareto-optimal front members.

Combinatorial explosion of possible solutions within a multivariable problem renders exhaustive search impossible. Stochastic search algorithms explore the solution space nonexhaustively in reasonable time. Multi-objective evolutionary algorithms (MOEAs) are stochastic generational multi-objective search algorithms. Using the *survival of the fittest* concept, they seek Pareto-optimal fronts by evolving progressively better solutions based on the relative *fitness* of previous solutions. Numerous algorithms of this type have been proposed. Details of the state of the art can be found in a book by Coello *et al.* [28] or a recent survey paper by Guliashki *et al.* [29].

D. Applying Evolutionary Computation in Sensornet Design

Jourdan and de Weck [30] were the first to apply MOEAs to a sensornet optimization problem, in which the optimal physical location of a set of motes was determined. Two objectives were defined, *maximal coverage* and *maximal lifetime*, with no fixed constraints. Although these early results were encouraging, they were derived from unrealistically small networks of just ten nodes.

Quintão *et al.* [31] addressed the optimization of dynamically controlled sensornet topologies with the competing objectives of *minimal energy cost* and *maximal network coverage*. Two approaches were compared; a linear programming formulation which yielded exact solutions with high computation overhead and an evolutionary approach which yielded good solutions in acceptable time. Greedy EA approaches were effective in finding local minima but were often ineffective in looking beyond these local minima to better solutions elsewhere in the parameter landscape. We address this problem in our work with a two-phase approach in which an initial principled search of the problem space is used to direct the EA to regions likely to contain global rather than local minima.

Sensornet node populations can be divided into disjoint subsets, called *sensor covers*, each providing full coverage of all sensor targets. Duty cycling at the level of complete sensor covers reduces energy consumption and increases network lifetime. The *disjoint set covers* problem finds the maximal number of sensor covers but is NP-complete; most heuristic approaches produce low quality solutions or require exponential time. Lai *et al.* [32] propose the Genetic Algorithm for Maximum Disjoint Set Covers (*GAMDSC*) genetic algorithm to find near-optimal solutions in polynomial time.

The *disjoint set covers* problem is also addressed by Hu *et al.* [33], who propose the *schedule transition hybrid genetic algorithm*. This is a hybrid approach combining a genetic algorithm with schedule transition operations, employing a forward encoding scheme to optimize both the sensor cover and duty scheduling aspects simultaneously.

Molina *et al.* [34] address the sensornet layout optimization problem, balancing the two competing objectives of *minimal energy cost* and *maximal network lifetime* with a constraint of *complete network coverage*. Two MOEAs are employed for this problem; Nondominated Sorting Genetic Algorithm II (NSGA-II) [35] and Indicator-Based Evolutionary Algorithm (IBEA) [36]. Each MOEA found sets of nondominated feasible solutions that were found to be efficient, but no statistically significant difference was observed between the algorithms.

Chaudhuri and Dasgupta [37] compared two MOEAs, NSGA-II [35] and SPEA2 [8], in maximizing the competing demands of *coverage* and *lifetime* in sensornet design. NSGA-II tended to find *better, feasible, acceptable, balanced solutions*, whereas SPEA2 tended to find *extreme solutions*. It may therefore be useful to split optimization resources between multiple MOEAs; some reliably produce reasonable solutions but riskier alternatives may find even better solutions.

Seo *et al.* [38] propose the *location-aware two-dimensional genetic algorithm* to control the number and size of node clusters, and the distance between nodes, in sensornet deployments. This algorithm uses 2-D chromosomes, unlike the 1-D chromosomes used by most GAs, to encode more naturally the 2-D geographic distribution of nodes.

Other studies demonstrate the application of MOEAs to optimization problems in sensornet design unrelated to those considered in this paper. Yang *et al.* [39] applied the NSGA-II algorithm sought to find the optimal configuration of adaptive antennas. Brandolese and Rucco [40] seek to optimize application-level dynamic linking and management of cooperating functions in order to maximize network lifetime.

III. Experiment Design

In this section, we define the experimental configuration used in Sections IV–VI to tune a sensornet protocol and explore the parameter landscape. We define the structure and content of the simulated network, the controlled factors, the solution quality metrics, and the simulation environment.

A. Search-Driven Protocol Optimization

Two experimental approaches are employed in this paper. A principled search approach, using FD, is used to sample the entire problem space in a systematic and even manner [7]. This is a broad-but-shallow deterministic search method. An evolutionary approach is then employed, using the SPEA2 [8] and TA [9] algorithms to sample the problem space in a guided and uneven manner. This is a narrow-but-deep stochastic search method.

To minimize experimental cost and maximize solution quality for the EA, we could obtain a survey of the problem space, executing a low-resolution version of the FD approach to locate regions of the problem space in which some high quality solutions reside. We would then apply this insight to focus the EAs, seeding the working sets with values in these regions but allowing the EAs to explore the entire problem space. However, the results presented in Section VII demonstrate that this is not necessary for the MOEAs considered in this paper.

Now, consider other protocol tuning methods. Perhaps, the simplest approach, from a computation viewpoint, is for the sensornet designer to obtain a protocol tuning by a *manual approach*. Applying intuition and personal experience allows a candidate solution to be obtained with fixed cost. Although this method is O(1) in all problem aspects, there is no guarantee that the result is near-optimal, and there is no guarantee that the result will conform to any specific requirement or constraint.

A similarly flawed approach is to ignore the issue and simply accept a default set of parameters. First, this assumes the existence of default values; this assumption is incorrect for the TBG and IGF protocols considered in this paper, for which no such default values are defined. Second, if default values were available, there is again no guarantee that these would be near-optimal for any specific problem, nor that that the result will conform to any specific requirement or constraint.

A simple extension is *trial-and-improvement*, in which the sensornet designer iteratively changes the value of one or more controlled factors and then measures the impact experimentally. This nonprincipled search method has $\mathrm{O}(n)$ cost in the number of iterations, but there is no guarantee that any individual iteration will yield a solution of equal or better quality to that which precedes it. There is also no defined mechanism with which to decide which controlled factor to change, and to which value, at any given iteration.

A *random search* approach simply defines n candidate protocol configurations by randomly selecting values for controlled factors within defined bounds, assessing the performance of each by experiment, and then selecting that which offers the highest quality. This approach is trivially $\mathrm{O}(n)$ in the number of randomly defined configurations. Randomly sampling the search space at n points is useful in seeding the SPEA2 and TA evolutionary search methods discussed in Sections V and VI, respectively.

Random search may compare favorably to evolutionary techniques when the problem to be optimized has many objectives [41]. However, when applied in isolation, random search offers no guarantee of even coverage of the problem space and offers no guarantees of solution quality. Attempting to force even coverage of the search space results in a method similar to FD, as discussed in Section IV.

B. Experimental Technique

To assess the quality of a given candidate protocol tuning, we measure its performance in a representative sensornet configuration. It would be impractical to employ real networks of meaningful size in quality assessment; the overheads of time, logistics, and financial cost are too high. It is also impossible to guarantee a consistent and unchanging physical environment for the total runtime of the tens of thousands of experiments implied by the search process. This would severely undermine the validity and repeatability of the measurements taken from the sensornet and which are critical to our analysis.

Our evaluation of the fitness function for candidate protocol configurations was obtained from a simulation model of the sensornet system. This is in contrast to formal mathematical modeling approaches. The construction and analysis of formal models would allow the evaluation and verification of required network properties. Given a suitable formal model, analytical methods such as mixed integer programming could be applied to optimize for one or more factors of interest.

However, formal modeling of sensornets is currently an open research problem with numerous inherent difficulties [23]. Although we strongly support current efforts to produce usable formal models of sensornet systems [42], [43], the current state of the art has not yet reached the point where this is feasible for systems of the scale considered in this paper. There also remains the risk that the formal model omits some significant detail about the modeled problem, which would undermine any results obtained by this approach [44].

We, therefore, conduct all experiments by simulation using the yet another sensornet simulator (YASS) sensornet simulator [45], which is optimized and validated for experiments of this type and their corresponding duty patterns. This allows sufficient simulation experiments, to obtain an acceptable sampling of the search space, to run to completion in acceptable time. Other similar sensornet simulators could be employed with comparable results.

When assessing the time-dependent behavior of a given simulated system, we must consider the difference between simulated time, τ, and wall time, t. All measurements, behaviors, and decisions, within the simulated systems considered in this paper are defined in terms of τ.

C. Network Configuration

The techniques outlined in this paper are independent of the specific protocols and network designs explored in the following experiments. However, these experiments explore only a finite portion of the unbounded design space of all networks and all protocols. It is likely that the trends we identify in network performance responses as a function of protocol tuning parameters are likely to apply in similar networking contexts. Nevertheless, we limit the scope of our claims to the portion of design space defined in this section, within which we have confidence in our findings as they are demonstrated to have statistical significance.

A set of three typical sensornets was defined and reused for all experiments. Each sensornet consisted of 500 static motes of identical capability modeled on the Crossbow MICA2 mote. Motes were distributed randomly within a square of side length 21 km yielding a geographic distribution of uniform planar density without network voids. The resulting average *degree of connectivity*, the number of peer nodes with which a given node can feasibly communicate, is approximately 20 which is typical of sensornets [6].

All internodal communication was defined to occur through anisotropic radio broadcast in an obstacle-free vacuum. Signal propagation and attenuation was modeled using the Friis free space model [46] with exponent of 2.0.

The simulated motes run a distributed sensing application in which every node periodically produces a small data packet. In our experiments, we specify exactly one destination per packet. The destination of each packet is randomly selected from all motes in the network to prevent bias from any implicit structure in the mote distribution.

Each node in the network can act as a packet source, a packet destination, or a packet relay. Section III-D describes the protocols that regulate and control these behaviors.

D. Protocols and Their Controlled Factors

In this paper, we consider two protocols designed for MANETs, both of which implement a low-state lazy binding approach. The TBG protocol [5] is described in Section III-D3, and the IGF protocol [6] is described in Section III-D4. These protocols were chosen for their simplicity. More complex protocols often incorporate simple protocols during early discovery phases or to maintain information. However, if implemented carelessly, these simple protocols can be highly wasteful, and hence offer an excellent opportunity for energy efficiency improvement. For example, unbounded flooded messages can easily cover the entire network [47] which is wasteful if the source and destination are physically close.

Note that in selecting these protocols, we make no claims as to their merit for any given sensornet application. More specifically, we do not claim that when optimally configured, they necessarily offer superior performance to alternative protocols. However, we see no reason that the methods described in this paper could not be applied to these other protocols.

1) Defining the Search Space: It is rare for network protocol definitions to include default values for configurable parameters, as values appropriate for any given network may not be appropriate in others [12]. This is particularly true of sensornets for which performance is highly sensitive to the operating environment. If protocol definitions do not provide default values, there is no default configuration to act as a baseline for comparison. We must therefore compare relative solution quality across the defined search space.

Where a parameter is defined over a large or infinite range, it may be necessary to restrict the search effort to a range of values of acceptable size. The issue of defining parameter search ranges is part of the problem definition, rather than part of the problem solution, and hence does not fall within the scope of this paper. We used search ranges published elsewhere [20] as these were found to produce a search space in which some configurations performed better than others; otherwise, optimization would be unnecessary.

2) Protocol-Independent Controlled Factors: Some tunable parameters are specific to a given protocol, but others are common to several protocols. In this section, we define controlled factors $X_1 - X_5$ which are common to both *TBG* and *IGF*, and may interact with other shared parameters and protocol-specific parameters. We define our experiments to explore as much of the parameter space as is possible. For each parameter $X_1 - X_5$, we limit our search to a subset of the defined range within which a measurable difference in response is known to exist [20].

X_1: **Seen packet buffer size.** The number of packets received or transmitted by a node of which knowledge is retained. Nodes do not retransmit a previously transmitted packet if the latter is held in this cache. New packets displace a randomly selected cached packet if the buffer is full. Measured in *packets*. Defined for integral values in the range $[0, \infty)$. Search range is [1, 10].

X_2: **Waiting packet buffer size.** The number of packets which can be simultaneously enqueued for transmission or retransmission. Packets are consumed from the queue head and added to the queue tail. If the queue is full when a new packet is added, a randomly selected enqueued packet is dropped. Measured in *packets*. Defined for integral values in the range $[1, \infty)$. Search range is [1, 10].

X_3: **Initial backoff.** Before beginning transmission of a packet, the sending node will sense the wireless medium. If the medium is clear, transmission begins immediately, otherwise an exponential backoff strategy is applied in which the nth term is the nth power of this base value. Measured in *seconds*. Defined in the range $(0, \infty)$. Search range is [0.1, 1].

X_4: **Packet lifetime.** The maximum permitted time for a packet to remain in transit. If the lifetime is exceeded before reaching the destination, the packet is dropped. Measured in *seconds*. Defined in the range $(0, \infty)$. Search range is [0.1,10].

X_5: **TTL.** The total number of node-node hops permitted for packets traversing the network. If this TTL is exceeded prior to reaching the destination, the packet is dropped. Measured in *hops*. Defined for integral values in the range $[1, \infty)$. Search range is [1, 10].

Other networking protocols may be influenced by a different set of factors, which may or may not intersect the above set. However, any protocol for which there are quantitatively defined controlled factors can be explored using this process.

3) TTL-Bounded Gossip Protocol: An adapted form [20] of the TBG protocol [5] is the first protocol under consideration. This protocol is ignorant of energy, network topology, and the host application, ensuring no bias in the results produced. Flooding and gossiping protocols of this form are commonly used within more complex protocols [5] to establish delivery routes or maintain awareness of network status, widening the scope of our results to all such protocols.

The protocol makes no demands of a node wishing to broadcast a packet, either for packets newly created by the application or when forwarding packets. When a packet is broadcast, each recipient makes an independent probabilistic decision whether to rebroadcast the packet to its neighbors, if it is not to be dropped or consumed. The packet thus radiates outward from the source node, hopefully arriving at least once at each intended destination.

In addition to the protocol-independent controlled factors defined above in Section III-D2, an additional controlled factor must be specified.

X_6: **Gossip rebroadcast probability.** The probability that upon receiving a packet, which is not to be consumed or dropped at the recipient, a given node will enqueue the packet for later retransmission to its neighbors. Unitless. Defined in the range [0, 1]. Search range is [0, 1].

4) Implicit Geographic Forwarding Protocol: An adapted form of the *IGF* [6], in which the backtracking support is disabled for simplicity as the test networks are void-free, is the second protocol under consideration. This protocol is ignorant of energy, network topology, and the host application, ensuring no bias in the results produced. Sensornets are strongly bound to their deployment environment, and are inherently geography-aware [48]. Geographical context can therefore be exploited in decision making, avoiding the overheads associated with maintaining logical network routing tables.

Unlike flooding-derived protocols, IGF implements a three-phase handshaking sequence to moderate data packet broadcast. Consider a packet p with source A and destination D, currently at node S. Node S broadcasts a short *request-to-send* (RTS) received by neighboring nodes $N_i \in N_{neighbors}$. Each RTS recipient N_i considers its geographic position relative to S and D, and if the angle $\angle DSN_i < \theta$ (where θ is a controlled factor, X_7) N_i broadcasts a short *clear-to-send* (CTS). $\angle DSN_i$ is trivially 0° if $N_i = D$. If S receives one or more CTS replies, it selects the node N_i offering the smallest $\angle DSN_i$ and selects this as the next recipient. Packet p is then broadcast with this choice added to its header. All neighbors $N_i \in N_{neighbors}$ except the selected N_i can safely ignore p. When the selected N_i receives p, it sends a short acknowledgement (ACK) to S, completing this stage of the process. The process repeats, with the previous N_i becoming the new S, until the packet arrives at D or a node N_i with no suitable forwarding candidate neighbors.

The current simulated time is given by τ from each simulated node's internal real-time clock. Function $SEQ(\pi)$ extracts the IGF sequence number from packet π. Function $ANGLE(\sigma)$ extracts the CTS angle stored in CTS packet σ. Function $SENDER(\pi)$ obtains the identity of the last sender of packet π; this is not necessarily the original source. Function $DEST(\pi)$ extracts the destination specified in packet π. Function $RELAY(\pi)$ extracts the selected next-hop relay node from packet π if defined for π.

In addition to the protocol-independent controlled factors defined above in Section III-D2, two additional controlled factors must be specified.

X_7: **CTS threshold angle.** When node N_i receives a CTS message from S, it will not send an RTS unless $\angle DSN_i < X_7$. This factor is intended to prevent many low-quality or poorly located forwarding candidates sending RTS messages and prevents packets being forwarded in the opposite direction to the destination if $X_7 < 90$. Measured in *degrees*. Defined in the range [0, 180]. Search range is [5, 85].

X_8: **State timeout base.** Complete IGF cycles imply several wait/timeout periods. To minimize the search space, we define all as multiples of a single parameter X_8, such that $CTS\_WAIT = X_8$, $DATA\_WAIT = 2X_8$, and $ACK\_WAIT = X_8$. Measured in *seconds*. Defined in the range $(0, \infty)$. Search range is [0, 1].

E. Measurable Attributes of Solution Quality

To assess the relative or absolute quality of a given candidate protocol tuning, we must first make measurable its behavior in a representative application context. Each metric should correspond to some desirable notion of solution quality. For sensornet protocols, the appropriate categories of solution quality are *performance*, *reliability*, and *efficiency* [20]. A set of solution quality metrics labeled M_1 to M_5 is defined below. For all metrics, lower values indicate a higher quality solution. A value of zero represents a perfect solution in a given metric, although this may not be attainable; the optimal value may be somewhat higher, but the value is not known in all cases.

1) Performance: We measure performance in terms of *packet latency*, the average time taken for a packet to traverse unit distance within the network. Where the physical network topology is such that for all nodes the distance between that node and all of its immediate communication partners is equal, these metrics are equivalent. For all other network topologies, including those considered in this paper, these metrics are nonequivalent but with correlation of strength defined by the standard deviation of all values in the set node-neighbor physical distances.

M_1: **Latency per unit physical distance.** The average amount of time taken for a packet to progress 1 m from the source node toward its destination within the network along some arbitrary delivery path. Measured in *seconds per meter*. Defined in the range $(0, \infty)$.

M_2: **Latency per unit network distance.** The average amount of time taken for a packet to progress a single node-node hop from the source node toward its destination within the network along some arbitrary delivery path. Measured in *seconds per hop*. Defined in the range $(0, \infty)$.

2) Reliability: We measure reliability in terms of *packet delivery failure proportion*, the proportion of packets created within the network which are not successfully delivered. In this paper, we assume that all packets have a single source and a single destination. For network application in which packets have multiple destinations, the network designer must define the notion of successful delivery strictly to calculate this metric. For example, in some applications, a multicast packet might be considered successfully delivered if at least one possible destination node receives the packet, or all possible destinations receive the packet, or some proportion between these extremes, or some more complex definition such as reaching a set of geographic regions.

M_3: **Packet delivery failure proportion.** The percentage of packets created by the simulated application which the network attempted to deliver but were lost before reaching the intended destination(s). Unitless. Defined in the range [0, 1].

3) Efficiency: We measure efficiency in terms of *energy per unit distance*, the amount of energy required for a packet to traverse unit distance within the network. It is generally impossible to determine how much energy was consumed in the successful or unsuccessful delivery attempt for any individual packet, as this attempt influences and is influenced by all other contemporary packet delivery attempts. A value of zero would indicate that the network consumes zero Joules in delivering a packet; this is obviously not attainable in a real network, but values closer to zero are more desirable. As with the *performance* metrics described in Section III-E1, we must differentiate between the nonequivalent notions of physical and logical network distance.

M_4: **Energy per unit physical distance per packet.** The average energy consumed by the network to successfully progress a single packet by 1 m from the source node toward its destination within the network along some arbitrary delivery path. Measured in *Joules per packet per meter*. Defined in the range $(0, \infty)$.

M_5: **Energy per unit network distance per packet.** The average energy consumed by the network to successfully progress a single packet by 1 m from the source node toward its destination within the network along some arbitrary delivery path. Measured in *Joules per packet per hop*. Defined in the range $(0, \infty)$.

F. Comparing Quality of Candidate Solutions

The metrics M_1 to M_5 defined above are all mutually independent, although any given pair may or may not exhibit some degree of correlation. It follows that M_1 to M_5 may be targeted as individual objectives by sensornet designers.

However, real sensornet designs are likely to require an acceptable compromise between multiple competing objectives. It is therefore necessary to define a mechanism by which the relative quality of two or more candidate solutions can be compared to determine which offers the best compromise.

Identifying the Pareto-optimal front from a set of points in a multi-objective search space is an important goal of many multi-objective search techniques. Both SPEA2 and TA aim to approximate the Pareto-optimal front, which in turn summarizes the set of good compromise solutions in which further improvement in a given measured attribute necessarily implies weakening of one or more other measured attributes. Whereas the Pareto-optimal front is invaluable in understanding the interrelationships inherent in the problem, it defines many possible compromise solutions and as such cannot be implemented directly within the sensornet; the designer must select exactly one protocol tuning for deployment.

Having obtained an approximation of the Pareto-optimal front by application of a MOEA such as SPEA2 or TA, or an arbitrary set of sampling points such as that provided by FD experiments, we can then select a single candidate solution for deployment. We achieve this by defining a measure of solution quality which allows the sensornet designer to specify the relative importance of each measurable attribute, and which takes into account the observed range of each measurable attribute. This solution quality measure defines an ordering of candidate solutions by quality, from which the sensornet designer can simply select that with the best quality measure.

Assume we have n controlled factors $X_1 - X_n$ and m metrics $M_1 - M_m$. A candidate solution $S_\alpha = \{X_{\alpha 1}, \ldots, X_{\alpha n}\}$ maps to a set of metrics $T_\alpha = \{M_{\alpha 1}, \ldots, M_{\alpha m}\}$. The mapping of $S \mapsto T$ is not known *a priori* but instead is evaluated experimentally as described in Section III-B for specific values of S. A perfect solution $S_{perfect}$ would yield a set of metrics $T_{perfect}$ such that $\forall M \in T_{perfect} \bullet M = 0$. Although $S_{perfect}$ does not necessarily exist, we define the quality measure E in (1) of any given candidate solution S_α based on the Euclidean distance from the point in solution phase space defined by T_α to the point $T_{perfect}$.

Some network performance attributes may be of greater importance than others to a sensornet designer. We therefore define weighting w_i for metric M_i such that a larger weighting value indicates a greater importance attached to the network performance attributes quantified by a given metric

$$E = \sqrt[2]{\sum_{i=1}^{m} w_i (s_i M_i)^2}. \tag{1}$$

Each metric $M_1 - M_m$ may be defined over a different range, so it is inappropriate to compare the absolute measured values directly. We define a scaling factor s_i for metric M_i such that all possible values of $s_i M_i$ are found in the range

[0, 1], noting that the the ideal value of any given metric is also the lowest possible value, 0. It is only meaningful to compare two E values if all scaling values s_i are equal for each E. If for a given metric M_i is defined over a finite range, then, the value of s_i is well-defined and does not vary between network configurations under consideration. However, if a given metric M_i is defined over an infinite range, then, there does not exist a single well-defined value of s_i. We define s_i for a given set of experimental results by setting $s_i = MAX(M_i)^{-1}$ where $MAX(M_i)$ is the largest metric M_i value observed.

It is important to remember that minimizing the E metric is *not* the objective of a *single-objective* optimization process; it is merely a tool with which to select a single best compromise solution from the set of high-quality candidates identified by the *multi-objective* optimization process.

In this paper, we define a generic experimental method for sensornet protocol optimization and provide a case study of its application in Section VII. In this case study, we consider a distributed sensing application in which each metric is of equal significance to the overall solution quality. As this optimization problem does not assign greater significance to any specific quality attribute, in the experimental work that follows, we set all $w_i = 1$ to give equal weighting to all metrics.

We set all s_i using the second definition above as some metrics defined in Section III-E are defined over an infinite range. As we consider 5 metrics $M_1 - M_5$, it follows that all values of E are defined in the range $[0, \sqrt{5}]$ where 0 is the solution quality deriving from the theoretically perfect solution, and $\sqrt{5}$ is the solution quality deriving from the worst quality solution considered in the set of all experiments. For the experimental results we present in Section VII, we scale all E values by a constant factor of $(\sqrt{5})^{-1} = 5^{-(1/2)}$, mapping the range $[0, \sqrt{5}]$ to [0, 1] for ease of comparison while remaining exactly equivalent to the original.

Sensornet designers may need to find solutions conformant to specific constraints in one or more metrics. For example, if the sensornet application requires at least 50% of packets to be delivered successfully, then, candidate solutions for which $M_3 \geq 0.5$ are unacceptable. This can be achieved by extending the solution quality function defined in (1) to assign infinite E-values to candidate solutions which do not satisfy some arbitrary set of requirements.

Assume a Boolean function $reject(T_\alpha)$ applies the set of relevant tests to the output metrics T_α corresponding to candidate solution S_α, returning $false$ if acceptable and $true$ if not acceptable. We extend (1) using Iverson notation [49] to obtain (2) which assigns $E = \infty$ for unacceptable solutions and leaves E unchanged for acceptable (though perhaps suboptimal) solutions. Candidate solutions S_α for which $E(T_\alpha) = \infty$ can be considered during the search process but are then discarded explicitly prior to implementing model fitting as per Section IV-B under the FD approach, or discarded implicitly as part of the normal evolutionary process of the SPEA2 and TA approaches as appropriate

$$E = \sqrt[2]{\sum_{i=1}^{m} w_i(s_i M_i)^2 + [reject(T_\alpha)]\,\infty}. \tag{2}$$

G. Adapting for Other Protocols

The principled and evolutionary search methods described in this paper can be applied to other protocol tuning problems, as the principal steps are protocol-agnostic.

1) *Derive a set of controlled factors.* These are extracted directly from the protocol definitions.
2) *Define the search space.* At this stage, it is acceptable to base initial guesses on prior experience, provided that these are used only as starting points for trial-and-improvement experiments to define useful ranges for controlled factors.
3) *Define a set of measurable attributes.* These pertain to solution quality in the context of the specific optimization problem, and their ranges or magnitudes are considered implicitly by following the experimental procedure.
4) *Define any applicable constraints.* Any such constraints are likely to be application specific, and hence are not given further coverage in this paper.
5) *Implement search-based optimization.* Apply the FD methods described in Section IV, or the evolutionary methods described in Sections V and VI, or some combination thereof, to obtain a near-optimal solution.

H. Variance Analysis

We can reduce the cost of the most expensive component, fitness function evaluation, by reducing the time within which network simulations run. However, if this period is too small, we risk unacceptable levels of experimental error leading to meaningless results. We mitigate this risk in Phase 1 by analyzing the variance of network metrics with respect to simulated time, calculating the minimum required for an acceptable and defined level of experimental error.

Determining an appropriate network stabilization period, prior to measurement, is nontrivial but nevertheless possible. We used a well-known experimental approach in which we measured variance of metrics over time, and evaluated the stabilization period separately for each metric. We adopted a conservative strategy in which the longest observed stabilization period was selected, and a significant safety margin was added beyond any observed measurements.

We measured the minimum time required for each metric under each protocol by experiment. Metrics are sampled periodically but are influenced by total simulated period from the start to the sampling point. Assuming that the network eventually reaches a steady state, measured metrics converge on the actual value with sample accuracy increasing with simulated time, until sampled values fall within experimental error margin at which point no further meaningful improvement is possible. At this point the metrics can be sampled as representative of the stable value [50].

Identical examples of network design and protocol were used in variance analysis and protocol optimization work, so the measured stabilization periods are appropriate. We measured stabilization periods for protocol configurations taken from the center of search ranges, and in which the assigned value for a given parameter was set as the range minimum or maximum of

TABLE I
PHASE 1: τ_α VALUES FOR METRICS $M_1 - M_5$

	M_1	M_2	M_3	M_4	M_5
TBG	19	43	58	49	46
IGF	27	78	63	51	61

the search range. We defined the stabilization time as being the greatest observed time among the set of these configurations, including the extreme configurations.

Assume the value of some convergent metric M_α at simulated time τ is given by $M_\alpha(\tau)$. $M_\alpha(\tau)$ approaches its limiting value $M_\alpha(\infty)$ as $\tau \to \infty$. At some simulated time τ_α, the value $M_\alpha(\tau_\alpha)$ becomes sufficiently close to $M_\alpha(\infty)$ such that for all $\tau > \tau_\alpha$, the value $M_\alpha(\tau)$ is within $\pm\eta\%$ of $M_\alpha(\infty)$. We define metric M_α as *converged* at this simulated time τ_α. Any further variation, including that deriving from noise and unblocked nuisance factors, is within $\pm\eta\%$ experimental error margin. We set $\eta = 5$ such that measured metrics used in later analysis have $\pm5\%$ measurement error.

Consider $M = \{M_1, M_2, M_3, M_4, M_5\}$, the set of metrics defined in Section III-E. Table I presents τ_α measured experimentally for metrics $M_\alpha \in M$ with each value rounded as $\lceil \tau_\alpha \rceil$. For the TBG protocol, $\forall M_\alpha \in M \bullet \tau_\alpha < 60$ s. For the IGF protocol, $\forall M_\alpha \in M \bullet \tau_{Ci} < 120$ s. We therefore select simulation length $\tau_{sim} = 120$ s, allowing a large safety margin for any anomalous solution instability.

IV. DOE APPROACH: FACTORIAL DESIGN

In this section, we define a DOE method based on FD with which to explore the parameter landscape at broad scope but shallow depth.

A. Two-Phase Experiment Design

FD [7] is used to systematically explore the entire parameter landscape. This approach gives broad but shallow coverage of all possible combinations of all acceptable ranges of controlled factors. Statistical models are fitted to experimental results to yield a generalized model of the relationship between controlled factors and each measured response. This model is useful for predicting likely network performance for any arbitrary set of input values [50]. The model can be used in the opposite direction by defining sections of the multiresponse hypersurface corresponding to the desired network performance, and working backward to input values by solving the simultaneous equations of the fitted model to yield a set of inequalities defining usable ranges of input-controlled factors.

We address the combinatorial explosion by applying a two-phase method. Phase 1 allows the experimenter to identify which of the controllable factors are actually important, and which can be safely ignored. Phase 2 explores the significant controllable factors in much greater detail. The experimenter can therefore avoid wasting resources and analytical effort on matters which will not significantly influence the outcome, and more detailed statistical models can be derived for the same experimental cost. Before Phase 1 begins, we assume that the variance analysis described in Section III-H has completed to determine the experimental parameters necessary to obtain meaningful and repeatable results.

B. Model Fitting

The FD of the experiment suite described in Section IV-A samples the parameter space at q^p points as described in Section IV-C. These pairs of sample points and simulation-derived metrics represent exact solutions to specific known points in the generalized model of the relationship between controlled factors and output metrics. However, these are not directly usable if we wish to know the relationship between input and output, or vice-versa, for other points in the input-output phase space.

To consider points in the parameter space that have not been measured directly, we need to interpolate by fitting a statistical model to the known sampled points to derive a set of equations describing a hypersurface in the phase space [51]. We then work with the fitted surface rather than specific individual experimental results. An appropriate statistical model must be selected, which yields a surface with shape similar to that which would be observed if an infinite number of sample points were used. Previous work [20] has shown that linear first-order interaction models are a suitable approximation for the protocols considered in this paper; we confirmed this by examining the correlation coefficient between the measured and predicted values for the experiments in Section IV-D.

Sampling the parameter space at more points yields a fitted model which is a better approximation of the real relationship by providing more data for the model fitting algorithm. For a finite set of sample points, there exists the risk that an interesting feature of the solution landscape falls between sample points, and hence is not present in the fitted model.

For each output metric under consideration, a linear interaction model of the form given in (3) was fitted to the result set in MATLAB. β_0 is a constant, X_i is the ith controlled factor value, β_i is the coefficient for controlled factor X_i, β_{ij} is the coefficient for the interaction between controlled factors X_i and X_j, and ε is the normally distributed noise term. The response Y is influenced linearly by each factor and each pairing of potentially interacting factors

$$Y = \beta_0 + \sum_{i=1}^{n} \beta_i X_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \beta_{ij} X_i X_j + \varepsilon. \tag{3}$$

For each metric $M_1 - M_5$, a separate linear interaction model is produced in which six axes represent controlled factors $X_1 - X_6$ and a further axis in which the height of the hypersurface varies with the values of the output metric M_i. Axes corresponding to controlled factors $X_1 - X_6$ are common to all metrics $M_1 - M_5$ so a more complex surface can represent the interrelationships between all factors and all metrics.

Finding sets of values for controlled factors corresponding to solutions with appropriate characteristics is equivalent to identifying regions of the axes representing controlled factors $X_1 - X_6$ with appropriate fitted surface height in the axes corresponding to output metrics $M_1 - M_5$.

Similarly, finding optimal or worst case sets of controlled factors is equivalent to finding minima and maxima in the fitted surface. This is implemented by solving sets of simultaneous inequalities when identifying regions with suitable characteristics, or by solving sets of simultaneous equations when addressing optimal or worst case characteristics.

C. Cost Analysis

Given p controlled factors, each sampled at q points in the permitted region, we have q^p protocol configurations to assess. We assess each protocol configuration with r networks to prevent results being unduly influenced by a given network design, yielding rq^p experimental configurations to consider by simulation. We repeat each experimental configuration s times to prevent results being unduly influenced by any single unusual simulation instance, yielding the requirement to run rsq^p simulations in total.

Assuming each simulation completes in approximately equal wall time, t, we find that total experiment time grows exponentially in p, polynomially in q, and linearly in r and s. As total experiment time is exponential in p, it is obvious that any reduction in p is valuable and is more significant than similar reductions in q, r, or s. Phase 1 addresses this problem by identifying controllable factors which can safely be disregarded. It is therefore possible in Phase 2 to increase q after reducing p and still have the full experiment set complete in acceptable wall time.

All simulations are mutually independent and can therefore be executed in parallel, reducing the total runtime to that of a single simulation if sufficient processing hosts are available. Assume a multiprocessing environment in which $x \in \mathbb{N}$ independent simulations can execute in parallel. For FD experiments, there are no dependencies between simulations so any number can execute in parallel, all at cost t. The total wall time cost C is given by (4). Note that $C \propto 1/x$, reaching a minimum of $C = t$ where $x = rsq^p$

$$C = \frac{rsq^p}{x} t. \tag{4}$$

Increasing the number of experimental configurations increases the quality of fitted statistical models, and hence solution quality, but also increases experiment cost. A balance must be found which obtains solutions of acceptable quality within reasonable time. We measured wall time for all experiment simulations and took the mean as $t = 78.51$ s.

In Phases 1 and 2, we set $r = 3$ and $s = 3$ to evaluate each candidate solution several times in a set of dissimilar networks to minimize the influence of outliers and to ensure that the derived solutions are not biased heavily in favor of a single network design. The number of parameters p in Phase 1 is taken directly from the set of controlled factors defined for each protocol in Section III-D. In Phase 2, the value of p is defined by the number of controlled factors found to be statistically significant in Phase 1.

As we will fit linear models to the results for factor significance screening in Section IV-D1, we must consider at least two values for each controlled factor, such that $q_{\min} = 2$. If $q = 2$, we take the first such value at the low extreme of the defined interval, and the second at the high extreme. If $q > 2$, we again define sample values at the low and high extremes and distribute the remaining sample values evenly throughout the sampled range between these extremes. This ensures that sampling covers the full spectrum of possible behavior.

D. Intermediate Results

In this section, we present the results of applying the two-phase experimental method described in Section IV-A to the TBG and IGF protocols. The intermediate results obtained by this mechanism are used to calculate the final results for the FD approach presented in Section VII.

1) Phase 1—Factor Significance Screening: In Phase 1, we identify which of the protocol controlled factors are the best predictors of the network performance metrics. This requires a small number of points in the parameter space to be sampled in the axis corresponding to each controlled factor, and a set of simulation experiments to be run to measure network performance under each combination. The ANOVA method is applied to assess which controlled factors are significant to the experimental outcomes [51]. Any factors which are deemed statistically insignificant are dropped.

In the interests of brevity, we do not include the full p-value data sets in this paper, as we need consider only the summarized results presented below. However, the full p-value data sets are available for download [52].

Controlled factors $\{X_1 - X_6\}$ were considered at this stage for protocol A. The test suite size was calculated using the formula given in Section IV-C with $p = 6$, $q = 3$, $r = 3$, and $s = 3$. This gives a test suite size of $3 \times 3 \times 3^6 = 6561$, hence 6561 points in the factor–response phase space are available for model fitting.

Factors $\{X_4, X_5, X_6\}$ are significant in isolation with 95% confidence $(p < 0.05)$ for at least two of the metrics $M_1 - M_5$, and at least one of $\{X_4, X_5, X_6\}$ is evident in almost all interaction pairs deemed significant with 95% confidence. Factors $\{X_1, X_2, X_3\}$ are not significant in isolation for any metric, or as a member of an interaction pair which does not include any of $\{X_4, X_5, X_6\}$. Notably, the protocol-specific factor X_6 is statistically significant indicating that attempts to tune this protocol are appropriate.

Controlled factors $\{X_1 - X_5, X_7 - X_8\}$ were considered at this stage for protocol B. The test suite size was calculated using the formula given in Section IV-C with $p = 7$, $q = 3$, $r = 3$, and $s = 3$. This gives a test suite size of $3 \times 3 \times 3^7 = 19\,683$, hence 19 683 points in the factor–response phase space are available for model fitting.

Factors $\{X_4, X_7, X_8\}$ are significant in isolation with 99% confidence $(p < 0.01)$ for all metrics $M_1 - M_5$. The controlled factor X_2 is significant with 99% confidence $(p < 0.01)$ for metric M_1 and significant with 90% confidence $(p < 0.1)$ for metric M_3. At least one of $\{X_2, X_4, X_7, X_8\}$ is evident in all interaction pairs deemed significant with at least 95% confidence $(p < 0.05)$.

Other methods exist for factor significance screening and problem dimensionality reduction, including *principal* *component analysis*, *projection pursuit*, *topologically continuous maps*, and *vector quantization* [53]. We do not consider these alternative methods in this paper, but they could similarly be applied to identify controllable factors which can safely be discarded to reduce the problem size.

2) Phase 2—High Resolution Modeling: In Phase 2, we sample the parameter space along the corresponding axis in a greater number of points for each statistically significant controlled factor. Again, a set of simulation experiments was performed to measure network performance under each configuration. We extract a near-optimal protocol tuning from the results of the corresponding experiments. The ANOVA method was reapplied to confirm that significant factors were selected. Phase 1 identified which controllable factors are not significant to the outcome, for which the values selected are unimportant provided that they fall within the ranges explored in Section III-D; we select the midpoints of these ranges.

Factors $\{X_4, X_5, X_6\}$ were considered at this stage for the TBG protocol. The test suite size was calculated using the formula given in Section IV-C with $p = 3$, $q = 10$, $r = 3$, and $s = 3$. This gives a test suite size of $3 \times 3 \times 10^3 = 9000$, hence 9000 points in the factor–response phase space are available for model fitting. Factors $\{X_2, X_4, X_7, X_8\}$ were considered at this stage for the IGF protocol. The test suite size was calculated using the formula given in Section IV-C with $p = 4$, $q = 7$, $r = 3$, and $s = 3$. This gives a test suite size of $3 \times 3 \times 7^4 = 21\,609$, hence 21 609 points in the factor–response phase space are available for model fitting.

Experimenters interested in understanding the generalized relationship between significant factors and measurable responses can fit statistical models to the sampled points. This yields a set of coefficients for the selected statistical model which summarize the relationship between factors and responses for the selected protocol and network configuration as a set of simultaneous equations. However, we are interested in extracting a single near-optimal protocol tuning. As we have a reasonably dense grid sampling of the parameter space, we do not require the interpolation effect offered by model fitting, and can evaluate our experimental data directly.

We calculate the solution quality metric E, defined in Section I, for each of the rsq^p sampling points. To mitigate the influence of outliers and experimental noise, we calculate the mean value of E from the rs experiments corresponding to each of the q^p unique candidate solutions. We select the candidate solution associated with the lowest mean E value, and hence highest solution quality. Section VII presents the resulting solutions for the TBG and IGF protocols.

V. Evolutionary Approach: SPEA2

In this section, we define the experiments with which the parameter landscape is explored, at narrow scope but substantial depth, using evolutionary approaches and the SPEA2.

A. SPEA2 Experimental Configuration

SPEA2 [8] is a revised version of the strength Pareto evolutionary algorithm designed by Zitzler *et al.* [54]. The population size s defines the size of the working population and is also used as the capacity of the archive. In this instance, although we are examining 5 metrics, the uniform nature of the experimental networks produces significant correlation between the per-meter and per-hop variants of the energy and latency metric values for a given network. Consequently, we choose $s = 50$ based on the 3-metric value of the population scheme used by Khare *et al.* [55]. SPEA2 also takes a value k, which determines the k-th nearest neighbor in density estimation calculations. As in the original SPEA2 experiments [8], we take $k = \sqrt{2s}$, or the square root of the sample size.

Simulated binary crossover (SBX) [56], polynomial mutation [56], and random selection operators were used for this experiment. SBX takes as parameters a crossover rate, c, and an η_c value controlling the probability of *near-parent solutions* being generated; higher values produce closer matches to parents. Appropriate values must be selected for these parameters; there is some debate as to the relative importance of crossover and mutation [57]. The crossover rate was set to 0.7, so that crossover occurred often to generate a diverse range of child solutions. η_c was set to 15, to encourage relatively “close” solutions to be generated. Polynomial mutation also takes a mutation rate m and an η_m value controlling the *mutation distance*. The mutation rate was set to 1/6 such that, on average, one input variable would be mutated in each solution. η_m was set to 20 to promote small mutation steps and thus encourage convergence.

Experiments were conducted in which all values were represented internally as 64-bit precision floats. Where a given parameter is defined only for integral values, the float value was rounded down to the nearest integer at the point of use. Each candidate solution fitness evaluation considered the three networks defined in Section III-C, with each combination of candidate solution and network repeated three times to reject the influence of outliers.

Preliminary tests showed rapid convergence within the early generations, with few improvements thereafter. Based on these results, all tests were run for 50 generations to allow convergence to occur. Data on the best known candidate solutions were logged at every generation to provide insight into the running convergence of the system.

B. Cost Analysis

The SPEA2 algorithm runtime is negligible compared to that of fitness function evaluation by simulation, so we need to consider only those overheads relating to fitness function evaluation. Consider an evolutionary run with a population size of a for which b generations are required to attain the required solution quality. It is also necessary to evaluate the initial zeroeth population prior to evolution commencing.

Within each generation, it is necessary to evaluate the fitness function once for each candidate solution, requiring $a(b + 1)$ evaluation instances for all population members across all generations. As with the FD experiments, we test r networks and repeat each experimental configuration s times, requiring $a(b + 1)rs$ simulations in total. Total cost grows linearly in each of a, b, r, and s. This predictable and readily controllable cost growth is a desirable attribute.

It is possible, though unlikely, that a candidate solution already in the archive be selected as a parent but no crossover or mutation occurs. In this rare situation it would be possible to reuse an earlier fitness evaluation and hence reduce experimental cost. However, we do not cache fitness evaluation data between generations. By forcing evaluation of each candidate solution in each iteration, we significantly decrease the possibility that a low-quality candidate, assigned an undeservedly high-quality evaluation owing to experimental noise, can survive from generation to generation.

VI. Evolutionary Approach: Two-Archive

In this section we define the experiments with which the parameter landscape is explored, at narrow scope but substantial depth, using evolutionary approaches and the TA evolutionary algorithm.

A. *Two-Archive Experimental Configuration*

The TA algorithm is a multi-objective EA developed by Praditwong and Yao [9]. The population size s is used as the size of the working population, and also as the total capacity of the convergence and diversity archives combined. We again use $s = 50$ to match the SPEA2 experiment. TA also takes a parameter r which defines the ratio of parent selection from the convergence and diversity archives. A higher ratio leads to more solutions being chosen from the convergence archive, and thus faster solution convergence is observed at the cost of potentially reduced diversity in the range of solutions. We set $r = 0.9$ so as to favor parents from the convergence archive, thus ensuring strong convergence in the algorithm.

As with SPEA2, SBX [56], polynomial mutation [56], and random selection operators were used for this experiment. Identical values for crossover/mutation rates and η values were used: a crossover rate of 0.7, a crossover η_c value of 15, a mutation rate of 1/6, and a mutation η_m of 20, to ensure comparable results. Each candidate fitness evaluation implied three simulation iterations for each of the three networks defined in Section III-C to reject the influence of outliers. All tests were run for 50 generations.

B. *Cost Analysis*

The TA algorithm cost is similar to that of the SPEA2 algorithm cost, as discussed in Section V-B. As with SPEA2, the cost of executing the EA itself is negligible in comparison to the cost of fitness function evaluation, so we need only consider those costs relating to the latter. As with the SPEA2 experiments, we require $ars(b + 1)$ simulation instances in total. Total cost grows linearly in each of a, b, r, and s.

VII. Results

We present the results for FD, SPEA2, and TA experiments described in Sections IV–VI, respectively. We label the FD instance as A, the SPEA2 instance as B, and the TA instance as C.

TABLE II
Best Known TBG Tunings

	X_1	X_2	X_3	X_4	X_5	X_6
I_A	*6*	*6*	*0.5500*	2.8200	10	0.7300
I_B	5	3	0.8886	1.1151	9	0.7583
I_C	5	3	0.3864	6.8320	9	0.8425

TABLE III
Network Performance for Best Known TBG Tunings

	M_1	M_2	M_3	M_4	M_5	E
O_A	9.0312	15475	439010	0.39873	**758.28**	0.50528
O_B	8.7817	15458	453610	0.43416	847.75	0.50669
O_C	**8.1415**	**14297**	**419650**	**0.39808**	781.58	**0.46875**
Best	TA	TA	TA	TA	FD	

A. *Optimized Protocol Tunings for TBG*

Tables II and III give summarized optimized tuning results for the TBG protocol. The sets of tuned protocol values corresponding to A, B, and C are labeled I_A, I_B, and I_C, respectively. All figures for nonintegral factors are given to four decimal places.

For each experimental approach, the set of values assigned to controlled factors $X_1 - X_6$ corresponding to the highest quality solution discovered is given in Table II. For approach A, some controlled factors were not evaluated directly in Phase 2 of the experiment. For these controlled factors, italicized in Table II, we take the midpoint of search ranges defined in Section III-D.

We define the highest quality solution I_α for approach α as being that which offers the smallest Euclidean distance E_α between O_α and the theoretical perfect values of metrics, as defined in Section III-E and F. Table II shows the Euclidean distances E_A, E_B, and E_C from which I_A, I_B, and I_C were identified as the highest quality solutions derived by approaches A, B, and C, respectively. Note that the theoretical perfect metric values are not necessarily attainable under *any* real protocol tuning.

To ensure fair comparison of the quality of solutions obtained by the two experimental approaches, it is necessary to eliminate any factors which could unfairly influence the outcome. We achieved this goal by conducting further simulation experiments as per Section III where the simulation scenario is identical in all respects, except for the protocol parameter set which is either I_A, I_B, or I_C, as appropriate.

Three hundred simulations were executed for each of I_A, I_B, and I_C as defined in Table II; 100 repeats for each of the three networks considered in the experiments of Sections IV–VI. Where a controlled factor $X_1 - X_6$ is defined only for integral values, but the value identified by experiment and analysis is nonintegral, we configure our experiments with values rounded to the nearest integer. For each combination of experimental approach and metric $M_1 - M_5$, a set of 300 output values is produced. The arithmetic mean E of each set is taken as the final value and presented in Table III. The sets of output metrics corresponding to A and B are labeled O_A, O_B, and O_C, respectively.

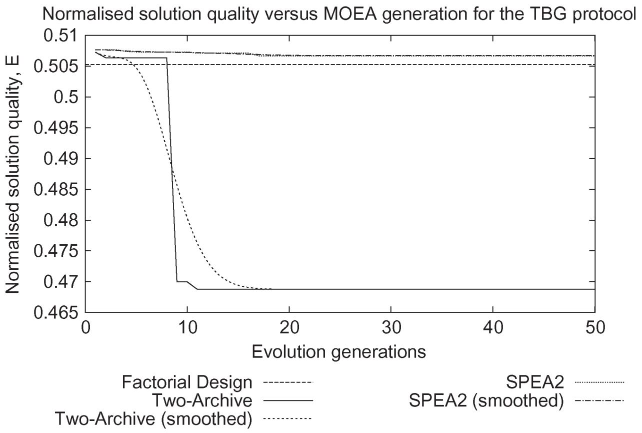

Fig. 1. Comparing TBG solution quality: factorial design, SPEA2, and TA.

All figures for $M_1 - M_5$ are given to five significant figures and are scaled by a factor of 10^6 for clarity. The best results are highlighted in bold type.

It is notable that the protocol tunings found by the three approaches, as shown in Table II, are different but broadly similar. The corresponding metrics given in Table III confirm that the network behavior induced by each of the three approaches was different but broadly similar.

The similarity of these protocol tunings, and the network behavior they induce, suggests that they are all located within a region of the parameter landscape which contains good solutions. The differences suggest that within this region, there may be multiple protocol tunings of nearly equal overall quality, but some difference in the balance of competing design goals. It is possible, though not guaranteed, that an optimal solution may also exist within this region.

Observe that the TA approach yielded the best result for metrics $M_1 - M_4$, the FD approach yielded the best result for metric M_5, and the SPEA2 approach did not yield the best result for any metric. It is therefore unsurprising that TA yielded the solution with best overall quality as measured in E, followed by FD and SPEA2 in order of declining quality. We conclude that the TA approach is superior for tuning the TBG protocol, but that any of those approaches considered in this paper would yield a reasonable solution.

Fig. 1 plots normalized solution quality E versus evolutionary generation to illustrate the convergence of SPEA2- and TA-derived solution quality toward the final solution quality which could be achieved with an infinite number of generations. For comparison, the solution quality obtained by the FD approach is also shown as a constant.

The quality of solutions obtained by the SPEA2 approach was not observed to supplant that attainable by FD results, but gradually improved until generation 18 after which no further improvement was observed. Although theoretically possible, the attainment of parity or advantage by further improvement under SPEA2 is unlikely to occur within acceptable time. Under the TA approach, however, the solution quality is near-constant for the first nine generations before improving dramatically, at which point it becomes significantly better than that attained by FD, with no further improvement observed after generation 11.

TABLE IV
BEST KNOWN IGF TUNINGS

	X_1	X_2	X_3	X_4	X_5	X_7	X_8
I_A	*6*	3	*0.5500*	10.000	*6*	31.6667	0.1000
I_B	7	6	0.5156	6.5660	1	42.9460	0.2642
I_C	7	6	0.1639	7.2508	1	46.9391	0.6605

TABLE V
NETWORK PERFORMANCE FOR BEST KNOWN IGF TUNINGS

	M_1	M_2	M_3	M_4	M_5	E
O_A	53.786	122510	338380	29.229	72626	0.15701
O_B	129.40	292880	**280390**	**23.773**	**58107**	0.15694
O_C	**38.853**	**82915**	303150	25.262	62262	**0.13918**
Best	TA	TA	SPEA2	SPEA2	SPEA2	

We now consider the tradeoff between experimental cost and solution quality between the experimental approaches. Recall from Section IV-C that FD experiments require rsq^p simulation instances for each of Phase 1 and Phase 2. For these experiments, this implies that $6561 + 9000 = 15\,561$ simulation instances are required. We compare the experimental costs of the evolutionary approaches to this baseline figure.

We first consider SPEA2, which requires $a(b + 1)rs$ simulation instances to be executed. Section V-A describes the experimental configuration for these simulation experiments. The best SPEA2-derived solution was obtained at generation 18, at which point 8550 instances had completed. This cost is 54.9% of the FD baseline. We conclude that significant cost advantage was observed for SPEA2 in tuning the TBG protocol if the process is terminated at this point.

Now, consider TA, also requiring $a(b + 1)rs$ simulation instances to be executed. Section VI-A describes the experimental configuration for these simulation experiments. TA produced a better solution than FD at generation 10 and produced its best solution at generation 11, corresponding to 4950 and 5400 simulation instances, respectively. As these costs are 31.8% and 34.7% of the FD baseline, and better solutions were obtained, significant cost advantage is observed for TA in tuning TBG if the process is terminated at this point.

B. Optimized Protocol Tunings for IGF

Tables IV and V give summarized optimized tuning results for the IGF protocol. The sets of tuned protocol values corresponding to A, B, and C are labeled I_A, I_B, and I_C, respectively. All figures for nonintegral factors are given to four decimal places.

For each experimental approach, the set of values assigned to controlled factors $X_1 - X_5$ and $X_7 - X_8$ corresponding to the highest quality solution discovered is given in Table IV. For approach A, some controlled factors were not evaluated directly in Phase 2 of the experiment. For these controlled factors, italicized in Table IV, we take the midpoint of search ranges defined in Section III-D.

Each IGF protocol tuning given in Table IV was evaluated using the process described for the TBG protocol tuning solutions as described in Section VII-A. Table II shows the Euclidean distances E_A, E_B, and E_C from which I_A, I_B, and I_C were identified as the highest quality solutions derived by

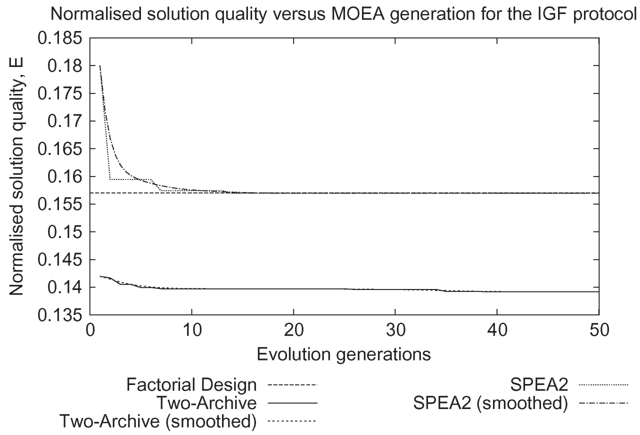

Fig. 2. Comparing IGF solution quality: factorial design, SPEA2, and TA.

approaches A, B, and C, respectively. Note that the theoretical perfect metric values are not necessarily attainable under *any* real protocol tuning.

All figures for $M_1 - M_5$ are given to five significant figures and scaled by a factor of 10^6 for clarity. The best results are highlighted in bold type.

The protocol tunings shown in Table IV have substantial differences in some controlled factors; notably, these are the factors identified as significant by the FD approach in Section IV-D1. The corresponding observed metrics given in Table V confirm that the network behavior induced by each of the three approaches was significantly different, despite each approach being applied to the same protocol tuning problem.

Observe that the TA approach yielded the best result for metrics $M_1 - M_2$, the SPEA2 approach yielded the best result for metrics $M_3 - M_5$, and the FD approach did not yield the best result for any metric. It is therefore unsurprising that SPEA2 yielded the solution with best overall quality as measured in E, followed by TA and FD in order of declining quality.

Interestingly, however, the SPEA2 values for $M_1 - M_2$ were an order of magnitude worse than those obtained under TA or FD, indicating that the SPEA2 values for $M_3 - M_5$ were much better than those of TA in order to counterbalance this disadvantage. We conclude that the SPEA2 approach is superior for tuning the IGF protocol, but that any of those approaches considered in this paper would yield a reasonable solution.

Fig. 2 plots normalized solution quality E versus evolutionary generation to illustrate the convergence of SPEA2- and TA-derived solution quality toward the final solution quality which could be achieved with an infinite number of generations. For comparison, the solution quality obtained by the FD approach is also shown as a constant.

The quality of solutions obtained by the SPEA2 approach improved quickly at first, but this improvement slowed quickly after the first few generations and converged on a very similar solution quality to that observed under FD; after generation 14, no further improvement was observed.

With TA, the solution quality gradually improved until generation 7, after which no further improvement was observed. Interestingly, all TA generations showed a higher solution quality than FD or SPEA2. Although it might be considered somewhat fortuitous that the first evolved generation was of such high quality, this nevertheless illustrates that the evolutionary strategy of TA is effective for this problem type.

Now, consider the tradeoff between experimental cost and solution quality between the experimental approaches. Recall from Section IV-C that FD experiments require rsq^p simulation instances for each of Phase 1 and Phase 2. For these experiments, this implies that $19\,683 + 21\,609 = 41\,292$ simulation instances are required. We compare the experimental costs of the evolutionary approaches to this baseline figure.

We first consider SPEA2, which requires $a(b+1)rs$ simulation instances to be executed as per Section V-B. The best SPEA2-derived solution was obtained at generation 14, at which point 6750 instances had completed. This cost is 16.3% of the FD baseline. We conclude that significant cost advantage was observed for SPEA2 in tuning the IGF protocol if the tuning process is terminated at this point.

Next, we consider TA, also requiring $a(b+1)rs$ simulation instances to be executed as per Section VI-B. TA produced a better solution than FD at generation 1 and produced its best solution at generation 7, corresponding to 900 and 3600 simulation instances, respectively. As these costs are 2.2% and 8.7% of the FD baseline, and better solutions were obtained, it follows that significant cost advantage is observed for TA in tuning IGF if the process is terminated at this point.

C. Comparative Cost Analysis

We now consider the relative costs of the tuning approaches. The costs of FD are described in Section IV-C, the costs of SPEA2 are described in Section V-B, and the costs of TA are described in Section VI-B. Note that in all cases, the overhead of auxiliary calculations is orders of magnitude less than that of fitness function evaluation, so we discount the former in our analysis.

Assume each simulation instance completes in t seconds. The FD approach has wall time cost C_α, given by (5). The SPEA2 and TA approaches have wall time cost C_β, given by (6)

$$C_\alpha = rsq^p t \tag{5}$$

$$C_\beta = ars(b+1)t. \tag{6}$$

Given a single uniprocessor host, the SPEA2 and TA approaches will terminate before the FD approach if $C_\beta < C_\alpha$, a condition which is fulfilled where $a(b+1) < q^p$.

Now, assume a multiprocessing environment in which x independent simulations can execute in parallel. For FD experiments, there are no dependencies between simulations so any number can execute in parallel, all at cost t. The total wall time cost C_γ is given by (7). Note that $C_\gamma \propto (1/x)$, reaching a minimum of $C_\gamma = t$ where $x \geq rsq^p$

$$C_\gamma = \frac{rsq^p}{x} t. \tag{7}$$

For SPEA2 and TA experiments, it is possible to run all ars simulations of a given generation in parallel at cost $arst$, but all simulations of a given generation must complete before the next generation can begin. The total wall time cost C_δ is given

TABLE VI
α, β, AND γ FOR COMBINATIONS OF MOEA AND SENSORNET PROTOCOL

		α	β	γ	R^2
TBG	Two-Archive	0.3456	4.887	0.4588	0.7403
	SPEA2	0.01115	6.575	0.5064	0.8891
IGF	Two-Archive	0.006443	1.164	0.1392	0.8817
	SPEA2	0.004746	-0.7953	0.1569	0.9838

by (8). Note that $C_\delta \propto (1/x)$, reaching a minimum of $C_\delta = (b+1)t$ where $x \geq ars$. If x is large, then, FD experiments will complete before SPEA2 and TA experiments.

$$C_\delta = \frac{ars}{x}(b+1)t. \tag{8}$$

We observe that the FD approach incurs a fixed wall time cost of C_α on a uniprocessor host, or C_γ on a multiprocessor host, regardless of solution quality. In contrast, the SPEA2 and TA evolutionary algorithms incur variable wall time costs of C_β on a uniprocessor host, or C_δ on a multiprocessor host, which are proportional to the number of generations, b. As the solution quality E is monotonically nondecreasing in b, a tradeoff exists between cost and quality, such that experimenters can restrict cost by specifying b.

D. *Quality of Evolving Solutions*

Sections VII-A and B show the potential savings in experimental cost which are possible by selecting an EA approach over a DOE approach to the protocol tuning problem. However, to achieve these potential savings, the experimenter must somehow determine the evolutionary generation at which the process is unlikely to yield further gains and hence should be terminated.

Figs. 1 and 2 plot normalized solution quality, E, versus evolutionary generation to illustrate the convergence of SPEA2- and TA-derived solution quality toward the final solution quality which could be achieved with an infinite number of generations. For comparison, the solution quality obtained by the FD approach is also shown.

The E plots for the EAs observe a step function as quality E increases discretely between generations when a better solution is found and added to the archive. Smoothing the discrete step function curve into a continuous curve by considering individual data as *control points* of a Bézier curve, we observe that a hyperbolic curve is a reasonable approximation.

We apply statistical model fitting techniques using *MATLAB Curve Fitting Toolbox* [58] to quantify the relationship between evolutionary generation and solution quality as given by (9). α, β, and γ are constants for each combination of EA and sensornet routing protocol considered in this paper, and $g \leq b$ is the generation for which an solution quality estimate E_g is required. Results are given in Table VI to four significant figures. R^2 values indicate the quality of fit [51]; all fittings given in Table VI are of sufficient quality for the intended purpose

$$E_g = \frac{\alpha}{g+\beta} + \gamma. \tag{9}$$

Sensornet designers can estimate the evolutionary generations g required to find a protocol tuning solution of sufficient quality E_{req} such that $E_g \leq E_{req}$ by solving the inequality given by (10) in g, though of course this does not actually yield the protocol tuning solution itself. If no positive real solutions for g exist, then, the EA is not expected to find any solution of suitable quality in finite time

$$g \geq \left\lceil \frac{\alpha}{E_{req} - \gamma} - \beta \right\rceil. \tag{10}$$

This technique allows sensornet designers to estimate in advance the computational overhead implied in finding protocol tuning solutions of a given quality. A related technique can be applied to estimate the rate at which solution quality improves with respect to MOEA generation. Equation (11) gives the derivative of the fitted curve given by (9) in terms of generation g

$$\frac{dE}{dg} = -\frac{\alpha}{(g+\beta)^2}. \tag{11}$$

As the MOEA progresses from generation to generation, the rate of change of solution quality can be estimated. The magnitude of this rate of change is relatively large at the start of the MOEA execution, but $(dE/dg) \to 0$ as $g \to \infty$ because E is monotonically nonincreasing. At some point, the rate of solution quality improvement will be sufficiently small that any further improvement is not significant, and hence the protocol tuning effort can be terminated.

A computational cost is incurred when assessing each generation of the MOEA. A technical discussion of the cost incurred for the SPEA2 and TA algorithms is given in Sections V-B and VI-B, respectively. Provided that the computational overhead of fitting a curve of form (9) is small compared to the overhead $arst$ of assessing a MOEA generation, it is feasible to implement this technique online during MOEA execution.

Under this approach the MOEA assesses the solution quality derivative given by (11) at the end of each generation; if the derivative magnitude is sufficiently large, another MOEA generation executes, otherwise the process terminates at a point where no further significant solution quality improvement is likely within reasonable time.

An interesting hybrid approach would be to extend the MOEAs such that each candidate solution retained at each generation is taken as the center of a small region of the parameter space which is sampled by FD methods as described in Section IV. The interpolation implicit in this approach would allow the MOEA to consider alternative candidate solutions that are close to, but potentially better than, those created explicitly by the mutation and crossover processes. This hybrid approach would of course increase experimental overhead with increased numbers of fitness function evaluations, and auxiliary calculations implied by the MOEA itself and the model fitting.

E. *Comparing Evolutionary and Nonevolutionary Approaches*

We see that both SPEA2 and TA yield high quality near-optimal solutions within a small number of evolutionary generations, with diminishing returns on further investments of experimental effort in the monotonically nondecreasing solution quality. Consider the experimental cost of the FD approach given in Section IV-C, and that of the SPEA2 and TA approaches given in Section V-B and VI-B, respectively. As the cost associated with the EAs is linear in generation count, limiting the number of generations also limits the cost.

Significantly, the SPEA2 and TA costs are independent of the number of controlled factors, unlike the FD cost. It follows that the merit of MOEAs becomes more apparent as the number of controlled factors grows and the cost of FD experiments becomes prohibitively large. It is the responsibility of the experimenter to decide whether it is appropriate to invest in the high monolithic cost of the FD approach, or the potentially lower but growing costs of the MOEA approaches. Provided that convergence on an acceptable solution occurs within a certain number of generations, the overhead of MOEA approaches is lesser; in Section VII-C, we calculate the critical point at which the MOEA approaches cease to offer cost benefits.

The FD approach samples the factor–response space evenly, whereas MOEAs focus computation resources on promising candidate solutions. The MOEAs tend to gravitate toward a single solution, whereas the FD approach provides an overview of the complete problem space which may encompass multiple good solutions, but if the experimenter requires only a single good solution, this increased diversity is less important than the experimental cost.

We conclude that both evolutionary and nonevolutionary approaches offer benefit to the sensornet designer, but serve different purposes. The designer might usefully apply the FD approach to summarize the factor–response relationship to narrow the search space to regions in which good solutions are known to reside, then, apply evolutionary approaches to navigate any nonlinear regions within this narrowed search space to obtain better near-optimal solutions.

VIII. Conclusion

In this section, we draw upon the content of Sections III–VII to present summarized conclusions against the research objectives defined in Section I.

Obj 1: *Obtain near-optimal solutions to the sensornet protocol tuning problem using evolutionary and principled search approaches*.

The FD, SPEA2, and TA approaches described in Sections IV–VI all achieve results close to the theoretical optimum for the protocol optimization problem described in Section III.

The IGF protocol considered in this paper represents the state of the art; it is lightweight, geography-aware, stateless, and contemporary. In contrast, the TBG protocol is simplistic and potentially highly inefficient. Despite these qualitative differences, poor configurations of the state-of-the-art protocol were substantially outperformed by good configurations of the less sophisticated protocol in our experiments.

Our results demonstrate that even if the network designer selects a state-of-the-art protocol such as IGF, selecting an appropriate configuration remains an open question. This important issue is regrettably ignored in many papers which describe new protocols, despite the potential impact on network applications employing these protocols.

Obj 2: *Compare solution quality attainable by evolutionary and principled search approaches*.

Results presented in Section VII show that MOEAs can significantly outperform a simple FD experimental approach when tuning sensornet protocols against multiple objectives, producing higher quality solutions with lower experimental overhead. This is the first study in which sensornet protocol optimization has been explicitly formulated as a multi-objective problem and state-of-the-art multi-objective EAs applied in its solution. The TA algorithm outperformed the SPEA2 algorithm, at each generation and in the final evolved solution, for each protocol considered in this paper.

Obj 3: *Consider the relationship between solution quality and experimental overhead for the optimization problem*.

Results presented in Section VII illustrate that the experimental cost of FD experiment suites is fixed and independent of solution quality, whereas EA approaches allow experimenters to manage the tradeoff between experimental cost and solution quality. Co-evolution of protocol designs and protocol tunings offers further scope for improved performance in future work.

References

[1] J. Tate, B. Woolford-Lim, I. Bate, and X. Yao, "Comparing design of experiments and evolutionary approaches to multi-objective optimisation of sensornet protocols," in *Proc. 11th IEEE Congr. Evol. Comput.*, Trondheim, Norway, May 18–21, 2009, pp. 1137–1144.

[2] X. Jiang, J. Taneja, J. Ortiz, A. Tavakoli, P. Dutta, J. Jeong, D. Culler, and P. Levis, "An architecture for energy management in wireless sensor networks," *ACM SIGBED Rev.*, vol. 4, no. 3, pp. 31–36, Jul. 2007.

[3] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. 11th IEEE Int. Conf. Acoust., Speech, Signal Process.*, Salt Lake City, UT, May 7–11, 2001, pp. 2033–2036.

[4] C. Toh, P. Mahonen, and M. Uusitalo, "Standardization efforts and future research issues for wireless sensors and mobile ad hoc networks," *IEICE Trans. Commun.*, vol. 88, no. 9, pp. 3500–3507, Sep. 2005.

[5] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proc. 5th ACM/IEEE Int. Conf. Mobile Comput. Netw.*, Seattle, WA, Aug. 15–19, 1999, pp. 151–162.

[6] T. He, B. Blum, Q. Cao, J. Stankovic, S. Son, and T. Abdelzaher, "Robust and timely communication over highly dynamic sensor networks," *Real-Time Syst.*, vol. 37, no. 3, pp. 261–289, Dec. 2007.

[7] A. Baker, S. Dixon, F. Drabble, J. Gibbings, A. Lewkowicz, D. Moffat, and R. Shaw, *The Systematic Experiment*, J. Gibbings, Ed. Cambridge, U.K.: Cambridge Univ. Press, 1986.

[8] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," in *Proc. 4th Conf. Evol. Methods Des., Optim. Control With Appl. Ind. Probl.*, Athens, Greece, Sep. 19–21, 2001, pp. 95–100.

[9] K. Praditwong and X. Yao, "A new multi-objective evolutionary optimisation algorithm: The Two-Archive algorithm," in *Proc. 3rd Int. Conf. Comput. Intell. Security*, Guangzhou, China, Dec. 15–19, 2007, pp. 95–104.

[10] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[11] R. Govindan, J. Hellerstein, W. Hong, S. Madden, M. Franklin, and S. Shenker, "The sensor network as a database," Comput. Sci. Dept., Univ. Southern California, Los Angeles, CA, Tech. Rep. 02-771, Sep. 2002.

[12] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. Hoboken, NJ: Wiley, 2005.

[13] C. Ee, R. Fonseca, S. Kim, D. Moon, A. Tavakoli, D. Culler, S. Shenker, and I. Stoica, "A modular network layer for sensornets," in *Proc. 7th USENIX Symp. Oper. Syst. Des. Implementation*, Seattle, WA, Nov. 6–8, 2006, pp. 18–27.

[14] G. Pottie and W. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, May 2000.

[15] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava, "Energy aware wireless microsensor networks," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 40–50, Mar. 2002.

[16] V. Kawadia and P. Kumar, "Power control and clustering in ad hoc networks," in *Proc. 22nd IEEE Conf. Comput. Commun.*, San Francisco, CA, Apr. 1–3, 2003, pp. 459–469.

[17] R. Tridgell, "The CCIR radio-paging code no. 1: A new world standard," in *Proc. 42nd IEEE Veh. Technol. Conf.*, San Diego, CA, May 23–26, 1982, vol. 32, pp. 403–406.

[18] M. Aly, A. Gopalan, and A. Youssef, "Load-balancing query hotspots for next-generation sensornets," in *Proc. 50th IEEE Global Telecommun. Conf.*, Washington, DC, Nov. 26–30, 2007, pp. 775–779.

[19] A. Fraboulet, G. Chelius, and E. Fleury, "Worldsens: Development and prototyping tools for application specific wireless sensor networks," in *Proc. 6th Int. Symp. Inf. Process. Sens. Netw.*, Cambridge, MA, Apr. 25–27, 2007, pp. 176–185.

[20] J. Tate and I. Bate, "Sensornet protocol tuning using principled engineering methods," *Comput. J.*, vol. 53, no. 7, pp. 991–1019, Sep. 2010.

[21] J. Ploennigs, P. Buchholz, M. Neugebauer, and K. Kabitzsch, "Automated modeling and analysis of CSMA-type access schemes for building automation networks," *IEEE Trans. Ind. Informat.*, vol. 2, no. 2, pp. 103–111, May 2006.

[22] P. Huang, D. Estrin, and J. Heidemann, "Enabling large-scale simulations: Selective abstraction approach to the study of multicast protocols," in *Proc. 6th Int. Symp. Model., Anal. Simul. Comput. Telecommun. Syst.*, College Park, MD, Jul. 24–28, 1998, pp. 241–248.

[23] D. Rao and P. Wilsey, "Modeling and simulation of active networks," in *Proc. 34th Annu. Simul. Symp.*, Seattle, WA, Apr. 22–26, 2001, pp. 177–184.

[24] A. Mohan, H. Wei, D. Gay, P. Buonadonna, and A. Mainwaring, "End-to-end performance characterization of sensornet multi-hop routing," in *Proc. 2nd IEEE Int. Conf. Pervasive Serv.*, Santorini, Greece, Jul. 11–14, 2005, pp. 27–36.

[25] M. Totaro and D. Perkins, "Using statistical design of experiments for analyzing mobile ad hoc networks," in *Proc. 8th Int. Symp. Model. Anal. Simul. Wireless Mobile Syst.*, Montreal, QC, Canada, Oct. 10–13, 2005, pp. 159–168.

[26] D. Yupho and J. Kabara, "The effect of physical topology on wireless sensor network lifetime," *J. Netw.*, vol. 2, no. 5, pp. 14–23, Sep. 2007.

[27] E. Alba, B. Dorronsoro, F. Luna, A. Nebro, P. Bouvry, and L. Hogie, "A cellular multi-objective genetic algorithm for optimal broadcasting strategy in metropolitan MANETs," *Comput. Commun.*, vol. 30, no. 4, pp. 685–697, Feb. 2007.

[28] C. Coello, G. Lamont, and D. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems,* 2nd ed. New York: Springer-Verlag, Sep. 2007, ser. Genetic Algorithms and Evolutionary Computation.

[29] V. Guliashki, H. Toshev, and C. Korsemov, "Survey of evolutionary algorithms used in multiobjective optimization," *Probl. Eng. Cybern. Robot.*, vol. 60, pp. 42–54, 2009.

[30] D. Jourdan and O. de Weck, "Layout optimization for a wireless sensor network using a multi-objective genetic algorithm," in *Proc. 60th IEEE Semiannu. Veh. Technol. Conf.*, Milan, Italy, May 17–19, 2004, pp. 2466–2470.

[31] F. Quintão, F. Nakamura, and G. Mateus, "Evolutionary algorithm for the dynamic coverage problem applied to wireless sensor networks design," in *Proc. 7th IEEE Congr. Evol. Comput.*, Edinburgh, U.K., Sep. 2–4, 2005, pp. 1589–1596.

[32] C. Lai, C. Ting, and R. Ko, "An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications," in *Proc. 9th IEEE Congr. Evol. Comput.*, Singapore, Sep. 25–28, 2007, pp. 3531–3538.

[33] X. Hu, Y. Yu, S. Chung, Y. Shi, and X. Luo, "Hybrid genetic algorithm using a forward encoding scheme for lifetime maximization of wireless sensor networks," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 766–781, Oct. 2010.

[34] G. Molina, E. Alba, and E. Talbi, "Optimal sensor network layout using multi-objective metaheuristics," *J. Universal Comput. Sci.*, vol. 14, no. 15, pp. 2549–2565, Jan. 2008.

[35] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Proc. 6th Int. Conf. Parallel Probl. Solving From Nature*, London, U.K., Sep. 18–20, 2000, pp. 849–858.

[36] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. 10th Int. Conf. Parallel Probl. Solving From Nature*, Birmingham, U.K., Sep. 18–22, 2004, pp. 832–842.

[37] K. Chaudhuri and D. Dasgupta, "Multi-objective evolutionary algorithms to solve coverage and lifetime optimization problem in Wireless Sensor Networks," in *Proc. 1st Conf. Swarm, Evol., Memetic Comput.*, Chennai, India, Dec. 16–18, 2010, pp. 514–522.

[38] H. Seo, S. Oh, and C. Lee, "Evolutionary genetic algorithm for efficient clustering of wireless sensor networks," in *Proc. 6th IEEE Conf. Consum. Commun. Netw.*, Las Vegas, NV, Jan. 11–13, 2009, pp. 258–262.

[39] E. Yang, N. Haridas, A. El-Rayis, A. Erdogan, and T. Arslan, "Multiobjective optimal design of MEMS-based reconfigurable and evolvable sensor networks for space applications," in *Proc. 2nd NASA/ESA Conf. Adapt. Hardw. Syst.*, Aug. 5–8, 2007, pp. 27–34.

[40] C. Brandolese and L. Rucco, "A genetic approach for WSN lifetime maximisation through dynamic linking and management," in *Proc. 7th ACM Workshop Perform. Eval. Wireless Ad Hoc, Sens., Ubiquitous Netw.*, Bodrum, Turkey, Oct. 17–21, 2010, pp. 99–100.

[41] D. Corne and J. Knowles, "Techniques for highly multiobjective optimisation: Some nondominated points are better than others," in *Proc. 9th Annu. Conf. Genetic Evol. Comput.*, London, U.K., Jul. 7–11, 2007, pp. 773–780.

[42] A. McInnes, "Using CSP to model and analyze TinyOS applications," in *Proc. 16th Int. Conf. Eng. Comput. Based Syst.*, San Francisco, CA, Apr. 14–15, 2009, pp. 79–88.

[43] P. Ölveczky and S. Thorvaldsen, "Formal modeling and analysis of the OGDC wireless sensor network algorithm in Real-Time Maude," in *Proc. 9th Int. Conf. Formal Methods Open Object-Based Distrib. Syst.*, Paphos, Cyprus, Jun. 6–8, 2007, pp. 122–140.

[44] M. Katelman, J. Meseguer, and J. Hou, "Redesign of the LMST wireless sensor protocol through formal modeling and statistical model checking," in *Proc. 10th Int. Conf. Formal Methods Open Object-Based Distrib. Syst.*, Oslo, Norway, Jun. 4–6, 2008, pp. 150–169.

[45] J. Tate and I. Bate, "YASS: A scalable sensornet simulator for large scale experimentation," in *Proc. 31st Communicating Process Architectures Conf.*, York, U.K., Sep. 7–10, 2008, pp. 411–430.

[46] H. Friis, "A note on a simple transmission formula," *Proc. IRE*, vol. 34, no. 5, pp. 254–256, May 1946.

[47] Z. Haas, J. Halpern, and L. Li, "Gossip-based ad hoc routing," *IEEE/ACM Trans. Netw.*, vol. 14, no. 3, pp. 479–491, Jun. 2006.

[48] M. Rudafshani and S. Datta, "Localization in wireless sensor networks," in *Proc. 6th Int. Conf. Inf. Process. Sens. Netw.*, Cambridge, MA, Apr. 25–27, 2007, pp. 51–60.

[49] D. Knuth, "Two notes on notation," *Amer. Math. Monthly*, vol. 99, no. 5, pp. 403–422, May 1992.

[50] A. Atkinson and A. Donev, *Optimum Experimental Designs*. Oxford, U.K.: Oxford Univ. Press, 1992, ser. Statistical Science.

[51] G. Box, J. Hunter, and W. Hunter, *Statistics for Experimenters: Design, Innovation, and Discovery,* 2nd ed. Hoboken, NJ: Wiley, 2005.

[52] B. Woolford-Lim, J. Tate, I. Bate, and X. Yao, Supplementary experimental results for linear interaction models, Jan. 2010. [Online]. Available: http://www.cs.york.ac.uk/rts/projects/smc/jan2010.pdf

[53] I. Fodor, "A survey of dimension reduction techniques," Lawrence Livermore Nat. Lab., Livermore, CA, Tech. Rep. UCRL-ID-148494, Jun. 2002.

[54] E. Zitzler and L. Thiele, "An evolutionary approach for multiobjective optimization: The strength Pareto approach," Comput. Eng. Netw. Lab., ETH Zürich, Zürich, Switzerland, TIK Rep. 43, May 1998.

[55] V. Khare, X. Yao, and K. Deb, "Performance scaling of multi-objective evolutionary algorithms," in *Proc. 2nd Evol. Multi-Criterion Optim. Conf.*, Faro, Portugal, Apr. 8–11, 2003, pp. 376–390.

[56] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, pp. 115–148, 1995.

[57] D. White and S. Poulding, "A rigorous evaluation of crossover and mutation in genetic programming," in *Proc. 12th Eur. Conf. Genetic Program.*, Tübingen, Germany, Apr. 15–17, 2009, pp. 220–231.

[58] MATLAB Curve Fitting Toolbox, MathWorks, Inc., Natwick, MA, 2009

Jonathan Tate (M'07) received the B.Sc. and M.Sc. degrees by research from the University of Durham, and the Ph.D. degree from the University of York, York, U.K., all in computer science.

He has worked in industry developing network analysis tools, and currently works for a financial software consultancy. His research interests include sensornet design optimization, energy-efficient network protocols, homeostatic system management, and software process quality measurement.

Dr. Tate is a member of the BCS.

Benjamin Woolford-Lim (S'09) received the B.Sc. degree in mathematics with physics from the University of Warwick, Coventry, U.K. and the M.Sc. degree in computer science and another in advanced computer science from the University of Birmingham, Birmingham, U.K., where he is currently working toward the Ph.D. degree, supervised by Prof. X. Yao.

His research interests are in evolutionary computation, specifically the relationships between multi-objective problems, algorithms, and operators.

Iain Bate received the D.Phil. from University of York, York, U.K., in 1999.

He is a Senior Lecturer in Real-Time Systems at the University of York. His research interests include scheduling and timing analysis, design and analysis of safety-critical systems, and engineering of complex systems of systems including sensornets. He is the editor-in-chief of the *Journal of Systems Architecture*.

Dr. Bate is a frequent member of program committees for distinguished international conferences.

Xin Yao (M'91–SM'96–F'03) received the B.Sc. degree from the University of Science and Technology of China (USTC), Hefei, China, in 1982, the M.Sc. degree from the North China Institute of Computing Technology, Beijing, China, in 1985, and the Ph.D. degree from USTC in 1990.

He was an Associate Lecturer and Lecturer with USTC in 1985–1990, a Postdoctoral Fellow at the Australian National University, Canberra, Australia and CSIRO, Melbourne, Australia, in 1990–1992, and a Lecturer, Senior Lecturer and Associate Professor at UNSW at ADFA, Canberra, Australia, in 1992–1999. Since April 1999, he has been a Professor (Chair) of computer science at the University of Birmingham, Birmingham, U.K., where he is currently the Director of the Center of Excellence for Research in Computational Intelligence and Applications. He is also a Distinguished Visiting Professor (Grand Master Professorship) of USTC. His major research interests include evolutionary computation and neural network ensembles. He has more than 350 refereed publications.

Dr. Yao is a Distinguished Lecturer of the IEEE Computational Intelligence Society, an invited keynote/plenary speaker of 60 international conferences, and a former (2003–2008) Editor-in-Chief of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION. He was a recipient of the 2001 IEEE Donald G. Fink Prize Paper Award, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding 2008 Paper Award, IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding 2009 Paper Award, and several other best paper awards.