Evolutionary Multi-Path Routing for Network Lifetime and Robustness in Wireless Sensor Networks

Alma A. M. Rahat[a,*], Richard M. Everson[a], Jonathan E. Fieldsend[a]

[a]*Department of Computer Science, University of Exeter, United Kingdom.*

Abstract

Wireless sensor networks frequently use multi-path routing schemes between nodes and a base station. Multi-path routing confers additional robustness against link failure, but in battery-powered networks it is desirable to choose paths which maximise the overall network lifetime — the time at which a battery is first exhausted. We introduce multi-objective evolutionary algorithms to find the routings which approximate the optimal trade-off between network lifetime and robustness. A novel measure of network robustness, the *fragility*, is introduced. We show that the distribution of traffic between paths in a given multi-path scheme that optimises lifetime or fragility may be found by solving the appropriate linear program. A multi-objective evolutionary algorithm is used to solve the combinatorial optimisation problem of choosing routings and traffic distributions that give the optimal trade-off between network lifetime and robustness. Efficiency is achieved by pruning the search space using k-shortest paths, braided and edge disjoint paths. The method is demonstrated on synthetic networks and a real network deployed at the Victoria & Albert Museum, London. For these networks, using only two paths per node, we locate routings with lifetimes within 3% of those obtained with unlimited paths per node. In addition, routings which halve the network fragility are located. We also show that the evolutionary multi-path routing can achieve significant improvement in performance over a braided multi-path scheme.

Keywords: Evolutionary routing; robust multi-path routing; network reliability; maximum lifetime routing; multi-objective optimisation; wireless sensor mesh networks.

1. Introduction

This paper examines the use of evolutionary algorithms to find routings in low-power wireless sensor networks that simultaneously optimise the network lifetime and overall network robustness.

Wireless sensors are autonomous devices that measure environmental parameters, such as temperature and humidity. In sensor networks many such devices are distributed over a wide area. Generally these sensors periodically report data back to a central base station, often employing a mesh network topology, in which each device is a node, to extend the range of the network. They are widely used in remote monitoring applications due to ease of installation and the ability to monitor areas that are difficult to access. Inevitably, such applications require these sensors to be battery powered. In addition to powering the sensors themselves, transmission and reception costs are frequently major drains on the batteries. As such, it is important to choose paths from each sensor to the base station that preserve the life of the batteries. Using paths that on average use the least energy can be detrimental to a group of nodes that relay the most paths [1]. We therefore focus on optimising the lifetime of the node that first exhausts its battery; this is the *network lifetime* — the time when the network first needs manual intervention to change a battery [2, 3].

Unpredictable and dynamic radio environments, leading to occasional link failures, are inevitable in wireless sensor network (WSN) deployments. The traditional routing approaches deployed in generic wired or wireless networks to achieve network robustness may not be feasible,

[*] Corresponding author
Email address: a.a.m.rahat@exeter.ac.uk (Alma A. M. Rahat)

primarily due to energy, computational and storage limitations at sensor nodes [4, 5]. Many existing routing approaches in WSNs consider single-path routing schemes — a single path from each source node to the base station — due to their simplicity and efficient resource utilisation. In case of link failure the single-path routing can be re-planned or re-optimised and the network reconfigured accordingly (see for example [6]). An alternative, which allows receipt of at least partial information during link failure, is to use a multi-path routing scheme in which each source node uses a number of paths to send data to the base station. In such a scheme each node sends a proportion of its messages via each of the available paths; only one path is used for each individual message, so if a link fails the proportion of messages sent via the other routes will be received successfully. Such multi-path routing has been shown to be both fault tolerant through the use of alternative paths and energy efficient through load balancing [4, 7]. As we discuss in more detail in section 2.3, current measures of robustness generally consider the paths available to each source node, without accounting for the effect on network robustness that results from failure of a link that is used by paths from multiple source nodes. In this paper we therefore quantify the network robustness in terms of the maximum expected data loss (across the entire network) that would occur in the event of a link failure.

Although multi-path routing is beneficial for achieving network robustness, it is likely to have deleterious effects on battery life because it utilises additional links. We therefore propose evolutionary algorithms to locate routes that find routings which approximate the optimal trade-offs between network lifetime and robustness.

Simultaneously improving both network lifetime and robustness is pivotal for devising a successful multi-path routing scheme in WSNs. Current routing protocols treat this two-objective optimisation problem as a single objective problem. For instance, Yahya *et al.* define a composite weighted link cost combining energy, available buffer storage, and radio interference where the relative importance of the cost are controlled with weights [8]. A preferred path is constructed based on this link cost, and when the cost becomes expensive beyond a threshold, a new path is used to send data. A similar strategy is adopted in [9]. However, the trade-off between different possible routings is not explored. The optimal trade-off front, also known as the Pareto front, consists of the routings which are not *dominated* by any other routing [10]; that is, routings for which there is no other routing with better network lifetime and robustness. *Evolutionary algorithms* (EAs) are an efficient method of finding the Pareto front. They deploy a population of possible routings and are capable of evolving a set of solutions that well approximate the optimal trade-off front [10].

Most current EA-based multi-objective routing optimisation approaches consider single-path routing schemes to optimise various objectives: energy efficiency, network lifetime, latency, robustness, expected transmission count, etc. [11, 12, 13, 14]. Here we describe a framework for multi-path routing optimisation with two objectives: maximise network lifetime and maximise robustness, and estimate the optimal trade-off front. This approach can achieve solutions with network lifetimes close to the theoretical maximum network lifetime (when no constraint on the number of routes per node is imposed) as presented in [1], and a range of solutions representing various levels of robustness. The major contributions of this paper can be summarised as:

- We describe a hybrid evolutionary search procedure to approximate the optimal trade-off between network lifetime and network robustness.

- We introduce a novel robustness measure (the *fragility*) of multi-path routing schemes to quantitatively analyse and compare the robustness of different multi-path routing schemes. The fragility accounts for the effect of failure of a link shared between multiple source nodes.

- We show how the proportion of time for which each path should be used in a multi-path scheme may be determined by an appropriate linear program to optimise either network lifetime or robustness.

- Novel search space pruning methods, based on braided and edge disjoint paths, are used to speed the evolutionary search by restricting the search space to regions likely to contain good solutions.

- The proposed methods are illustrated in a real network deployed in Victoria & Albert Museum, London, UK, and successfully locate a wide range of robust multi-path routing schemes with long network lifetimes and greater robustness, surpassing the performance of single-path routing schemes.

The rest of the paper is structured as follows. In section 2 we describe our network model and the associated formulation of network lifetime and robustness. Section 3 describes the multi-objective problem to be optimised and in section 4 a hybrid evolutionary algorithm to solve it is presented. Search space pruning, key to the efficiency of the evolutionary algorithm, is discussed in section 4.1. The method is evaluated and compared with popular methods on synthetic and real networks in section 5. Related work is discussed in section 6. Finally, conclusions are drawn in section 7.

2. Network Model, Lifetime and Robustness

In this section, we present a model for WSNs with multi-path routing and formulate network lifetime and robustness as objectives to be optimised.

2.1. Network Model

We consider a communication protocol in which all nodes periodically (e.g., once every minute) send their sensed data to the base station, potentially by relaying a message through one or more nodes. Such data reporting periods are repeated throughout the network lifetime: the time before which a node first exhausts its battery. This scenario is most common in industrial applications, especially for constant monitoring of locations.

Once a connectivity map, showing which nodes may communicate with each other, has been established, routing is performed under the assumption that links are reliable. Generally, pairs of nodes are configured to use the most energy efficient settings that allow reliable communication. Usually energy efficient links correspond to high baud rate and low transmission power.

Note that we used very low power sensor nodes in this paper. As such the frequent *pinging* in connectivity discovery is prohibitively expensive. Therefore, the connectivity discovery process is only triggered in case of establishing the network for the first time or severe deterioration in performance. Furthermore, nodes are not capable of performing route calculations due to very limited computational resources. Thus routing calculation and decisions are performed at the mains powered base station in a centralised manner as configuring the sensors infrequently over the radio link is relatively cheap (for instance, in the real world implementation we consider here, each network configuration cycle costs approximately 0.1% of the total battery energy per node).

We deem the hardware to be reliable, and thus node failure is a rare event that necessitates replacement of the node. On the other hand, the radio environment is seldom constant and links may occasionally fail due to changing atmospheric conditions, the passage of people, radio interference, and so on. One mechanism to combat the intermittent failure of links is to provide more than one path from each node to the base station. Each node then splits its traffic between the available paths, sending a proportion of messages via each of the available paths; exactly one path is used on each data reporting cycle, rotating between the available paths. Thus if there is a failure on one path, messages sent via other paths will still be received, providing at least partial information. We call the proportion of time that a particular path is utilised the *active time share* for that path.

A WSN is represented as a network graph, $G = \{V, E\}$, where V is a set of N sensor nodes v_i plus a base station node v_B, and E is the set of edges, describing with which other nodes each node can communicate [15]. Figure 1 illustrates a multi-path routing in which there are two routes from node v_i to the base station v_B: $R_{i1} = \langle v_i, v_j, \dots, v_B \rangle$ and $R_{i2} = \langle v_i, v_k, \dots, v_B \rangle$. We denote by τ_{id} the active time share of path R_{id}, namely the proportion of messages sent by v_i via route R_{id}. Clearly, $\tau_{id} \geq 0$ for all i, d and $\sum_d \tau_{id} = 1, \forall v_i \in V$.

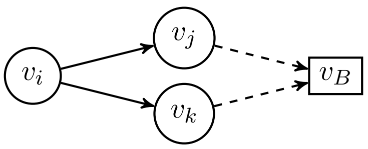


Figure 1: Two paths $R_{i1} = \langle v_i, v_j, \dots, v_B \rangle$ and $R_{i2} = \langle v_i, v_k, \dots, v_B \rangle$ from v_i to the base station v_B. In a multi-path routing scheme, R_{i1} is active for time share τ_{i1} and R_{i2} is active for time share τ_{i2}.

Hence, we define a D multi-path routing scheme $(\mathcal{R}, \mathcal{T})$ as a set of paths, where each node has D routes to the base station, and a set of associated time shares:

$$\mathcal{R} = \langle \{R_{1d}\}_{d=1}^{D}, \{R_{2d}\}_{d=1}^{D}, \dots, \{R_{Nd}\}_{d=1}^{D} \rangle, \tag{1}$$

$$\mathcal{T} = \langle \{\tau_{1d}\}_{j=1}^{D}, \{\tau_{2d}\}_{j=1}^{D}, \dots, \{\tau_{Nd}\}_{d=1}^{D} \rangle. \tag{2}$$

Practical memory constraints of current devices generally require that D, the number of paths for any node, is small, perhaps two or three.

2.2. *Network Lifetime*

The energy expended at a node due to transmitting its own data and relaying other nodes' data depletes charge in its battery and thus governs the lifetime of a node. An individual node's lifetime, and thus the network lifetime, then depends on the routing scheme as it dictates the total number of transmissions and receptions involving the node.

Let T_{kj} be the energy (charge) required at node v_k to send a message to v_j and let A_{pk} be the energy required to receive (and acknowledge) a message from v_p at v_k (Figure 2). Then in one reporting cycle, the energy cost at v_k associated with relaying messages in path $R_{id} = \langle v_i, \dots, v_p, v_k, v_j, \dots, v_B \rangle$ is

$$C_{id}^{k} = A_{pk} + T_{kj}. \tag{3}$$

At the originating node there is no reception cost, so that $C_{id}^{i} = T_{ir}$ where v_r is the node immediately downstream of v_i.

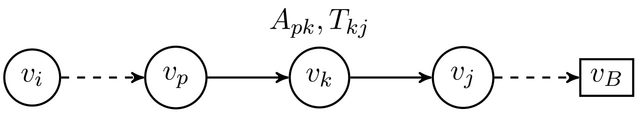


Figure 2: Notation for the d-th route for node v_i: $R_{id} = \langle v_i, \dots, v_p, v_k, v_j, \dots, v_B \rangle$. The energy costs to send data from v_p to v_k are T_{pk} at node v_p and A_{pk} at node v_k.

In order to calculate the battery lifetime remaining due to a multi-path routing scheme, we require additional intrinsic information about the nodes, namely the charge q_k remaining in the battery and the quiescent energy consumption per reporting cycle B_k due to constant microcontroller operation, sensor measurements, running an on-board display, etc. The life of the node v_k is therefore given by

$$L_k = \frac{q_k}{N_c(B_k + \sum_{R_{id} \in \mathcal{R}} C_{id}^{k} \tau_{id})} \tag{4}$$

where N_c is the number of reporting cycles per unit time (e.g. one year). Note that the energy expended at node v_k as a result of using route R_{id} depends on the associated time share τ_{id}; more frequent use naturally incurs a greater energy expense. We emphasise that $L_k \equiv L_k(\mathcal{R}, \mathcal{T})$ is a function of all the paths and the associated time shares which utilise v_k.

Our goal here is to maximise network lifetime, that is the time before any individual node requires its battery to be changed or recharged; thus we seek to maximise

$$L(\mathcal{R}, \mathcal{T}) = \min_{v_k \in V} L_k(\mathcal{R}, \mathcal{T}). \tag{5}$$

2.2.1. *Optimal Time Share*

Suppose that the routes $\mathcal{R}$ have been determined. Then the best distribution of time shares between multiple paths can be found as follows.

Inspecting (4), we note that rearranging this equation for a mathematical programming formulation will impose quadratic constraints with variables L_k and τ_{id} in product form. In order to reformulate this problem as a linear program, we consider the inverse lifetime

$$L'_k(\mathcal{T}) \equiv 1/L_k = \frac{N_c}{q_k} \left(B_k + \sum_{R_{id} \in \mathcal{R}} C^k_{id} \tau_{id} \right) \tag{6}$$

so that maximising $\min_{v_k \in V}\{L_k\}$ is equivalent to minimising $\max_{v_k \in V}\{L'_k\}$. Defining $L^{'*} = \max_{v_k \in V}\{L'_k\}$, the optimal time shares are then the solution to the linear program:

$$\min_{\mathcal{T}} L^{'*} \tag{7a}$$

subject to:

$$q_k L^{'*} - N_c \left(B_k + \sum_{R_{id} \in \mathcal{R}} C^k_{id} \tau_{id} \right) \geq 0 \ \ \forall k; \tag{7b}$$

$$\tau_{kd} \geq 0 \ \ \forall k, d; \tag{7c}$$

$$\sum_d \tau_{kd} = 1 \ \ \forall k. \tag{7d}$$

Note that in deriving the inequality constraints in (7b) we utilise the relationship that for all nodes in the system $L^{'*} \geq L'_k$ and replace L'_k by $L^{'*}$ in (6). We use this trick because *min-max* functions are non-smooth [16], and considering $L^{'*}$ instead of L'_k lets us formulate a convex linear program which can be solved in polynomial time [17].

Here, the inequalities (7b) (derived from (4) and (6)) ensure that the batteries have non-negative charge, while (7d) ensures that every node has paths allocated for all its messages.

Clearly, by solving the linear program (7) we can obtain a set of optimal time shares once a multi-path routing scheme is generated. This provides a means to efficiently calculate optimal time shares and thus compare routes $\mathcal{R}$ and $\mathcal{R}'$. In the evolutionary optimisation method that we present below, candidate paths are generated by the stochastic evolutionary mechanism; optimal network lifetimes for these are obtained by solving the linear program, obviating the need to perform a further stochastic search to locate optimal time shares.

2.3. *Robustness*

In WSNs, alongside network lifetime, it is important that data from sensors are delivered reliably. Data delivery may fail for various reasons: node failure, link failure, and congestion [18]. In modern deployments, node failure is a rare event due to recent advances in WSN technology; moreover, it is considered to be a critical event that requires immediate attention from the network administrator so that faulty nodes may be replaced or batteries can be changed. Additionally, the congestion problem at the Medium Access Control (MAC) layer may be addressed with Time Division Multiple Access (TDMA) techniques [19]. Furthermore, Forward Error Correction (FEC) methods can be deployed to improve link reliability in the Data Link Layer (DLL) [20]. However, even if these techniques are used, unpredictable link failures due to the changing radio environment may still occur [21, 22]. Robustness against occasional link failures is conferred by retransmission of packets for which an acknowledgement was not received, however, it is important to select reliable routes.

To select reliable routes and promote robustness against link failure, contemporary routing approaches consider optimising the packet delivery rate or the path failure probability [5]; see for example [23, 24, 25, 26, 27]. A common strategy is to send all messages via the most reliable path, switching to the next reliable path if the current path fails [7, 26]. Alternative approaches either distribute messages equally among paths, or in proportion to their expected reliability or residual energy [28, 8, 25, 29, 30]. These alternative strategies are preferred when downstream nodes are unable to signal link failure to source nodes and we employ these here. Generally, in assessing path reliability these protocols consider each source node in isolation and direct traffic according to the reliability of paths from that node. However, a link failure may simultaneously affect the data from multiple source nodes. This is because paths from different nodes may share edges, even if the multiple paths from each source node to the base station are disjoint. Assigning traffic on this basis may therefore overestimate the network's robustness, leaving it vulnerable to the failure of links carrying traffic from more than one source node.

We consider the expected message loss associated with the failure of a link in one of the paths used by a single node. We then characterise the*fragility* of the network as the maximum expected data loss in the event of a link failure anywhere in the network. Then maximising robustness is synonymous to minimising the fragility for a given multi-path routing scheme.

Let π_m be the failure probability of an edge $e_m \in E$. We assume that the edge failure probabilities are independent, so that the probability that a path $R_{id} \in \mathcal{R}$ fails in a unit time can be written as:

$$p_{id} = P(R_{id}\,\text{fails}) = 1 - \prod_{e_m \in R_{id}} (1-\pi_m) = \sum_{e_m \in R_{id}} \pi_m - \tilde{p}_{id}, \tag{8}$$

where $\tilde{p}_{id}$ represents all the higher order product terms. When the edge failure probabilities are small ($\pi_m \ll 1$), the higher order product terms (representing the probability of more than one edge failing simultaneously) are negligible, and p_{id} is thus well approximated by:

$$p_{id} \approx \sum_{e_m \in R_{id}} \pi_m. \tag{9}$$

In the Victoria & Albert network $\pi_m \approx 1\%$, and empirical results confirm that the approximation works well for edge failure probabilities $\pi_m \lesssim 20\%$.

If a node v_i sends U_i messages per unit time, then it uses R_{id} to send $U_i\tau_{id}$ of these messages and the expected loss of messages associated with a failure of a link in R_{id} is

$$U_i \tau_{id} p_{id}. \tag{10}$$

Now, in multi-path routing schemes the paths from a particular node may share edges. As a result, failure in a link in R_{id} may also be associated with loss in another path $R_{xy} \in \mathcal{R} \setminus R_{id}$ that shares edges with R_{id}. In this case the expected data loss associated with the failure of an edge is:

$$F_{id} = U_i \tau_{id} p_{id} + \sum_{\substack{R_{xy} \in \{\mathcal{R} \setminus R_{id}\} \\ v_x \in V}} U_x \tau_{xy} \left(1 - \prod_{e_n \in \{R_{xy} \cap R_{id}\}} (1-\pi_n) \right) \tag{11}$$

where the bracketed term is the probability that an edge common to R_{id} and another path R_{xy} fails.

In the following we use the first order approximation of path failure probabilities:

$$F_{id} = U_i \tau_{id} \sum_{e_m \in R_{id}} \pi_m + \sum_{\substack{R_{xy} \in \{\mathcal{R} \setminus R_{id}\} \\ v_x \in V}} U_x \tau_{xy} \sum_{e_n \in \{R_{xy} \cap R_{id}\}} \pi_n. \tag{12}$$

We call $F_i = \max_d F_{id}$ the *fragility* of the multi-path routes for v_i. To quantify the robustness of the entire network, we define the *fragility of the network* as the maximum expected data loss for any node:

$$F = \max_{v_i \in V} F_i = \max_{v_i \in V} \max_{d} F_{id}. \tag{13}$$

This path-based metric accounts for the failure probabilities of paths, the presence of shared edges, and the proportional usages of different paths in the network. It quantifies the maximum expected data loss in the WSN due to any link failure. Hence, optimising the fragility improves robustness of the whole network.

Clearly $F \equiv F(\mathcal{R}, \mathcal{T})$ depends on the choice of paths and time shares. To maximise the robustness of the network, we therefore minimise the fragility with respect to $\mathcal{R}$ and $\mathcal{T}$, that is, we seek to minimise the maximum expected data loss in the event of a link failure.

Our work on robustness is complementary to the work presented in [21, 22]: they devise ways to calculate statistical and empirical edge failure probabilities π_m and such information can be used to estimate robustness of multi-path routing schemes. If this information is not readily available, especially when the network is being established, it may be assumed that all edges are equally likely to fail and we set $\pi_m = \pi = const.$ for all m.

We next show that the optimum time shares $\mathcal{T}$ may be found by solving a linear program when the routes $\mathcal{R}$ are known, after which, in section 2.3.2, we illustrate the fragility measure for some simple intuitive cases. In section 3 we then present the multi-objective optimisation problem and an evolutionary approach to maximise both the network lifetime and robustness.

2.3.1. Optimal Time Share

In a similar manner to the network lifetime problem, we can devise a linear program to locate the active time shares $\mathcal{T}$ for a particular multi-path routing scheme $\mathcal{R}$ that minimise the network fragility (13).

With $F^*(\mathcal{T}) = \max_{R_{id} \in \mathcal{R}} F_{id}$, the linear program to minimise the network fragility can be described as follows:

$$\min_{\mathcal{T}} F^* \tag{14a}$$

subject to:

$$F^* \geq F_{id} \quad \forall R_{id} \in \mathcal{R}; \tag{14b}$$

$$\tau_{kd} \geq 0 \ \ \forall k, d; \tag{14c}$$

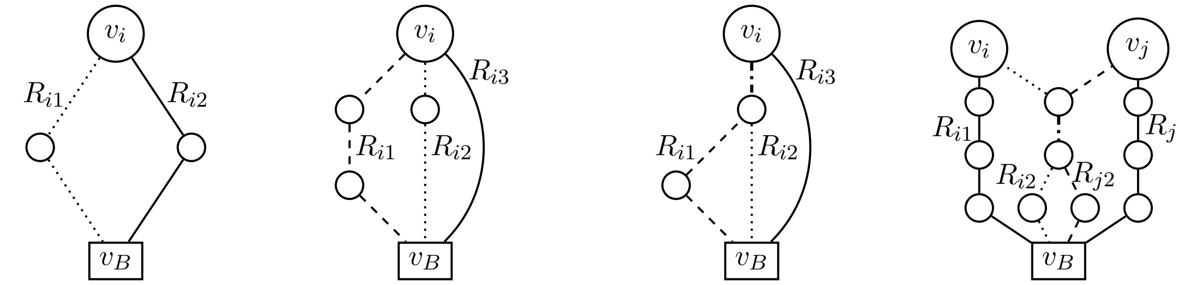
$$\sum_{d} \tau_{kd} = 1 \quad \forall k. \tag{14d}$$

Here too, setting $F^* = \max_{R_{id} \in \mathcal{R}} F_{id} \geq F_{id}$ enables us avoid the non-smoothness of the min-max problem (14a) and formulate the linear program. The set of constraints in (14b) indicates that any path specific fragility in (11) is less than or equal to the network fragility F^*. As before, the constraints in (14d) ensure that all data reporting cycles are used.

Solving this linear program results in a set of active time shares according to the best robustness for a routing scheme $\mathcal{R}$. This allows us to simply compare robustness of different multi-path routing schemes $\mathcal{R}$ with the knowledge that the active times for each route are optimal, and thus locate routing schemes with better robustness.

2.3.2. Case Studies

In this section we demonstrate the fragility in some simple scenarios, which are illustrated in Figure 3.



(a) Node v_i has two routes to the base station v_B: R_{i1} (dotted), and R_{i2} (solid) with associated time shares τ_{i1} and τ_{i2} respectively. With equal link failure probabilities, the routes are used equally, $\tau_{i1} = \tau_{i2} = 50\%$.

(b) Node v_i has three routes to the base station: R_{i1} (dashed) and R_{i2} (dotted) and R_{i3} (solid). With equal link failure probabilities π, the time shares are $\tau_{i1} = 18.2\%$, $\tau_{i2} = 27.3\%$ and $\tau_{i3} = 54.5\%$ and the fragility is $F_i \approx 0.545 U_i \pi$.

(c) Same configuration as Figure 3b except for the link (dash-dotted) shared between R_{i1} (dashed) and R_{i2} (dotted). With equal link failure probabilities the time shares are $\tau_{i1} = 12.5\%, \tau_{i2} = 25\%, \tau_{i3} = 62.5\%$. The fragility is $F_i = 0.625 U_i \pi$.

(d) v_i and v_j each have two routes with equal numbers of links, but share a single link (dash-dotted). With equal link failure probabilities π and traffic $U_i = U_j = U$, the optimal time shares are $\tau_{i1} = \tau_{j1} = 55.6\%$, $\tau_{i2} = \tau_{j2} = 44.4\%$. The fragility of v_i and v_j is $2.22 U \pi$.

Figure 3: Case studies illustrating the fragility (13).

Edge-disjoint routes between a pair of nodes

Edge-disjoint routes do not share any edges; however, they may share nodes. In such situations, the node fragility F_i is independent of the other nodes and the second term in (11) is zero.

As illustrated in Figure 3a, node v_i has two routes of equal length to the base station v_B: R_{i1} and R_{i2}. The active time shares for these routes are τ_{i1} and τ_{i2}. If the edge failure probabilities are equal for all the edges, by either solving the linear program or straightforward direct calculation, the minimum fragility occurs when $\tau_{i1} = \tau_{i2}$ and $F_{i1} = F_{i2}$, as would be expected from symmetry. Clearly, if the two routes have equal probability of failing then they should be used equally. Likewise, if the total failure probability for path R_{id} is p_{id} (c.f. (9)), then the minimum fragility occurs when the expected losses of each of the independent routes are equal, which occurs when:

$$\tau_{id} = \frac{1}{p_{id}} \left(\sum_{l} \frac{1}{p_{il}} \right)^{-1}. \tag{15}$$

When there are just two routes (Figure 3a) then (15) simplifies to $\tau_{i1} = p_{i2}/(p_{i1} + p_{i2}) = 1 - \tau_{i2}$. The fragility is $F_i = U_i \left(p_{i1}^{-1} + p_{i2}^{-1}\right)^{-1}$, which is clearly minimised when $p_{i1} = p_{i2}$ so that the risk of failure is borne equally by the two routes. The case for three routes with 3, 2, and 1 links respectively, each with equal failure probability π, is shown in Figure 3b. Here the routes are utilised in the proportions $\tau_{i1} = 18.2\%$, $\tau_{i2} = 27.3\%$ and $\tau_{i3} = 54.5\%$ so that the expected loss for each route F_{id} is equalised; at the optimum $F_i = 6 U_i \pi / 11 \approx 0.545 U_i \pi$.

Shared edges from a single node

Sometimes the paths from a single node to the base station will have to share edges. Clearly this makes the node more fragile because failure of one of the shared links will compromise more than one path, although there might be another independent route.

Figure 3c shows a node v_i with three routes in a similar configuration to Figure 3b, except that R_{i1} and R_{i2} share a link (shown dash-dotted). With equal link failure probabilities there is less advantage in using R_{i1} and R_{i2} than there was in the previous case. This is reflected in the time shares: $\tau_{i1} = 12.5\%, \tau_{i2} = 25\%$ and $\tau_{i3} = 62.5\%$ and the greater fragility $F_i = 0.625 U_i \pi$.

Multiple nodes sharing edges

When multiple routes share edges, failure of a single link may affect more than one node. In practice, since most nodes in a network generate data, there will be a large number of shared edges and the second term in (11) is significant.

As a simple illustration, Figure 3d shows two data reporting nodes v_i and v_j, which send their data to the base station each using two paths. These paths have equal numbers of links, so with equal edge failure probabilities, the fragility for each individual route is the same. In this illustration routes R_{i2} and R_{j2} share a single link. In this case, assuming the v_i and v_j generate equal traffic $U_i = U_j = U$, the optimal time shares are $\tau_{i1} = \tau_{j1} = 55.6\%$ and $\tau_{i2} = \tau_{j2} = 44.4\%$, so that traffic is directed away from the shared edge. Without the shared link the time shares for all routes would be 50% and the fragility $2U\pi$, whereas the shared link increases the best fragility of both nodes to $2.22U\pi$.

3. Multi-Objective Optimisation Problem

Our overall goal is to discover routes and time shares for a network that maximise the network lifetime and maximise the network robustness by minimising the fragility. These two objectives are expressed by (5) and (13) and may be collected together as a two-objective optimisation problem:

$$\text{maximise } f_1(\mathcal{R}, \mathcal{T}) = \min_{v_k \in V} L_k(\mathcal{R}, \mathcal{T}), \tag{16}$$

$$\text{minimise } f_2(\mathcal{R}, \mathcal{T}) = \max_{R_{id} \in \mathcal{R}} F_{id}(\mathcal{R}, \mathcal{T}). \tag{17}$$

These objectives might be augmented with others. For example, it might be important to maximise the lifetime of one or more particularly inaccessible nodes or to ensure the maximum robustness for other nodes. The optimisation procedure we describe below is easily generalised to these situations, but for simplicity we assume there are just these two objectives.

We emphasise that in some senses the time shares $\mathcal{T}$ are subsidiary to the routes $\mathcal{R}$, because if $\mathcal{R}$ is known then the $\mathcal{T}$ maximising the lifetime or minimising the fragility may be found by solving the appropriate linear program. For a routing scheme $\mathcal{R}$ we denote by $\mathcal{T}_L(\mathcal{R})$ and $\mathcal{T}_F(\mathcal{R})$ the solutions to the linear programs for lifetime and fragility respectively.

The network lifetime objective in (16) ensures that although energy efficient paths are selected, some load is distributed away from the heavily loaded nodes to prolong minimum lifetime. This may be in conflict, however, with the fragility objective in (17), where the load distribution that minimises fragility is purely dependent on the nature of the paths and shared edges between paths, without any regard to the traffic carried by each node and therefore the load imposed on its battery. As it will usually be impossible to optimise both objectives with a single routing, we therefore seek the set of solutions corresponding to the optimal trade-off between these objectives. In this case, there exists a set of solutions which are Pareto optimal; that is, there are no other feasible solutions available that improve performance on one objective, without a simultaneous decrease in the quality of the other objective (see, for example, [10]). With these Pareto optimal routing schemes on hand, the network engineer can make a reasoned decision about which to use.

The trade-off solutions to the multi-objective problems are more formally characterised in terms of *dominance*. In a multi-objective problem, one solution, $\mathbf{x} = (\mathcal{R}, \mathcal{T})$ is said to *dominate* another $\mathbf{x}' = (\mathcal{R}', \mathcal{T}')$ if it is wholly better than $\mathbf{x}'$. In terms of (16) and (17), a solution $\mathbf{x}$ dominates another $\mathbf{x}'$, written $\mathbf{x} \prec \mathbf{x}'$ iff:

$$\begin{aligned} f_1(\mathbf{x}) > f_1(\mathbf{x}') &\text{ and } f_2(\mathbf{x}) \leq f_2(\mathbf{x}') \\ \text{or } f_1(\mathbf{x}) \geq f_1(\mathbf{x}') &\text{ and } f_2(\mathbf{x}) < f_2(\mathbf{x}'). \end{aligned} \tag{18}$$

Two routing schemes are *mutually non-dominating* if neither dominates the other. The *Pareto set* is the maximal mutually non-dominating set of feasible solutions and the *Pareto front* is the image of the Pareto set in objective space; see e.g. [10].

Locating the Pareto set for this problem exactly would require searching all the possible multi-path routing schemes available in the network connectivity graph, G. However, the problem of counting the number of all possible simple paths in a graph is known to be $\#P$-complete; thus the corresponding problem of listing all simple paths is NP-complete [31, 32, 33]. Hence, there is no suitable algorithm that can solve this multi-objective problem in polynomial time and we therefore use a hybrid evolutionary approach to estimate the optimal trade-off between these conflicting objectives.

4. Hybrid Evolutionary Approach to Multi-Path Routing Optimisation

We use a straightforward elitist evolutionary algorithm as the basis of our search. The elitist EA starts with a random initial population of solutions and creates a mutually non-dominated archive from these solutions without any constraints on the number of solutions it may preserve (see, for instance, [34]). At each episode of evolution, two parent solutions from the archive are selected at random and evolutionary operators are applied to these solutions with the purpose of generating a better child solution. The evolutionary operators used here are *crossover* and *perturbation* (mutation). The crossover operator takes some parts of one of the selected individuals and combines them with the complementary parts of the other individual to construct a child. Essentially, this enables the algorithm to generate solutions that are not in the vicinity of the current members of the archive on the fitness landscape, and thus allows exploration. On the other hand, the perturbation strategy randomly makes small alterations to the child, and therefore exploits the fitness of that particular child. The child solution, however generated, is then compared with all the solutions in the archive and the mutually non-dominated solutions are retained. Therefore the archive remains a mutually non-dominating set throughout evolution. Thus, at any stage of the evolution, the archive represents the current best approximation of the Pareto set and can only move towards the true Pareto set. A more detailed account of the algorithm (see Algorithm 1) is given below.

Algorithm 1 Multi-objective multi-path routing optimisation using evolutionary algorithms.

```
Inputs
    $\mathcal{P}$ : Library of paths for each node
    $T$ : Number of iterations
    $s$ : Number of initially sampled routing schemes
    $\mu$ : Perturbation rate
    $c$ : Crossover rate

Steps
$A \leftarrow \text{InitialiseArchive}(\mathcal{P}, s)$                      ▷ Initialise random archive
for $i = 1 \to T$ do
    $\{\mathcal{R}^1, \mathcal{R}^2\} \leftarrow \text{Select}(A)$             ▷ Select two parent solutions
    $\mathcal{R}' \leftarrow \text{Crossover}(\mathcal{R}^1, \mathcal{R}^2, c)$
    $\mathcal{R}'' \leftarrow \text{Perturb}(\mathcal{R}', \mu, \mathcal{P})$                    ▷ Perturbation
    $A \leftarrow \text{NonDom}(A \cup \{(\mathcal{R}'', \mathcal{T}_L(\mathcal{R}'')), (\mathcal{R}'', \mathcal{T}_F(\mathcal{R}''))\})$
end for
$A \leftarrow \text{PostProcess}(A)$
return $A$                                          ▷ Approximation of the Pareto set
```

The algorithm is initialised by constructing s candidate routing schemes from a library of possible routes for each node, $\mathcal{P} = \langle \{R_{1m}\}_{m=1}^{M}, \{R_{2m}\}_{m=1}^{M}, \dots, \{R_{Nm}\}_{m=1}^{M} \rangle$, where M is the number of routes available for each node [1]. Preliminary experiments indicated that the algorithm is relatively insensitive to the exact number of initial solutions (s), as evolutionary mechanism may

[1] In practice some nodes may have fewer than M routes available.

eventually locate a range of routing schemes trading off between network lifetime and robustness. Nonetheless, from rapid optimisation perspective, $s = 100$ provides a good initial population for the networks used in the experiments.

To obtain an efficient search and combat the combinatorial explosion in the number of possible solutions with increasing network size, this library is built to contain routes that are likely to be good candidates for optimal solutions; details of its construction are given in section 4.1. In a D-path routing scheme, D paths for each node v_i are drawn at random from the available routes for v_i in $\mathcal{P}$, to form $\mathcal{R}$. Optimal time shares for $\mathcal{T}_L(\mathcal{R})$ and $\mathcal{T}_F(\mathcal{R})$ are found via linear programming. The candidate solutions $(\mathcal{R}, \mathcal{T}_L(\mathcal{R}))$ and $(\mathcal{R}, \mathcal{T}_F(\mathcal{R}))$ are added to the archive A. Solutions in A which are dominated by other solutions are removed from A, so that it remains a set of mutually non-dominating solutions.

At each generation of the evolutionary procedure the routes $\mathcal{R}^1$ and $\mathcal{R}^2$ corresponding to two solutions in A are selected at random from the elite archive A (line 3). These parent routing schemes are combined in a uniform crossover operation (line 4), in which a new routing scheme $\mathcal{R}'$ is constructed by selecting each path R_{id} in the paths for each node v_i from either R^1_{id} or R^2_{id} with probability c or $1 - c$ respectively, independently of other nodes and the paths for v_i. The new routing scheme $\mathcal{R}'$ is then perturbed by choosing a number of paths in the solution to alter based on the perturbation rate μ, and then replacing these from $\mathcal{P}$ at random (line 5). During the evolutionary optimisation we used $c = \mu = 0.1$ for the crossover and perturbation rates as these values allowed good convergence rates. Finally, having evaluated the optimal time shares and the corresponding objectives, if either of the newly constructed routings $(\mathcal{R}'', \mathcal{T}_L(\mathcal{R}''))$ or $(\mathcal{R}'', \mathcal{T}_F(\mathcal{R}''))$ is not dominated by any of the solutions in A, then it is added to A and any solutions in A that are dominated by these are removed from A. In this way non-dominated routing schemes are retained in the archive and the corresponding objectives can only approach the true Pareto front. The evolutionary process continues for a fixed number of generations, although another termination condition, such as a specified minimum dominated hypervolume [35] may be employed.

During the evolutionary process, we consider only the extremal time shares $\mathcal{T}_L(\mathcal{R}'')$ and $\mathcal{T}_F(\mathcal{R}'')$ for a particular candidate solution $\mathcal{R}''$ by solving appropriate linear programs. However, there may be many intermediate time shares between $\mathcal{T}_L(\mathcal{R}'')$ and $\mathcal{T}_F(\mathcal{R}'')$ that trade-off between network lifetime and robustness. Evaluating many potential intermediate time shares during evolution is computationally expensive. For completeness, we therefore consider the intermediate time shares for the solutions from A after the evolutionary optimisation is complete. This step has little impact on the overall run time of the algorithm, but may help improving the final approximation of the Pareto set. Full details are given in Appendix A.

4.1. Search Space Pruning

As noted above, the multi-objective optimisation problem is a combinatorial optimisation problem with, for practical WSNs, a vast number of potential solutions. The number of possible multi-path routing schemes depends on the number of available routes for each node in the network and the number of routes allowed per node. For instance, let the number of available loopless paths from v_i to v_B be a_i. If the number of paths per node is D, then the number of possible multi-path routing schemes is $(\prod_i a_i)^D$. Hence, it is crucial for practical implementations that we obtain an efficient algorithm by sensibly pruning the search space, while retaining important potential solutions, rather than considering solutions from the whole search space. In this section, we describe a number of methods of pruning the search space: *k-shortest path pruning*, *braided* and *edge-disjoint path pruning* are used to construct libraries of potential paths for each node $\mathcal{P}_{SP}, \mathcal{P}_{braid}$, and $\mathcal{P}_{edge}$ from the connectivity graph, G; we also reduce G to a new a graph G' using *max-min pruning* and construct additional libraries $\mathcal{P}'_{SP}, \mathcal{P}'_{braid}$, and $\mathcal{P}'_{edge}$ from it using k-shortest path pruning, braided and edge-disjoint path pruning. The evolutionary algorithm (Algorithm 1) then selects candidate routes from, $\mathcal{P} = \mathcal{P}_{SP} \cup \mathcal{P}_{braid} \cup \mathcal{P}_{edge} \cup \mathcal{P}'_{SP} \cup \mathcal{P}'_{braid} \cup \mathcal{P}'_{edge}$, where the union is performed separately over the paths for each node.

4.1.1. *k-shortest Path Pruning*

The k-shortest paths library reflects the intuition that short paths to the base station are most likely to be energy efficient. We therefore construct a library $\mathcal{P}_{SP}$ from the k shortest paths from each node to the base station. Several shortest path routes are available for each node because, if each node were to utilise its shortest path, nodes that occur in many of the shortest paths would be disproportionately burdened.

Algorithms for discovering the k shortest paths between a source and all the nodes in a weighted graph are well known and can be computed in $O(|E|+|V|\log|V|+k|V|)$ time [36, 37, 38]. However, as we noted above, the energy costs in this problem are associated with the nodes themselves rather than with the edges; see (3). We therefore weight the edges in the network graph to associate the energy cost at the nodes with the edges connecting them. Consider the nodes v_i and v_j. We define the weight of the edge between them as:

$$w_{ij} = \frac{T_{ij}}{q_i} + \frac{A_{ij}}{q_j}. \tag{19}$$

It is expected that if $q_i = q_j$, then $w_{ij} = w_{ji}$. This edge weighting models the fact that a high transmission cost can be borne by nodes with a high battery charge, but transmission is relatively expensive for nodes with low battery charge because each transmission will make a larger fractional depletion of the charge. Likewise, if a node is connected to mains power then transmissions are free, which is modelled by setting $q_i \to \infty$. We call the cost of a routing scheme calculated using the weights w_{ij} the *composite cost*.

We construct $\mathcal{P}_{SP}$ from the k shortest paths with respect to the composite cost using a straightforward modification of Eppstein's algorithm [37] to produce only simple or loopless paths.

4.1.2. *Braided and Edge-Disjoint Path Pruning*

In the interest of retaining potential good routes from the perspective of robustness, we build two additional path libraries: $\mathcal{P}_{braid}$, containing braided paths and $\mathcal{P}_{edge}$, containing edge-disjoint paths, which have been shown to be highly resilient and energy efficient [7].

Ganesan *et al.* [7] describe braided paths as partially disjoint paths, i.e. paths in a braid are allowed to share nodes and edges as long as they differ in some edges or nodes. These braided paths are based on a primary path, which we take to be the shortest path from a node to the base station considering the composite cost as edge weights; these primary paths are found during the construction of $\mathcal{P}_{SP}$ or can be found using Dijkstra's algorithm [39].

Two particular types of braided paths are presented by Ganesan *et al.*: idealised and localised braids. In idealised braids, two paths must differ in at least one node; while in localised braids two paths are allowed to share nodes as long as they have different edges to and from at least one shared node. Therefore to produce idealised braids, we remove one node at a time from the primary path from the network graph, and calculate new primary path. Similarly, for localised braids, we remove only the edges connecting a node in the primary path, and generate the primary path on the new graph. This is repeated for each node and associated edges in the primary path. Hence, the number of braided paths depends on the length of the primary path.

To generate edge-disjoint paths, starting from the primary path, we remove edges of the 1-st to m-th path from the network and calculate the next shortest path, i.e. the $(m+1)$-th completely edge-disjoint path, based on the composite edge costs. We repeat this process to construct $\mathcal{P}_{edge}$ with k edge-disjoint paths. In a partially connected network, as is usually the case in WSNs, a node may be completely disconnected from the network by removing a small number of edges, and as a consequence may have very few edge-disjoint paths.

4.1.3. *Max-Min Lifetime Pruning*

In order to generate candidate routes that tend to prolong the lifetime of the network we consider routing in the situation where, rather than just D routes, each node may use an arbitrary number of routes to the base station. Chang *et al.* [1] have shown that in this case the distribution of traffic along the various links that maximises the network lifetime can be found by solving a linear program (in polynomial time). The solution to this problem specifies the edge utilisation u_{ij}, namely the traffic carried by each edge, in the connectivity graph G. We generate a reduced graph $G' \subset G = (V, E)$, by deleting from G all the edges e_{ij} for which $u_{ij} = 0$; for full details of the construction of G' see [40]. We then use G' as the basis for generating libraries $\mathcal{P}'_{SP}, \mathcal{P}'_{braid}$ and $\mathcal{P}_{edge}$ of k-shortest, braided and edge-disjoint paths. The use of G' prunes the size of the search space, but retains routes that are good for prolonging the minimum lifetime.

5. Evaluation

In this section, we evaluate the proposed method in synthetic and real networks. In each case, we compare with the theoretical limit in network lifetime, obtained by solving the linear program devised by Chang *et al.* [1]. We emphasise that this best possible network lifetime can only be achieved if an arbitrarily large number of paths is available. In the multi-path schemes considered here nodes are limited to a small number (one, two or three) of paths. Nonetheless, we show that the method can obtain multi-path routing schemes with a network lifetime within 3% of the theoretical upper bound, while discovering a range of other routing schemes representing various levels of trade-off between network lifetime and robustness.

We also present a comparison with Ganesan *et al.'s* [7] braided path method, which is popular for fault tolerance and energy efficiency (see section 5.3). In this case, the braided path solution is always dominated by the solutions discovered through evolutionary routing.

5.1. *Synthetic Network Evaluation*

We first illustrate our multi-path routing algorithms on randomly generated synthetic networks. In these networks nodes were distributed uniformly at random in a rectangular area, and an edge defined between each node and its three nearest neighbours. This process results in networks in which every node is attached to at least three others, however as the nearest neighbour relation is not symmetric, some nodes may be connected to more than three neighbours. Transmission and reception costs associated with each link were selected randomly from 5 configurations. The most energy required for a single transmission amounted to approximately 2.77 times the quiescent current expended between transmissions, while the least energy was 0.17 times the quiescent energy. Note that while nodes on the periphery of the network will make only a single transmission per reporting cycle, those closer to the base station may make many transmissions and receptions as they relay other nodes' data. For simplicity we assume that the probability that any link fails is π for all links in the network.

To permit easy visualisation we begin with a simple network comprising 11 nodes plus a base station and we allow two paths per node, making it a 2-path routing scheme. Figure 5a shows the connectivity map for this network. For simplicity, all node batteries had the same initial charge and all edges were assumed to have the same link failure probability. In subspace pruning, we used $k = 10$ shortest paths for building each $\mathcal{P}_{SP}$ and $\mathcal{P}'_{SP}$; in addition we used braided and edge-disjoint path pruning to build the complete path library $\mathcal{P}$. The initial evolutionary population was created by randomly selecting 100 routes $\mathcal{R}$ from $\mathcal{P}$. Then solving the linear programs (7a) and (14a) generated distinct time shares $\mathcal{T}_L(\mathcal{R})$ and $\mathcal{T}_F(\mathcal{R})$ respectively for each route, resulting in 200 initial solutions $(\mathcal{R}, \mathcal{T})$. The initial archive comprised the non-dominated solutions among these random solutions. After 40000 generations, the Pareto front estimation was judged to have been well-converged based on the dominated hypervolume measure [35].

Figure 4 shows the estimated Pareto front from the evolutionary optimisation. As the figure shows, the 2-path routing schemes in the estimated Pareto set not only contains a wide range of solutions representing various levels of trade-off between network lifetime and robustness, but also performs better than solutions in the random initial archive.[2] The network engineer may then inspect these approximately Pareto optimal solutions and select a suitable multi-path routing scheme for this particular network.

[2] We note that the random initial archive was constructed from solutions within the *pruned* subspace of available routes. Routing schemes selected at random from the entire space generally perform much more poorly than these.

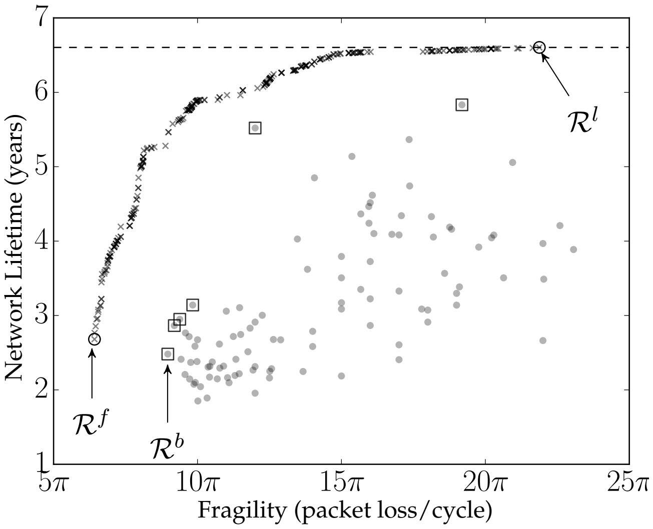


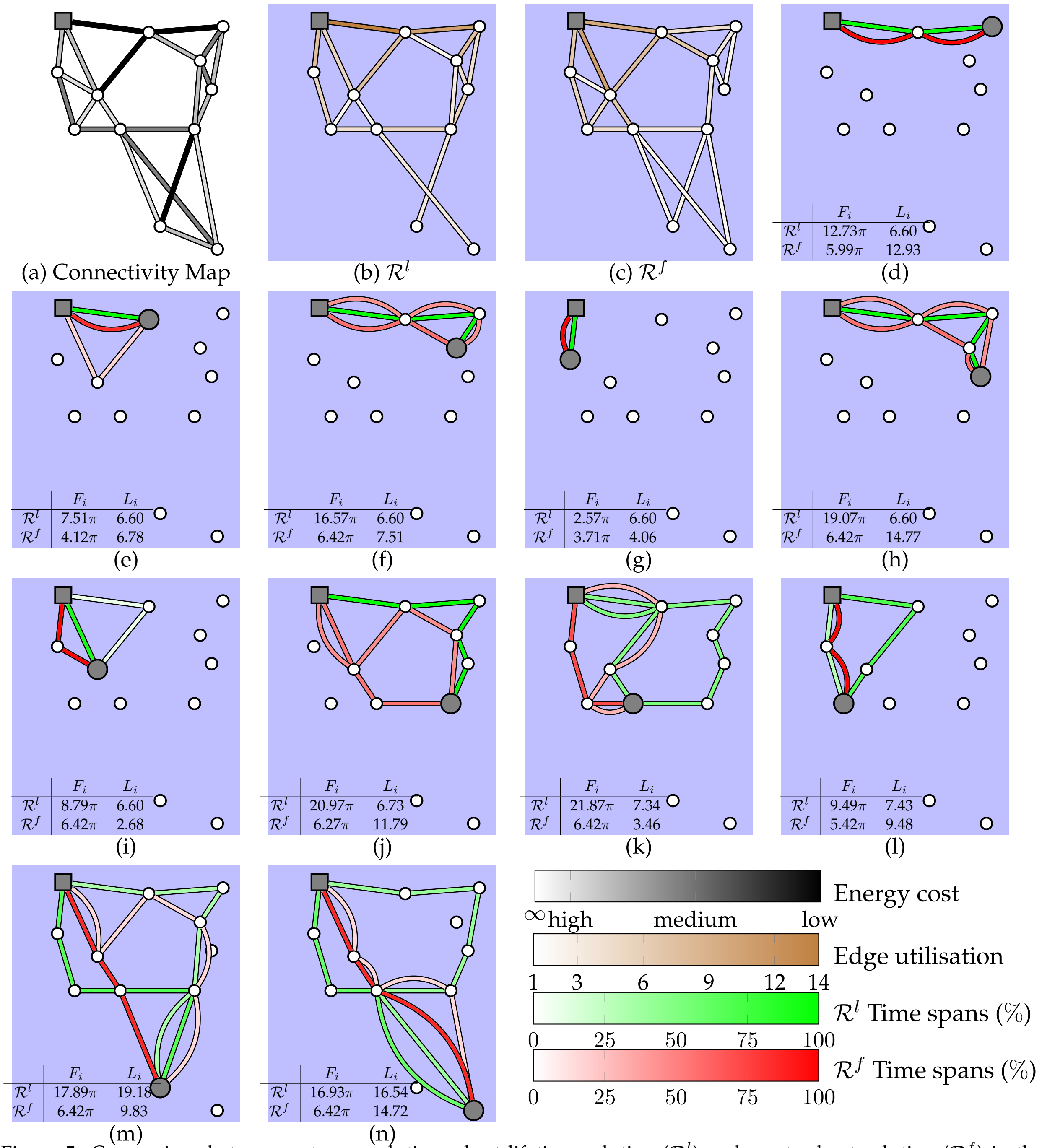
Figure 4: Pareto front approximation showing the trade-off between network lifetime and fragility, resulting from the evolutionary multi-path routing optimisation in a random network with two paths per node. The Pareto front approximation is shown with crosses, with $\mathcal{R}^l$ and $\mathcal{R}^f$ being the best network lifetime and the most robust solutions respectively. Initial random solutions are depicted with solid dots and the non-dominated solutions in the initial archive are indicated with empty square around the associated random solutions. The approximated performance of the braided multi-path routing scheme $\mathcal{R}^b$ is shown with a solid diamond, which is dominated by the final approximated Pareto front. The dashed horizontal line shows the upper bound for network lifetime if unlimited paths per node are permitted. Note that the fragility (maximum expected packet loss per reporting cycle) is proportional to π the failure probability for any link in the network. For instance, if $\pi = 1\%$, then the range of fragility is 0.05 to 0.25 maximum expected packet loss per cycle on the abscissa.

The best lifetime solution $\mathcal{R}^l$ achieves a network lifetime of 6.6 years with two routes allowed per node; this is within 0.001% of the theoretical upper bound of the network lifetime for this particular network, which can be calculated when there is no limit to the number of paths available to each node. At the other extreme, the evolutionary algorithm has located a comparatively robust solution reducing the fragility to 29.3% of the best lifetime solution, at the expense of a 59.4% decrease in network lifetime. In most circumstances the network operator will want to choose a solution between these two extremes. Inspection of the figure shows that a solution close to the "knee" in the Pareto front ($\mathcal{R}^k$; fragility ≈ 9.9, lifetime ≈ 5.9) may be preferred because decreasing the fragility further leads to a rapid deterioration in lifetime, whereas large increases in fragility are required to achieve significantly longer lifetimes.

To clarify the nature of both objectives and their relationship with multi-path routing schemes, we further visualise in Figure 5 the routes corresponding to the solutions $\mathcal{R}^l$ and $\mathcal{R}^f$, which optimise the lifetime and fragility respectively. This figure shows the connectivity map for the network (Figure 5a), the overall edge utilisations for $\mathcal{R}^l$ and $\mathcal{R}^f$ (panels 5b and 5c) and each node's paths in $\mathcal{R}^l$ and $\mathcal{R}^f$. Time shares τ_{i1} and τ_{i2} allocated to each of the two paths from a given node v_i are indicated using the colour scale, green for $\mathcal{R}^l$ and red for $\mathcal{R}^f$.

The edge utilisations for $\mathcal{R}^l$ in Figure 5b indicate that energy efficient edges are utilised to achieve long network lifetime. This is clear when edge utilisations are compared with the connectivity map in Figure 5a: higher edge utilisations occur where energy costs are relatively low. This is also evident from inspecting individual node's paths: for 6 of the 11 nodes $\mathcal{R}^l$ uses single-paths, i.e. only a single path is used to send all the data ($\tau_{i1} = 1$ and $\tau_{i2} = 0$). Multiple paths tend only to be used for nodes distant from the base station (Figures 5k-5n) in order to relieve more heavily loaded nodes closer to the base station.

By contrast, edge utilisations for the least fragile routing $\mathcal{R}^f$ (Figure 5c) show no obvious relationship with energy costs, although nodes close to the base station are inevitably used more heavily than distant nodes. Instead, it is clear that robustness is achieved by providing two paths for 7 of the 11 nodes. Where only a single path is used for a node (e.g., Figure 5g), we have verified that adding an additional path increases the *overall* fragility of the network because it


F_i L_i
$\mathcal{R}^l$ 12.73π 6.60
$\mathcal{R}^f$ 5.99π 12.93
F_i L_i
$\mathcal{R}^l$ 7.51π 6.60
$\mathcal{R}^f$ 4.12π 6.78
F_i L_i
$\mathcal{R}^l$ 16.57π 6.60
$\mathcal{R}^f$ 6.42π 7.51
F_i L_i
$\mathcal{R}^l$ 2.57π 6.60
$\mathcal{R}^f$ 3.71π 4.06
F_i L_i
$\mathcal{R}^l$ 19.07π 6.60
$\mathcal{R}^f$ 6.42π 14.77
F_i L_i
$\mathcal{R}^l$ 8.79π 6.60
$\mathcal{R}^f$ 6.42π 2.68
F_i L_i
$\mathcal{R}^l$ 20.97π 6.73
$\mathcal{R}^f$ 6.27π 11.79
F_i L_i
$\mathcal{R}^l$ 21.87π 7.34
$\mathcal{R}^f$ 6.42π 3.46
F_i L_i
$\mathcal{R}^l$ 9.49π 7.43
$\mathcal{R}^f$ 5.42π 9.48
F_i L_i
$\mathcal{R}^l$ 17.89π 19.18
$\mathcal{R}^f$ 6.42π 9.83
F_i L_i
$\mathcal{R}^l$ 16.93π 16.54
$\mathcal{R}^f$ 6.42π 14.72
Energy cost
∞ high medium low
Edge utilisation
1 3 6 9 12 14
$\mathcal{R}^l$ Time spans (%)
0 25 50 75 100
$\mathcal{R}^f$ Time spans (%)
0 25 50 75 100

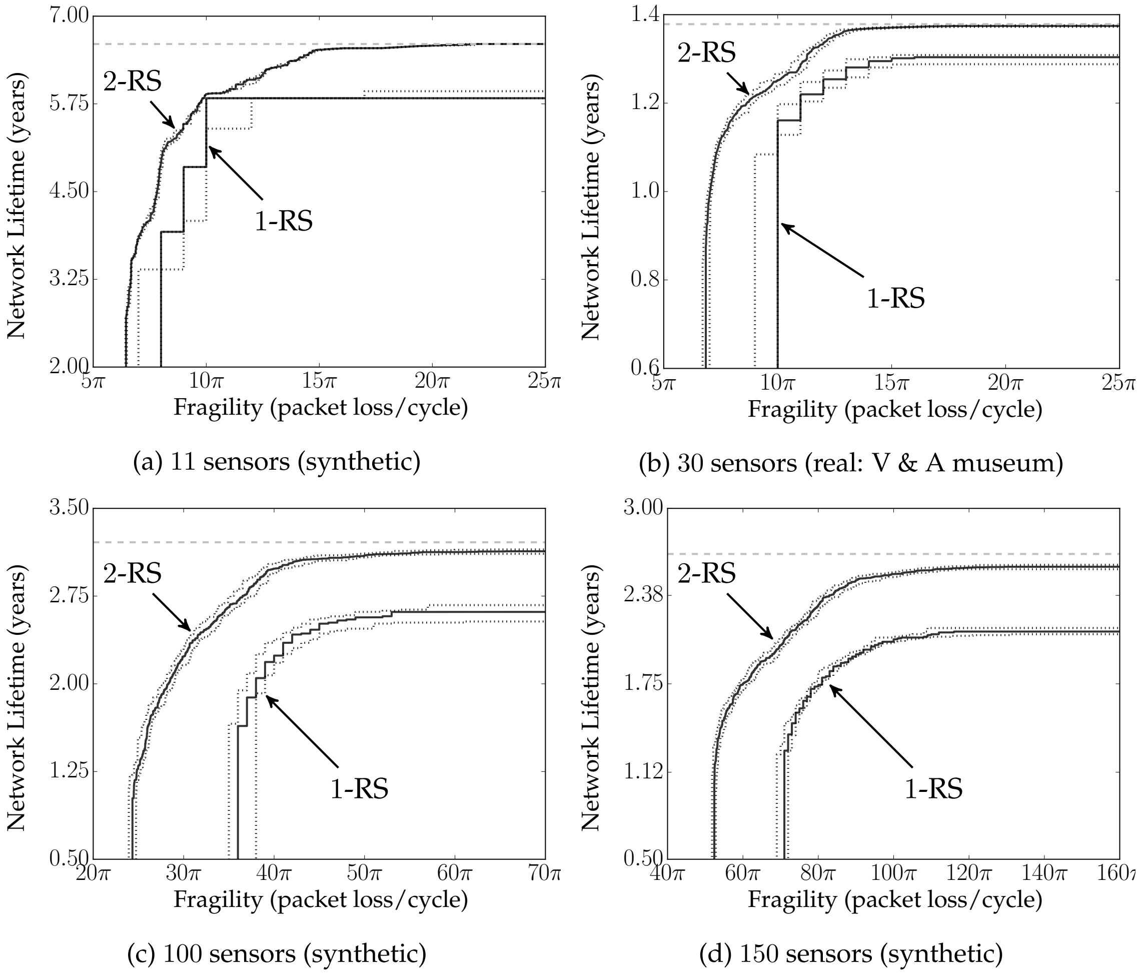
(a) Connectivity Map (b) $\mathcal{R}^l$ (c) $\mathcal{R}^f$ (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n)

Figure 5: Comparison between extreme solutions: best lifetime solution ($\mathcal{R}^l$) and most robust solution ($\mathcal{R}^f$) in the median run of 31 runs for the random network; see Figure 4. (a) shows the connectivity map with all available links between nodes and associated energy costs (darker represents lower energy cost), with the grey square node indicating the base station. (b) and (c) depict the edge utilisations for all routes in solutions $\mathcal{R}^l$ and $\mathcal{R}^f$ respectively (darker represents higher utilisation). Each of (d) – (n) shows the active time shares between a pair of paths forming the solutions $\mathcal{R}^l$ (shades of green) and $\mathcal{R}^f$ (shades of red) for individual nodes (solid grey), with the associated lifetimes and fragility written at the bottom left. Only paths with non-zero time shares are shown. Overall $\mathcal{R}^l$ prefers energy efficient single routes with only few nodes using multi-paths to distribute load from most heavily loaded nodes, while $\mathcal{R}^f$ mainly balances traffic to form disjoint paths and improve fragility irrespective of energy costs. Note that the fragility F_i (maximum expected packet loss per reporting cycle) is scaled by π, where $\pi_m = \pi$ is the failure probability at any edge e_m.



(a) 11 sensors (synthetic)

(b) 30 sensors (real: V & A museum)

(c) 100 sensors (synthetic)

(d) 150 sensors (synthetic)

Figure 6: Performance comparison between single-path routing schemes (1-RS) and 2-path routing schemes (2-RS) approaches for random synthetic networks with 11 (a), 100 (c) and 150 sensors (d), and a real network with 30 sensors (b) deployed in Victoria & Albert Museum. The solid lines depict 50% summary attainment surfaces and dotted lines show 10% and 90% surfaces. The theoretical upper bound of the maximum minimum lifetime, calculated by solving the linear program proposed by Chang *et al.* [1], for the network is shown with horizontal dashed line in grey. In all cases the multi-path schemes (2-RS) completely dominates the single-path schemes (1-RS), and the best network lifetimes achieved using multi-path schemes (2-RS) are within 3% of the theoretical maximum. Note that the fragility (maximum expected packet loss per reporting cycle) is scaled by π, where $\pi_m = \pi$ is the failure probability at any edge e_m.

requires additional traffic to use an already vulnerable link. It is, of course, possible to ensure that all nodes incorporate some redundancy by using distinct paths at least a fraction τ_{min} of the time by replacing the positivity constraints (7c) and (14c) with $\tau_{kd} \geq \tau_{min}$ for all k, d.

In Figure 6a, we present the summary attainment surfaces [41, 42]. These are the Pareto fronts achieved by 10%, 50% and 90% of 31 independent runs of the evolutionary optimiser. Clearly, the attainment surfaces for 2-path schemes are superior to those from single-path schemes in terms of both lifetime and fragility. We attribute this to the possibility of load balancing between the paths for a single nodes as well as between nodes. Also, the narrow width of the attainment surfaces indicates desirable repeatability and convergence properties of the proposed evolutionary approach.

We obtained similar results with a random network comprising 100 (Figure 6c) and 150 (Figure 6d) sensor nodes. To achieve satisfactory performance, the number of function evaluations required for 100 node network was 100000, and for 150 node network was 200000. As Table 1 shows, in both cases, the best network lifetime was within 3% of the optimal solution for networks with unlimited paths per node [1].

Table 1: Comparison between the theoretical optimal lifetime $\mathcal{L}^*$ and the lifetime of the best lifetime solution ($\mathcal{R}^l$) from the median run of evolutionary 2-path routing for synthetic and real networks. $\mathcal{L}^*$ was found by solving the linear program proposed by Chang *et al.* [1]; this may not be realised in multi-path system with a limited number of paths available to each node.

Number of sensor nodes	Network lifetime (years)		$L(\mathcal{R}^l)/\mathcal{L}^*$
	$L(\mathcal{R}^l)$	$\mathcal{L}^*$	
11 (synthetic)	6.60	6.61	99.8%
30 (real)	1.37	1.38	99.2%
100 (synthetic)	3.12	3.21	97.2%
150 (synthetic)	2.59	2.67	97.0%

5.2. *Real Network Evaluation*

Finally, we show the performance of the approach on a real network deployed in the Victoria & Albert Museum, London.[3] To ensure the preservation of their artefacts, many museums and art galleries employ battery-powered wireless sensor networks in order to monitor temperature and humidity. Battery power combined with wireless means that such sensors can be easily placed practically anywhere, such as in environmentally controlled cabinets, without the need for additional wiring. The network comprises 30 sensor nodes and a base station, distributed across five floors, and within an area of approximately $35,000$ m^2. The thick, solid walls of the museum mean that some sensors which are close spatially nevertheless cannot directly communicate (or can only do so reliably at a high transmission power). Nodes are connected to between 3 and 21 other nodes, with the average degree being 11.9. The characteristics of the radio environment also vary with the passage of visitors through the galleries (over 3.4 million visitors passed through the museum in 2015), which may lead to occasional link failures.

We used the same evolutionary algorithm configuration as described for the synthetic network for evolutionary multi-path routing optimisation in this network; except 60000 function evaluations were required to achieve similar convergence quality as indicated by the dominated hypervolume measure [35]. Routing optimisation, using two paths per node, resulted in a range of trade-off solutions, with the best lifetime solution in the median run of 31 optimisations achieving 99.51% of the theoretical best network lifetime with unlimited paths (Table 1). In this network the quiescent energy expenditure is higher, so that transmission and reception energy costs amount to between 0.017 and 0.27 times the quiescent energy expended in a reporting cycle. Consequently, the maximum available lifetime and the range of available lifetimes is smaller than in the synthetic network where transmission and reception costs are more significant. Nonetheless, the evolutionary search has located a wide range of lifetimes and shows the trade-off with robustness. At the other extreme, the fragility of most robust solution was 37.9% of the robustness of the best lifetime routing at an expense of a reduction of 29.5% in network lifetime in comparison with the best lifetime solution.

The Figure 6b shows the $10\%, 50\%$ and 90% summary attainment surfaces from 31 optimisation runs for one and two paths per node routings. As for the synthetic network the close proximity of the attainment surfaces, indicates the repeatability and convergence of the optimisation. The optimised 2-path routings clearly dominate the single path routings, providing better lifetimes and robustness. In this network, allowing an additional path for each node does not provide significantly better routings.

5.3. *Comparison with Braided Multi-Path Routing Scheme*

In their seminal and popular work, Ganesan *et al.* [7] described the braided multi-path routing scheme, which is known to be fault tolerant and energy efficient. Although this scheme is implemented in a distributed manner, we can approximate the performance of a centralised implementation and thus compare it with our evolutionary multi-path approach.

[3] To allow comparison with this work, data on the network can be found at http://emps.exeter.ac.uk/computer-science/wsn/

In a braided multi-path scheme all partially disjoint paths from the primary path are constructed for each source node (see section 4.1.2 for a brief description of this process). When the primary path fails to deliver messages, data is sent via alternative paths with a preference towards shorter paths, as indicated by the maintenance overhead metric described in [7]. In essence a list of paths for each node, ordered according to increasing length or total failure probability is maintained; each path is only used when all paths with higher preferences have failed. In this context the proportional usage (time share) of each route is thus dependent on its path success probability and the path failure probabilities of the paths with higher preferences. We model the time share of a path R_{id} as:

$$\tau_{id} = (1 - p_{id}) \prod_{l=1}^{d-1} p_{il} \qquad 1 \leq d \leq D \tag{20}$$

where p_{id} is the probability path R_{id} failing.

With this model of the time shares it is clear that $\sum_{d=1}^{D} \tau_{id} \leq 1$, where there are D braided paths from a node v_i to the base station. Consequently, there may be residual time for which no route has been allocated a time share. For a fair comparison with our approach, we proportionally distribute this residual time between paths and thus we model the time shares as:

$$\tau'_{id} = \frac{\tau_{id}}{\sum_l \tau_{il}}. \tag{21}$$

Table 2: The braided multi-path ($\mathcal{R}^b$) is compared with the most robust solution ($\mathcal{R}^f$), the best lifetime solution ($\mathcal{R}^l$) and a selected solution from the "knee" of the estimated front ($\mathcal{R}^k$) from the median run of evolutionary 2-path routing for synthetic and real networks. The minimum fragility and maximum network lifetime are shown in bold.

Number of sensor nodes	Fragility				Network lifetime (years)			
	$\mathcal{R}^b$	$\mathcal{R}^f$	$\mathcal{R}^l$	$\mathcal{R}^k$	$\mathcal{R}^b$	$\mathcal{R}^f$	$\mathcal{R}^l$	$\mathcal{R}^k$
11 (synthetic)	8.79	**6.42**	21.90	8.20	2.48	2.68	**6.60**	5.28
30 (real)	14.24	**7.07**	18.65	11.81	0.61	0.96	**1.37**	1.34
100 (synthetic)	77.19	**24.48**	62.02	39.97	0.31	1.17	**3.12**	2.98
150 (synthetic)	144.75	**51.36**	124.4	88.76	0.51	1.01	**2.59**	2.50

We present the comparison between the approximate performance of the braided multi-path routing and the evolutionary 2-path routing in Table 2. As the table shows, the 2-path routings obtained by evolutionary optimisation all dominate the braided path routings. The most robust solution $\mathcal{R}^f$ improves both objectives, with fragility improved by at least 25% in comparison to $\mathcal{R}^b$. Choosing a routing, $\mathcal{R}^k$, close to the knee of the estimated trade-off front can increase the network lifetime significantly while also improving fragility in comparison to $\mathcal{R}^b$. In general, the evolutionary 2-path solutions always lead to longer network lifetimes and than the braided path routings. The better performance of the 2-path evolved routings is due to the fact that the braided multi-path scheme tends to prefer shorter paths and the braided paths for each source are chosen without regard for other nodes; they are therefore less effective at utilising the load balancing capability of multi-path routing schemes which may be detrimental to a small group of nodes that relay the most traffic. In contrast, most solutions from the estimated trade-off front are more robust and the advantage becomes more greater in larger networks. We attribute this improvement to the flexibility that larger networks provide for achieving greater distribution of expected data loss between paths.

Note that we did not impose any constraints on the number of paths for the braided multi-path scheme, which was free to utilise all possible braided paths. The evolved multi-paths were able to achieve better performance even though they were constrained to have only 2 paths.

6. Related Work

Multi-path routing has become popular in the past few years (for recent surveys on multi-path routing in wireless sensor networks see [23, 5, 4]). This is primarily because multiple paths for sending data are particularly useful for balancing load to achieve better network lifetime (as a proxy for energy efficiency) and improving reliability under uncertainty at links. However, finding multiple paths and deciding on the time shares between them are challenging tasks. As we have pointed out here there is a trade-off between network lifetime and robustness against link failures. In this paper, we have focused on devising generals methods for locating routes and the associated traffic, and we proposed an evolutionary approach to estimate the optimal trade-off between these objectives. In this section we discuss this algorithm in the context of recent related work.

Most current routing approaches in wireless sensor networks deal with network lifetime as a single objective optimisation problem; this is known as maximum lifetime routing. Saleh *et al.* present many such methods in their survey paper [43]. The most notable work in this area is due to Chang *et al.* [1], who model the routing problem in terms of a linear program which may be solved efficiently. One criticism of the original work is that it does not consider the link level handshaking: the process of confirming the reception of data by the destination at the link level [44]. In our formulation in section 2.2, we included the link level handshaking. More importantly, the Chang *et al.* solution is applicable when there is no limit to the number of paths that a node may utilise. Unfortunately, very low power sensor networks, particularly those with limited memory and processing resources at each node, must limit the number of paths available to each node. Consequently, alternative methods such as those presented here must be found for networks with these characteristics. As we have shown, the evolutionary approach is able to find solutions with few paths per node with lifetimes that closely approach the Chang *et al.* upper bound. Nevertheless, the multi-objective nature of the algorithm shows that it may be advantageous to sacrifice some longevity for robustness to link failure.

To select paths that are resilient to unpredictable faults, often node-disjoint or partially disjoint paths from a node to the base station are sought. The earliest example of such a multi-path routing protocol in WSNs is due to Intanagonwiwat *et al.* [45]. They devised the directed diffusion algorithm that constructs node-disjoint paths in a distributed fashion. All discovered paths are maintained and the best among these are used to send data back to the base station. In case of a failure, the next best path is used. However, depending on the network topology there may not be many completely disjoint paths between a node and the base station. Moreover, using completely disjoint paths results in longer paths, which generally consumes more power, lowering the network lifetime. Also, path discovery and maintenance is dependent on broadcast messages flooded through the network. To reduce the overhead due to flooding, Challal *et al.* introduced a sub-branch multipath routing protocol that uses only one message per node to discover paths, but insist on using node-disjoint paths [46]. Similarly, Yang *et al.* proposed a sleeping multi-path routing protocol to conserve energy in directed diffusion [26]. It uses expected path success and a reliability requirement on paths to limit the number of active alternative paths and hence reduce the overhead of maintaining many disjoint paths. To improve the energy efficiency by reducing the length of the alternative paths, Ganesan *et al.* proposed the braided multi-path protocol [7]. In this protocol, rather than building completely disjoint paths, partially disjoint paths are constructed. This is based on the intuition that a path may fail due to an edge failure, while the other edges may still be usable. Exploiting this idea, the method allows efficient use of shorter paths as it does not require the next best path to be completely disjoint. This has been shown to be 50% more reliable and energy efficient than using idealised node-disjoint multipath protocol in [45]. In these protocols, as new routes are used on failure, the uncertainty at links is managed reactively. Also, network lifetime is not directly addressed.

The optimisation scheme here produces superior results to these methods because it is able to directly optimise the network robustness expressed via the fragility which, as discussed in section 2.3, measures the maximum expected packet loss for the entire network in case of a link failure. This is in contrast to most current methods [18, 23, 24, 25, 26, 27] for assessing path robustness, which consider the paths for each source node without regard for the links which are used by paths for other nodes. As our simulation results show, the optimisation produces routing schemes with much improved robustness compared with the popular and effective braided paths heuristic [7]. Our work builds on the braided paths idea by incorporating many braided paths into the library of paths from which evolutionary algorithm builds complete routing schemes.

Many recent routing protocols have attempted to combine different objectives, such as network lifetime, network robustness and data latency, into a single objective. For instance, Huang *et al.* formulate an integer linear program (ILP) to minimise the number of active routes with constraints on acceptable delay and reliability [47]. The active routes are used equally to send data. Bagula proposes an extension to the ILP with an additional constraint on energy consumption [25]. The data from a source node is then sent via a randomly selected path from the available multi-paths. Ben-Othman *et al.* defines a composite edge cost as the weighted sum of residual energy, available buffer size and the signal-to-noise-ratio [9, 8]. The path costs are calculated by summing the relevant edge costs. Also, a reliability constraint and expected path success are used to calculate the optimal number of alternative paths as described in [48]. Data from a node is split into smaller segments and sent through different paths using a queuing model. The data is reconstructed on reception by the base station. Similarly, in [49], Radi *et al.* formulate another cost metric which is a product of accumulated expected transmission count, inverse residual battery and experienced interference level. Node-disjoint paths are then built by minimising this metric. The traffic between routes are optimised by solving a linear program based on a path load cost metric and traffic ratio.

Like some other authors, rather than combining the possibly competing objectives into a single objective, we prefer to expose the individual objectives by optimising them all simultaneously to approximate the Pareto front. With this on hand the network manager can make an informed decision about the trade-offs involved in selecting a routing scheme. More technically, it is known that combining objectives as a weighted sum results in a single objective problem which may not capture the full extent of the trade-offs between the constituent objectives even as the weights are varied [50]. Most current EA-based multi-objective routing optimisation approaches consider single-path routing schemes to optimise various objectives: energy efficiency, network lifetime, latency, robustness, expected transmission count, etc. [11, 12, 13, 14]. To the best of our knowledge, this work represents the first attempt to estimate and explore the optimal trade-off between network lifetime and robustness using multi-path routings in wireless sensor networks.

7. Conclusions

Despite advances in battery technology it remains important to extend the network lifetime of wireless sensor networks by choosing routes that balance the loads placed on individual nodes. However, as we have shown here, optimisation solely of the lifetime may be detrimental to the robustness of the network to link failure. In this paper we have therefore presented novel methods for discovering the best trade off between network lifetime and network robustness in multi-path routing schemes.

Chang *et al.* [1] elegantly showed how the network lifetime may be maximised by solving a linear program. However, their work is applicable only when each node may use an unlimited number of paths, which is often impractical for low-power, limited memory devices. In contrast, limiting the number of paths results in an NP-hard combinatorial problem. Nonetheless, in the example networks presented here our multi-objective evolutionary algorithm is able to locate routes with network lifetimes within 3% of the Chang *et al.* lifetime, even with only two paths per node. Undoubtedly there will be networks that require more paths to approach the Chang *et al.* lifetime, but our experience suggests that two paths allow routes with very good lifetimes to be found.

Network robustness to link failure is quantified by the network *fragility*, which measures the maximum expected data loss in the complete network if a link fails. This measure incorporates the probabilities of individual links failing. Initially, in the absence of any prior information these probabilities may be set equal, but during network operation links may be monitored to better estimate these probabilities. With these on hand, the optimisation may be repeated to minimise the fragility. In the case of protracted link failure, we have shown elsewhere [6] that re-optimisation of the routing using the present solution as a base can cope with dynamic conditions.

Evolutionary algorithms are an effective way to obtain good solutions to combinatorial optimisation problems. Efficient optimisation for this problem was obtained by two measures. First, the space of possible solutions was pruned to limit the search to routes likely to have good lifetime and/or robustness. Secondly, we showed how to find the optimal division of traffic between the paths from each node by solving a linear program for either lifetime or fragility. The resulting hybrid algorithm is efficient because the need to use a further, relatively slow, evolutionary process to find the optimal division of traffic is obviated.

We note that it is straightforward to extend the two-objective algorithm presented here to three or more objectives. For instance, if there is a group of nodes whose lifetime should be given priority because they are in particularly inaccessible locations, then the minimum lifetime within this group may be treated as an additional third objective [40]. Also, objectives describing other desirable network properties, such as latency, could also be optimised.

This work has used linear models for the batteries, which is a good approximation. However, close to the end of life and under intermittent loads batteries may be display nonlinear behaviour. Future work will include modelling the uncertainties introduced by battery nonlinearities together with uncertainties introduced by lack of information about link failure probabilities in order to allow for more robust optimisation of WSNs under these uncertainties.

Acknowledgments

Part of this work was supported by a Knowledge Transfer Partnership awarded to the University of Exeter and the IMC Group Ltd (KTP008748). We would like to thank Martin Hancock (Technical Director, The IMC Group Ltd.), and Neil Lundy (Engineering Manager, The IMC Group Ltd.) for their support and contributions. We would also like to thank Boris Pretzel (Chief Scientist, Victoria & Albert Museum), and Bhavesh Shah (Scientist, Victoria & Albert Museum), for facilitating the real network test.

Appendix A. Exploiting Intermediate Time Shares

The evolutionary approach discussed in this paper is capable of generating a wide range of solutions representing varying degrees of trade-offs between network lifetime and robustness. During the evolutionary process, a set of routes $\mathcal{R}$ is proposed through the evolutionary mechanism, and then the optimal time shares $\mathcal{T}_L(\mathcal{R})$ and $\mathcal{T}_F(\mathcal{R})$ maximising the network lifetime and robustness respectively are located by solving the LPs (7) and (14) (step 6 in Algorithm 1). However, for any given $\mathcal{R}$, there is a continuum of time shares $\mathcal{T}$ which optimally trade-off between network lifetime and robustness for that particular $\mathcal{R}$. These intermediate time shares between the extremal solutions are used during the evolutionary search. In this appendix, we describe how these intermediate time shares may be used to improve approximation to the Pareto front.

The optimal trade-off front is a result of solving a *bi-objective linear program* (BOLP) [51]:

$$\min_{\mathcal{T}} L^{'*}(\mathcal{T}, \mathcal{R}) \tag{A.1}$$

$$\min_{\mathcal{T}} F^{*}(\mathcal{T}, \mathcal{R}) \tag{A.2}$$

subject to the constraints described in (7b), (14b), (14c), and (14d). Since we wish to find the Pareto optimal time shares for a given $\mathcal{R}$, for notational simplicity from now on we suppress the dependence of these objectives on $\mathcal{R}$.

The BOLP may be expressed as a weighted sum LP to minimise over all time shares $\mathcal{T}$:

$$h_\lambda(\mathcal{T}) = \lambda L^{'*}(\mathcal{T}) + (1-\lambda)F^{*}(\mathcal{T}), \tag{A.3}$$

with the same constraints, where $\lambda \in [0,1]$ controls the weight associated with each objective, and thus represents its relative importance. The Pareto optimal time shares can be found by solving the weighted sum LP for sufficiently many λ.

In fact, Ehrgott showed that the front is a convex polygon, and thus may be described entirely in terms of the vertices and relevant λs (Figure A.7) [51]. Exploiting this convexity property, he describes a bi-objective simplex method to exactly locate the λs for the vertices on the front. However, the number of vertices on the front is not known *a priori*. Also there may be many vertices depending on the complexity of the problem. Hence incorporating the intermediate time shares from all the vertices, during or post evolution, may be computationally expensive. We therefore elect to locate a few solutions evenly spaced between the minimum inverse network lifetime $L'^*(\mathcal{T})$ and the minimum fragility $F^*(\mathcal{T})$ following the evolutionary search.

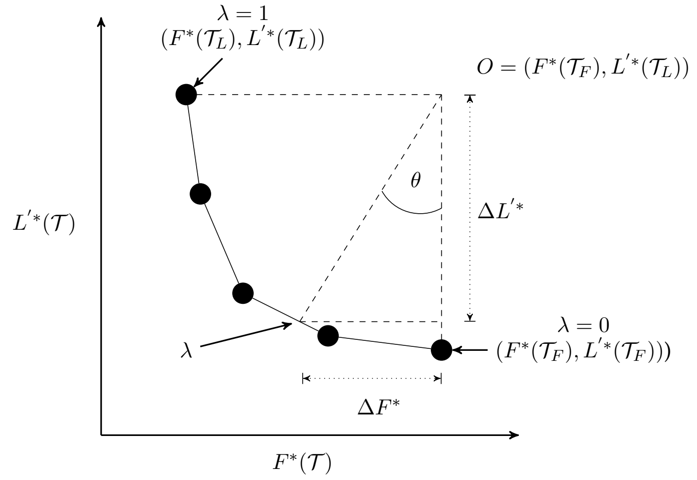

$\lambda = 1$
$(F^*(\mathcal{T}_L), L'^*(\mathcal{T}_L))$
$O = (F^*(\mathcal{T}_F), L'^*(\mathcal{T}_L))$
θ
$\Delta L'^*$
$L'^*(\mathcal{T})$
$\lambda = 0$
$(F^*(\mathcal{T}_F), L'^*(\mathcal{T}_F)))$
λ
ΔF^*
$F^*(\mathcal{T})$

Figure A.7: An illustration of the optimal Pareto front for a given routing scheme $\mathcal{R}$, obtained by solving the BOLP in (A.3). The front is a convex polygon that is completely defined by the vertices (solid dots). The extremal vertex and the associated time share $\mathcal{T}_L$ for inverse network lifetime can be achieved by setting $\lambda = 1$ and solving the BOLP (the BOLP reduces to the LP in (7)). Similarly, $\mathcal{T}_F$ for optimal fragility can be obtained with $\lambda = 0$. Given a target angle θ, we use a bisection algorithm to locate a solution on the BOLP front together with a λ corresponding to θ.

Algorithm 2 Bisection search for target angle, θ.

```
Inputs
    $\mathcal{R}$ : A $D$-path routing scheme
    $\theta$ : Target angle corresponding to a particular weight ($\lambda$)
    $\delta\theta$ : Tolerance for angle
    $O$ : Reference point $(F^*_o, L'^*_o) = (F^*(\mathcal{T}_F), L'^*(\mathcal{T}_L))$

Steps
$\lambda_a \leftarrow 0$
$\lambda_b \leftarrow 1$
repeat
    $\lambda' \leftarrow \frac{\lambda_a+\lambda_b}{2}$
    $\mathcal{T}' = \text{argmin}_{\mathcal{T}}\, h_{\lambda'}(\mathcal{T})$        ▷ Solve weighted sum BOLP for $\mathcal{T}$ with $\lambda'$ (A.3)
    $\hat{F^*}, \hat{L'^*} \leftarrow \text{Normalise}(F^*(\mathcal{T}'), L'^*(\mathcal{T}'))$
    $\theta' = \arctan\left(\frac{F^*_o - \hat{F^*}}{L'^*_o - \hat{L'^*}}\right)$
    if  $\theta' < \theta - \delta\theta$ :  $\lambda_0 \leftarrow \lambda'$
    else  $\lambda_1 \leftarrow \lambda'$
until $|\theta - \theta'| \leq \delta\theta$
return $\mathcal{T}'$        ▷ Time shares pertinent to the target angle
```

Solving the BOLP (A.3) for evenly spaced $\lambda \in [0, 1]$ does not usually result in an even coverage of the BOLP Pareto front [50]. To locate evenly spaced solution we normalise the objectives so that the sides of a triangle: $(F^*(\mathcal{T}_L), L^{'*}(\mathcal{T}_L))$–$(F^*(\mathcal{T}_F), L^{'*}(\mathcal{T}_L))$ and $(F^*(\mathcal{T}_F), L^{'*}(\mathcal{T}_L))$–$(F^*(\mathcal{T}_F), L^{'*}(\mathcal{T}_F))$ have equal length (see Figure A.7). Then we choose evenly spaced target angles $\theta \in [0, \frac{\pi}{2}]$ and use a bisection algorithm to simultaneously find the λ corresponding to the target angle and the solution of the BOLP. The bisection algorithm is described in Algorithm 2.

In order to augment the estimated Pareto set, A, we apply this method for each $\mathcal{R} \in A$ in turn, and add each BOLP solution to A, retaining only the non-dominated solutions. Thus A represents the final approximation of the Pareto set for all routing schemes and associated time shares.

References

[1] J.-H. Chang, L. Tassiulas, Maximum Lifetime Routing in Wireless Sensor Networks, IEEE/ACM Transactions on Networking 12 (4) (2004) 609–619.

[2] Z. Cheng, M. Perillo, W. Heinzelman, General Network Lifetime and Cost Models for Evaluating Sensor Network Deployment Strategies, IEEE Transactions on Mobile Computing 7 (4) (2008) 484–497.

[3] H. U. Yildiz, K. Bicakci, B. Tavli, H. Gultekin, D. Incebacak, Maximizing Wireless Sensor Network Lifetime by Communication/Computation Energy Optimization of Non-repudiation Security Service: Node Level Versus Network Level Strategies , Ad Hoc Networks 37, Part 2 (2016) 301 – 323.

[4] M. Radi, B. Dezfouli, K. A. Bakar, M. Lee, Multipath Routing in Wireless Sensor Networks: Survey and Research Challenges, Sensors 12 (1) (2012) 650–685.

[5] K. Sha, J. Gehlot, R. Greve, Multipath Routing Techniques in Wireless Sensor Networks: A Survey, Wireless Personal Communications 70 (2) (2013) 807–829.

[6] A. A. M. Rahat, R. M. Everson, J. E. Fieldsend, Multi-objective Routing Optimisation for Battery-powered Wireless Sensor Mesh Networks, in: Proceedings of the 2014 Conference on Genetic and Evolutionary Computation, GECCO '14, ACM, New York, NY, USA, 2014, pp. 1175–1182.

[7] D. Ganesan, R. Govindan, S. Shenker, D. Estrin, Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks, ACM SIGMOBILE Mobile Computing and Communications Review 5 (4) (2001) 11–25.

[8] B. Yahya, J. Ben-Othman, REER: Robust and Energy Efficient Multipath Routing Protocol for Wireless Sensor Networks, in: Proceedings of the 28th IEEE Conference on Global Telecommunications, GLOBECOM '09, IEEE Press, Piscataway, NJ, USA, 2009, pp. 5482–5488.

[9] J. Ben-Othman, B. Yahya, Energy Efficient and QoS Based Routing Protocol for Wireless Sensor Networks, Journal of Parallel and Distributed Computing 70 (8) (2010) 849–857.

[10] C. A. Coello Coello, G. B. Lamont, D. A. V. Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, Springer, 2007.

[11] F. Xue, A. Sanderson, R. Graves, Multi-Objective Routing in Wireless Sensor Networks with a Differential Evolution Algorithm, in: Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control, ICNSC '06, 2006, pp. 880–885.

[12] K. Jaffres-Runser, C. Comaniciu, J.-M. Gorce, A Multiobjective Optimization Framework for Routing in Wireless Ad Hoc Networks, Tech. Rep. RR-7180, INRIA, <inria-00449010> (2009).

[13] N. Magaia, N. Horta, R. Neves, P. R. Pereira, M. Correia, A Multi-Objective Routing Algorithm for Wireless Multimedia Sensor Networks, Applied Soft Computing 30 (2015) 104–112.

[14] H. Yetgin, K. T. K. Cheung, L. Hanzo, Multi-Objective Routing Optimization Using Evolutionary Algorithms, in: Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC '12, 2012, pp. 3030–3034.

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, 2001.

[16] I. Zang, A Smoothing-Out Technique for Min-Max Optimization, Mathematical Programming 19 (1) (1980) 61–77.

[17] D. Bertsimas, J. N. Tsitsiklis, Introduction to Linear Optimization, Vol. 6, Athena Scientific Belmont, MA, USA, 1997.

[18] L. Paradis, Q. Han, A Survey of Fault Management in Wireless Sensor Networks, Journal of Network and Systems Management 15 (2) (2007) 171–190.

[19] V. Cionca, T. Newe, V. Dadarlat, TDMA Protocol Requirements for Wireless Sensor Networks, in: Proceedings of the Second International Conference on Sensor Technologies and Applications, SENSORCOMM '08, 2008, pp. 30–35.

[20] J. Jeong, C. T. Ee, Forward Error Correction in Sensor Networks, in: Proceedings of the 1st International Workshop on Wireless Sensor Networks, WWSN '07, 2007.

[21] A. Cerpa, J. Wong, L. Kuang, M. Potkonjak, D. Estrin, Statistical Model of Lossy Links in Wireless Sensor Networks, in: Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks, IPSN '05, 2005, pp. 81–88.

[22] J. Zhao, R. Govindan, Understanding Packet Delivery Performance in Dense Wireless Sensor Networks, in: Proceedings of the First International Conference on Embedded Networked Sensor Systems, SenSys '03, ACM, New York, NY, USA, 2003, pp. 1–13.

[23] J. Tsai, T. Moors, A Review of Multipath Routing Protocols: From Wireless Ad Hoc to Mesh Networks, in: Proceedigns of the ACoRN Early Career Researcher Workshop on Wireless Multihop Networking, Vol. 30, 2006.

[24] T. Korkmaz, K. Sarac, Characterizing Link and Path Reliability in Large-Scale Wireless Sensor Networks, in: Proceedings of the IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob '10, IEEE, 2010, pp. 217–224.

[25] A. B. Bagula, Modelling and Implementation of QoS in Wireless Sensor Networks: A Multiconstrained Traffic Engineering Model, EURASIP Journal on Wireless Communications and Networking 2010 (2010) 1.

[26] O. Yang, W. Heinzelman, Sleeping Multipath Routing: A Trade-off Between Reliability and Lifetime in Wireless Sensor Networks, in: Proceedings of the 30th IEEE Conference on Global Telecommunications, GLOBECOM '11, 2011, pp. 1–5.

[27] A. Dâmaso, N. Rosa, P. Maciel, Reliability of Wireless Sensor Networks, Sensors 14 (9) (2014) 15760–15785.

[28] W. Lou, An Efficient N-to-1 Multipath Routing Protocol in Wireless Sensor Networks, in: Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, IEEE, 2005, pp. 8–pp.

[29] E. Felemban, C.-G. Lee, E. Ekici, MMSPEED: Multipath Multi-SPEED Protocol for QoS Guarantee of Reliability and Timeliness in Wireless Sensor Networks, IEEE Transactions on Mobile Computing 5 (6) (2006) 738–754.

[30] Y. Ming Lu, V. W. S. Wong, An Energy-efficient Multipath Routing Protocol for Wireless Sensor Networks: Research Articles, International Journal of Communication Systems 20 (7) (2007) 747–766.

[31] L. G. Valiant, The Complexity of Computing the Permanent, Theoretical computer science 8 (2) (1979) 189–201.

[32] L. G. Valiant, The Complexity of Enumeration and Reliability Problems, SIAM Journal on Computing 8 (3) (1979) 410–421.

[33] B. Roberts, D. P. Kroese, Estimating the Number of s-t Paths in a Graph, Journal of Graph Algorithsm and Applications 11 (1) (2007) 195–214.

[34] J. E. Fieldsend, R. M. Everson, S. Singh, Using Unconstrained Elite Archives for Multiobjective Optimization, IEEE Transactions on Evolutionary Computation 7 (3) (2003) 305–323.

[35] E. Zitzler, Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications, Ph.D. thesis, ETH Zurich, Switzerland (1999).

[36] J. Y. Yen, Finding the K Shortest Loopless Paths in a Network, Management Science 17 (11) (1971) 712–716.

[37] D. Eppstein, Finding the K Shortest Paths, SIAM Journal of Computing 28 (2) (1999) 652–673.

[38] A. W. Brander, M. C. Sinclair, A Comparative Study of k-Shortest Path Algorithms, in: Proceedings of the Eleventh UK Performance Engineering Workshop, 1995, pp. 370–379.

[39] E. W. Dijkstra, A Note on Two Problems in Connexion with Graphs, Numerische Mathematik 1 (1) (1959) 269–271.

[40] A. A. M. Rahat, R. M. Everson, J. E. Fieldsend, Hybrid Evolutionary Approaches to Maximum Lifetime Routing and Energy Efficiency in Sensor Mesh Networks, Evolutionary Computation Journal 23 (3) (2015) 451–507.

[41] C. Fonseca, P. Fleming, On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers, in: H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature - PPSN IV, Vol. 1141 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1996, pp. 584–593.

[42] J. Knowles, A Summary-Attainment-Surface Plotting Method for Visualizing the Performance of Stochastic Multiobjective Optimizers, in: Proceedings of the International Conference on Intelligent Systems Design and Applications, ISDA '05, 2005, pp. 552–557.

[43] S. Saleh, M. Ahmed, B. M. Ali, M. F. A. Rasid, A. Ismail, A Survey on Energy Awareness Mechanisms in Routing Protocols for Wireless Sensor Networks using Optimization Methods, Transactions on Emerging Telecommunications Technologies 25 (12) (2014) 1184–1207.

[44] H. U. Yildiz, B. Tavli, H. Yanikomeroglu, Transmission Power Control for Link-Level Handshaking in Wireless Sensor Networks, IEEE Sensors Journal 16 (2) (2016) 561–576.

[45] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks, in: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, MobiCom '00, ACM, New York, NY, USA, 2000, pp. 56–67.

[46] Y. Challal, A. Ouadjaout, N. Lasla, M. Bagaa, A. Hadjidj, Secure and Efficient Disjoint Multipath Construction for Fault Tolerant Routing in Wireless Sensor Networks, Journal of Network and Computer Applications 34 (4) (2011) 1380 – 1397, advanced Topics in Cloud Computings.

[47] X. Huang, Y. Fang, Multiconstrained QoS Multipath Routing in Wireless Sensor Networks, Wireless Networks 14 (4) (2008) 465–478.

[48] S. Dulman, T. Nieberg, J. Wu, P. Havinga, Trade-Off between Traffic Overhead and Reliability in Multipath Routing for Wireless Sensor Networks, in: Proceedings of the IEEE Conference on Wireless Communications and Networking, Vol. 3 of WCNC '03, 2003, pp. 1918–1922.

[49] M. Radi, B. Dezfouli, S. A. Razak, K. A. Bakar, LIEMRO: A Low-Interference Energy-Efficient Multipath Routing Protocol for Improving QoS in Event-Based Wireless Sensor Networks, in: Proceedings of the Fourth International Conference on Sensor Technologies and Applications, SENSORCOMM '10, 2010, pp. 551–557.

[50] I. Das, J. Dennis, A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems, Structural optimization 14 (1) (1997) 63–69.

[51] M. Ehrgott, Multicriteria Optimization, Springer Science & Business Media, 2006.

Vitae

Alma A. M. Rahat is a Research Fellow at the University of Exeter, UK. He has a BEng (Hons) in Electronic Engineering from the University of Southampton, UK, and a PhD in Computer Science from the University of Exeter, UK. He has worked in the electronics industry as a product development engineer before starting his PhD. His research interests lie in fast hybrid single and multi objective optimisation methods, real-world problems and machine learning. Current research is on robust optimisation of wireless sensor networks under uncertainties and surrogate-assisted optimisation approaches for computationally expensive problems.

Richard M. Everson received the B.A. degree in Natural Science from Cambridge University, and the Ph.D. degree in applied mathematics from Leeds University, Leeds. He was with Brown University and with Yale University where he was engaged in fluid mechanics and data analysis problems, before moving to Rockefeller University to work on optical imaging and modeling of the visual cortex. He is now Professor of Machine Learning at the University of Exeter, Exeter. His research interests include machine learning and data science, especially applied to multi-objective optimization of large systems in the presence of uncertainty.

Jonathan E. Fieldsend is a Senior Lecturer in Computer Science at the University of Exeter. Most of his research, both theoretical and applied, is on the interface of machine learning and nature inspired computation. He has published over 70 peer-reviewed journal and conference articles. He is a member of the ACM Special Interest Group on Genetic and Evolutionary Computation (SIGEVO) and the IEEE Computational Intelligence Society (CIS). He is a vice-chair of the IEEE CIS Task Force on Data-Driven Evolutionary Optimization of Expensive Problems, and a member of the IEEE CIS Task Force on Multi-Modal Optimisation.