

PAPER • OPEN ACCESS

## Self-organization scheme for balanced routing in large-scale multi-hop networks

To cite this article: Mihai-Alin Badiu *et al* 2021 *J. Phys. A: Math. Theor.* **54** 045001

View the [article online](#) for updates and enhancements.



**IOP | ebooks™**

Bringing together innovative digital publishing with leading authors from the global scientific community.

Start exploring the collection—download the first chapter of every title for free.

# Self-organization scheme for balanced routing in large-scale multi-hop networks

Mihai-Alin Badiu<sup>1,\*</sup> , David Saad<sup>2</sup>  and Justin P Coon<sup>1</sup> 

<sup>1</sup> Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, United Kingdom

<sup>2</sup> Non-linearity and Complexity Research Group, Aston University, Birmingham, B4 7ET, United Kingdom

E-mail: [mihai.badiu@eng.ox.ac.uk](mailto:mihai.badiu@eng.ox.ac.uk), [d.saad@aston.ac.uk](mailto:d.saad@aston.ac.uk) and [justin.coon@eng.ox.ac.uk](mailto:justin.coon@eng.ox.ac.uk)

Received 9 June 2020, revised 23 November 2020

Accepted for publication 14 December 2020

Published 6 January 2021



CrossMark

## Abstract

We propose a self-organization scheme for cost-effective and load-balanced routing in multi-hop networks. To avoid overloading nodes that provide favourable routing conditions, we assign each node with a cost function that penalizes high loads. Thus, finding routes to sink nodes is formulated as an optimization problem in which the global objective function strikes a balance between route costs and node loads. We apply belief propagation (its min-sum version) to solve the network optimization problem and obtain a distributed algorithm whereby the nodes collectively discover globally optimal routes by performing low-complexity computations and exchanging messages with their neighbours. We prove that the proposed method converges to the global optimum after a finite number of local exchanges of messages. Finally, we demonstrate numerically our framework's efficacy in balancing the node loads and study the trade-off between load reduction and total cost minimization.

Keywords: distributed optimization, routing, load balancing, belief propagation, self-organization

## 1. Introduction

Large-scale wireless networks employing multi-hop transmissions are an integral component of the internet of things [1]. For example, such networks can consist of a massive number of sensors that collect data from the environment and send it to central controllers. In essence,

\* Author to whom any correspondence should be addressed.



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

the multi-hop transmission problem can be mapped onto a routing task on a network with a particular non-linear cost. Specifically, since in multi-hop networks each wireless node can relay other nodes' messages, it is highly relevant to direct the information flows from the source nodes to the destinations efficiently in terms of, e.g., energy consumption or reliability. Sending the flows along the minimum-cost paths towards the destinations potentially leads to overloading those nodes that provide favourable routes, which can cause quick battery depletion or decrease the resilience of the network against node failures [2–5]. Therefore, information should be routed through the network so as to minimize costs while trying to balance the node-loads. Moreover, given their scale, such networks must be designed to be self-organizing and adaptive.

There is a large body of work studying energy efficient routing protocols (see, e.g., the survey [6]). A typical objective is to maximize the network lifetime by maximizing the minimum lifetime over all nodes, where the lifetime of a node is defined as the ratio between its residual energy and its energy expenditure [2–4]. However, the network lifetime objective does not account for the total routing cost (total energy in this case) and thus can be inefficient in this respect, similar to minimum-cost routing being suboptimal for node balancing. It is therefore relevant to investigate objectives that favour solutions that are somewhere 'in-between' these two extremes.

At the heart of the model studied is the routing task itself, which is a hard computational problem and obtaining exact *globally optimal* solutions is possible only in small systems using established optimization methods, such as Integer Linear Programming, that do not scale well with the system size. Various heuristics are also being used to obtain suboptimal solutions for small to medium size networks in specific contexts, for instance, congestion pricing [7], differential games [8] and space-time network [9]. However, due to the inherent non-linearity and complexity of the problem they suffer from multiple sub-optimal solutions.

Statistical physics methods have been developed to study large-scale systems with non-linear interactions between system variables that exhibit macroscopic emergent behaviour. Disordered systems are arguably most suitable for studying optimization under constraints, where the disorder may refer to the underlying topology, i.e. the specific choice of edges in a graph, or strength of interaction between system constituents [10]. Dedicated techniques developed in this area and have been used to gain insight and devise optimization algorithms for real systems with similar characteristics, from error-correcting codes [11], to K-satisfiability [12], graph colouring [13, 14] and compressed sensing [15]. While the routing problem is slightly different due to the non-linear interaction between non-localized objects -the routes- it can also be addressed using similar methods. The cavity and replica methods have been employed to study generic routing problems [16, 17] as well as specific variants for problems that require node-disjoint [18] and edge-disjoint [19] routing, of particular relevance to the optimization of integrated circuits and optical communication networks, respectively. The cavity method also provides an effective routing algorithm in its microscopic manifestation.

In this work, we propose an algorithmic strategy for distributed multi-hop networking whereby the nodes coordinate and organize themselves so as to route the information to the destinations in an efficient and balanced way. To this end, we consider balanced routing as the minimization of a network objective function, which includes the overall cost of the routes (given by generic link costs) and an additional term that penalizes the node-loads. The objective function provides a tunable trade-off between total cost efficiency and fairness of the distribution of the node loads. The possible routes from source nodes to destinations are coupled in the objective function, which creates a competition for the shortest (i.e., least cost) routes to the sinks. To solve the optimization problem, we use the min-sum version of the belief

propagation (BP) method [10]. In this way, we obtain a distributed algorithm which finds globally optimal routes in a decentralized manner with low-complexity local computations and message exchanges between neighbouring nodes. We also show that the proposed method converges to the global optimum in a finite number of iterations.

## 2. Network model and problem formulation

We assume a data collection scenario in which a set  $\mathcal{V}_s = \{1, \dots, n\}$  of  $n$  nodes generate and/or relay information that has to be delivered to any subset of the  $m$  destination nodes (e.g., gateways, access points) in  $\mathcal{V}_d = \{n+1, \dots, n+m\}$ . The nodes in  $\mathcal{V}_s$  are simple devices with constrained resources (energy, memory, processing capabilities, etc) and can participate in routing each other's packets towards the destination nodes. Packets generated by a source node in  $\mathcal{V}_s$  can travel to a destination in  $\mathcal{V}_d$  over different routes; moreover, they can be delivered to different destination nodes.

We model the wireless network as a directed graph  $G(\mathcal{V}, \mathcal{E})$ , with  $\mathcal{V} = \mathcal{V}_s \cup \mathcal{V}_d$  and  $\mathcal{E}$  being the set of edges (links). An edge  $(i, j) \in \mathcal{E}$  indicates that node  $i$  can transmit to node  $j$  directly. For each  $i \in \mathcal{V}$ ,  $\mathcal{E}_i$  denotes the set of all edges incident to  $i$ , while  $\mathcal{E}_i^{\text{out}}$  and  $\mathcal{E}_i^{\text{in}}$  stand for the sets of its outgoing and respectively incoming edges. Node  $i \in \mathcal{V}_s$  generates information at a rate of  $r_i$  units (we assume a certain unit rate  $[r]$ ), where  $r_i \in \mathbb{N}$ ; if  $r_i = 0$ , the node is just a relay node. The capacity of edge  $e \in \mathcal{E}$  is  $u_e$  units,  $u_e \in \mathbb{N}_{>0}$ , such that the amount of flow  $x_e$  units carried by  $e$  satisfies  $0 \leq x_e \leq u_e$ . The assumption that the rates and capacities are integer multiples of  $[r]$  is not restrictive, because any set of rational numbers can be expressed in this way by finding an appropriate unit  $[r]$ . Moreover, if any of the rates and capacities have irrational values, it is necessary to convert them to rational numbers to represent them on a computer. We associate each link  $e \in \mathcal{E}$  with the weight  $c_e > 0$  representing the cost of transferring a unit over edge  $e$ . For example, the cost can be the transmit power required to ensure a certain data rate, the expected transmission count (ETX), or hop-count (when  $c_e = 1$ ). We further assume that the network is in the unsaturated traffic regime and packets are transferred between neighbours according to a medium access scheme, which we do not concern ourselves with here.

The routing solution space consists of those configurations  $\{x_e\}_{e \in \mathcal{E}}$  which satisfy the flow conservation constraints

$$\sum_{e \in \mathcal{E}_i^{\text{out}}} x_e - \sum_{e \in \mathcal{E}_i^{\text{in}}} x_e = r_i, \quad \text{for all } i \in \mathcal{V}_s, \quad (1)$$

and the capacity constraints  $0 \leq x_e \leq u_e$ , for all  $e \in \mathcal{E}$ . The two constraints ensure that all generated flows are delivered to the destinations such that edge flows do not exceed the respective capacities. We assume that the solution space is non-empty. The total cost of a configuration  $\{x_e\}_{e \in \mathcal{E}}$  is  $\sum_{e \in \mathcal{E}} c_e x_e$ . Furthermore, we define the load of node  $i$  to be the amount of flow  $\sum_{e \in \mathcal{E}_i^{\text{out}}} x_e$  it has to forward.

In general, there are many feasible configurations, each implying different sets of routes, path lengths, total costs, distribution of node loads, etc. A common objective is to minimize the total cost, which, as one can notice, turns data collection into a (linear) minimum cost network flow problem [20]. However, such an approach may yield solutions wherein some nodes that provide low-cost forwarding edges experience high loads. We are therefore interested in balancing the node loads in a cost-effective manner.

### 3. Proposed objective for load balancing

We seek a trade-off between minimization of the total cost and minimization of the loads of individual nodes. To this end, for each  $i \in \mathcal{V}_s$  we introduce the strictly-increasing convex function  $\phi_i : [0, \infty) \rightarrow \mathbb{R}$  to penalize the load of the  $i$ th node. The functions can vary over the nodes to reflect their different load-tolerances depending on residual energies, capabilities etc. Now, we formulate the optimization problem

$$\begin{aligned}
 & \underset{\mathbf{x} \in \mathbb{R}^{|\mathcal{E}|}}{\text{minimize}} && (1-w) \sum_{e \in \mathcal{E}} c_e x_e + w \sum_{i \in \mathcal{V}_s} \phi_i \left( \sum_{e \in \mathcal{E}_i^{\text{out}}} x_e \right) \\
 & \text{subject to} && \sum_{e \in \mathcal{E}_i^{\text{out}}} x_e - \sum_{e \in \mathcal{E}_i^{\text{in}}} x_e = r_i, \quad \forall i \in \mathcal{V}_s, \\
 & && 0 \leq x_e \leq u_e, \quad \forall e \in \mathcal{E},
 \end{aligned} \tag{2}$$

where  $w \in [0, 1]$  is a parameter that balances cost-efficiency and load minimization. When  $w = 0$ , we recover the linear minimum cost flow problem [20], which gives the most cost-efficient flow configuration; however, this setting usually does not provide well-balanced loads and therefore we focus on  $w > 0$ . The selection of its value depends on the particular application. We keep it generic in the following and discuss the selection for the example in the results section.

In the following, we assume that the functions  $\phi_i$  are piecewise-linear convex (PLC) with integral breakpoints, which is very convenient for obtaining a simple message-passing algorithm with provable convergence to the correct solution, as we show next in propositions 1 and 2. An example of such function is one that takes the value  $y^\alpha$ , with  $\alpha > 1$ , at each breakpoint  $y \in \mathbb{N}$  and varies linearly between consecutive breakpoints; the higher the value of  $\alpha$ , the stronger the load  $y$  is penalized. Such a choice provides a simple way to select the efficiency-fairness trade-off by tuning the parameter  $\alpha$ .

### 4. BP algorithm for balanced routing

BP is a generic message-passing algorithm for solving large-scale inference and optimization problems in graphical models. It has a distributed nature whereby the nodes of the graph perform simple local computations and exchange messages with their neighbours. While BP provides correct solutions when the underlying graph is a tree, its correctness and convergence cannot be generally guaranteed for graphs with cycles, with few exceptions [10, 21]. Nonetheless, for graphs with cycles, the BP heuristic often performs very well. In network problems, the min-sum algorithm is applied to find the shortest path between two nodes [22] or minimize path lengths and link congestion [16]. For the min-cost network flow problem with linear or PLC costs on edges, BP was shown in [21] to converge to the correct solution (if the solution is unique). Compared to [21], our objective (2) (with  $w > 0$ ) additionally includes node costs given by the PLC functions  $\{\phi_i\}$ ; therefore, the application of BP gives the novel algorithm described next<sup>3</sup>.

<sup>3</sup> Alternatively, by using the node splitting technique [20, p 41], one can transform (2) into a min-cost network flow problem with PLC costs on edges, which can be solved using BP [21, theorem 6.1]. However, BP on the transformed graph is different from algorithm 1 that we obtain here, see footnote 4.

For each node  $i \in \mathcal{V}_s$ , we define a function  $\psi_i$  that reflects the flow conservation constraint at node  $i$ , i.e., it maps each vector  $\mathbf{z} \in \mathbb{R}_+^{|\mathcal{E}_i|}$  of edge flows to

$$\psi_i(\mathbf{z}) = \begin{cases} 0, & \text{if } \sum_{e \in \mathcal{E}_i^{\text{out}}} z_e - \sum_{e \in \mathcal{E}_i^{\text{in}}} z_e = r_i, \\ \infty, & \text{otherwise.} \end{cases}$$

Furthermore, we define

$$f_i(\mathbf{z}) = \psi_i(\mathbf{z}) + w \phi_i \left( \sum_{e \in \mathcal{E}_i^{\text{out}}} z_e \right),$$

which additionally includes the load penalty for node  $i \in \mathcal{V}_s$ . On the contrary, destination nodes do not have any constraints and ‘accept’ any flows on their incoming edges, so we set  $f_i(\mathbf{z}) = 0$ , for any  $i \in \mathcal{V}_d$  and  $\mathbf{z} \in \mathbb{R}_+^{|\mathcal{E}_i|}$ . Next, we capture the cost and capacity constraint of edge  $e \in \mathcal{E}$  by introducing the function  $g_e : \mathbb{R} \rightarrow \mathbb{R} \cup \{\infty\}$  given by

$$g_e(z) = \begin{cases} (1-w)c_e z, & \text{if } 0 \leq z \leq u_e, \\ \infty, & \text{otherwise.} \end{cases}$$

We can now reformulate (2) as the equivalent problem

$$\underset{\mathbf{x} \in \mathbb{R}^{|\mathcal{E}|}}{\text{minimize}} \sum_{e \in \mathcal{E}} g_e(x_e) + \sum_{i \in \mathcal{V}} f_i(\mathbf{x}_{\mathcal{E}_i}), \quad (3)$$

where  $\mathbf{x}_{\mathcal{E}_i}$  includes those components of  $\mathbf{x}$  with indices in  $\mathcal{E}_i$ .

We apply the min-sum version of BP to solve (3). Given that each edge variable node has exactly two neighbour function nodes from the set  $\{f_i\}$ , we simplify the standard message updates by defining the messages (4) in algorithm 1. At iteration  $t$ , for each node  $i \in \mathcal{V}_s$  and incident edge  $e \in \mathcal{E}_i$ , where either  $e = (i, j) \in \mathcal{E}_i^{\text{out}}$  or  $e = (j, i) \in \mathcal{E}_i^{\text{in}}$ , the algorithm computes the message  $m_{i \rightarrow e}^t$ , which becomes an input to neighbour  $j$  at the next iteration. Since  $f_i$  is the zero function for all  $i \in \mathcal{V}_d$ , the messages computed by destination nodes do not change with  $t$  and thus are not updated.

Algorithm 1 has the following interpretation. Every node is seeking to determine the flow on each of its incident edges while satisfying its local flow conservation constraint and minimizing its load. The message  $m_{i \rightarrow e}^t(z)$  can be viewed as a local cost that node  $i$  attributes to allocating  $z$  units to edge  $e$ ; thus, the message is a function of the flow. For any  $z$ , the message update (4) includes: (i) the cost of sending flow  $z$  over edge  $e$  and (ii) the minimum cost of allocating flows to the rest of the edges that are incident to  $i$  such that flow conservation is ensured. The latter cost is the result of a local optimization, which looks for the feasible configuration of the flows on the incident edges that minimizes an objective function that includes the cost of the load of node  $i$  and the local costs (messages) estimated by the neighbouring nodes<sup>4</sup>. The message updates have low-complexity, as we show next.

<sup>4</sup>If using BP on the augmented graph obtained by node-splitting (see footnote 3), then  $m_{i \rightarrow e}^t$  effectively depends on messages over incoming (if  $e \in \mathcal{E}_i^{\text{out}}$ ) or outgoing (if  $e \in \mathcal{E}_i^{\text{in}}$ ) edges that were computed at  $t-2$ , i.e., it uses outdated information, which slows down convergence, as figure 2(c) shows.

**Algorithm 1.** Distributed algorithm for balanced routing.**Input:** The graph  $G(\mathcal{V}, \mathcal{E})$ , edge costs  $\{c_e\}$  and capacities  $\{u_e\}$ , data rates  $\{r_i\}$ , and parameters  $\alpha$  and  $w$ **Output:** Estimates  $\{\hat{x}_e\}_{e \in \mathcal{E}}$  of the optimal edge flows of (2)1: Initialize  $m_{i \rightarrow e}^0(z) = g_e(z)$ , for all  $i \in \mathcal{V}$ ,  $e \in \mathcal{E}_i$ ,  $z \in \mathbb{R}_+$ 2: **for**  $t = 1$  **to**  $T$  **do**3: For each  $i \in \mathcal{V}_s$  and  $e \in \mathcal{E}_i$ , update

$$m_{i \rightarrow e}^t(z) = g_e(z) + \min_{\tilde{\mathbf{z}} \in \mathbb{R}_+^{|\mathcal{E}_i|}: \tilde{z}_e = z} \left\{ f_i(\tilde{\mathbf{z}}) + \sum_{e' \in \mathcal{E}_i \setminus e} m_{k \rightarrow e'}^{t-1}(\tilde{z}_{e'}) \right\}, \quad (4)$$

for all  $z \in \mathbb{R}_+$ , where  $e' = (i, k)$  or  $(k, i)$ .4: **end for**5: For each  $e = (i, j) \in \mathcal{E}$ , compute the belief function

$$b_e^t(z) = m_{i \rightarrow e}^t(z) + m_{j \rightarrow e}^t(z) - g_e(z) \quad (5)$$

and determine its minimizer

$$\hat{x}_e^t = \arg \min_z b_e^t(z) \quad (6)$$

6: **return**  $\hat{\mathbf{x}}^t = (\hat{x}_e^t)_{e \in \mathcal{E}}$ 

**Proposition 1 (complexity).** For each  $i \in \mathcal{V}_s$ ,  $e \in \mathcal{E}_i$  and  $t \geq 1$ , the message  $m_{i \rightarrow e}^t$  is a PLC function with breakpoints in  $\{0, 1, \dots, u_e\}$ . The complexity of its update (4) is linear in the total capacity of the input and output edges of node  $i$  and logarithmic in  $|\mathcal{E}_i^{\text{in}}|$  and  $|\mathcal{E}_i^{\text{out}}|$ .

**Proof.** The proof is by induction on  $t$ . At  $t = 0$ , algorithm 1 initializes the messages to trivial PLC functions. Suppose at iteration  $t - 1$  all messages are PLC functions with integral breakpoints. We provide the proof for  $m_{i \rightarrow e}^t$  with  $e \in \mathcal{E}_i^{\text{out}}$ , as the case  $e \in \mathcal{E}_i^{\text{in}}$  is very similar. Let  $\psi_y^{(1)} : \mathbb{R}^{|\mathcal{E}_i^{\text{in}}|} \rightarrow \mathbb{R} \cup \{\infty\}$ ,  $y \in \mathbb{R}$ , be

$$\psi_y^{(1)}(\mathbf{z}) = \begin{cases} 0, & \text{if } \sum_{e' \in \mathcal{E}_i^{\text{in}}} z_{e'} + r_i = y, \\ \infty, & \text{otherwise,} \end{cases}$$

and define  $\psi_v^{(2)} : \mathbb{R}^{|\mathcal{E}_i^{\text{out}}| - 1} \times \mathbb{R} \rightarrow \mathbb{R} \cup \{\infty\}$ ,  $v \in \mathbb{R}$ , given by

$$\psi_v^{(2)}(\mathbf{z}, y) = \begin{cases} 0, & \text{if } y - \sum_{e' \in \mathcal{E}_i^{\text{out}} \setminus e} z_{e'} = v, \\ \infty, & \text{otherwise.} \end{cases} \quad (7)$$

Now, we define the function

$$h(y) = \min_{\tilde{\mathbf{z}} \in \mathbb{R}_+^{|\mathcal{E}_i^{\text{in}}|}} \left\{ \psi_y^{(1)}(\tilde{\mathbf{z}}) + \sum_{e' = (k, i) \in \mathcal{E}_i^{\text{in}}} m_{k \rightarrow e'}^{t-1}(\tilde{z}_{e'}) \right\} + w\phi_i(y). \quad (8)$$

The minimization in the rhs is a so-called interpolation of PLC functions whose complexity is logarithmic in the number of functions and linear in the total number of their linear pieces [21]. Since  $m_{k \rightarrow e'}^{t-1}$  has breakpoints in  $\{0, 1, \dots, u_{e'}\}$  and  $\phi_i$  is also PLC with integral breakpoints, it follows that the function  $h$  is itself PLC with integral breakpoints and at most  $U_i^{\text{in}}$  pieces, where  $U_i^{\text{in}} = \sum_{e' \in \mathcal{E}_i^{\text{in}}} u_{e'} + r_i$ ; moreover,  $h$  can be computed in  $O(U_i^{\text{in}} \log |\mathcal{E}_i^{\text{in}}|)$  operations. Now, we



write (4)

$$m_{i \rightarrow e}^t(z) = g_e(z) + \min_{\tilde{\mathbf{z}}, y} \left\{ \psi_z^{(2)}(\tilde{\mathbf{z}}, y) + h(y) + \sum_{e'=(i,k) \in \mathcal{E}_i^{\text{out}} \setminus e} m_{k \rightarrow e'}^{t-1}(\tilde{z}_{e'}) \right\}. \quad (9)$$

Given that  $h$  and the messages at  $t - 1$  are PLC with integral breakpoints, the interpolation in the second line gives again a PLC function; its computation takes  $O(U_i^e \log |\mathcal{E}_i^{\text{out}}|)$  operations, where  $U_i^e = \sum_{e' \in \mathcal{E}_i^{\text{out}} \setminus e} u_{e'}$ . The addition of  $g_e$ , which is linear in  $[0, u_e]$ , makes  $m_{i \rightarrow e}^t$  PLC with integral breakpoints.  $\square$

We establish that algorithm 1 outputs the optimal solution after a finite number of iterations.

**Proposition 2 (convergence).** *Suppose (2) has a unique optimal solution  $\mathbf{x}^*$ <sup>5</sup>. Then, there exists a finite integer  $T^*$  such that the output of Algorithm 1 satisfies  $\hat{\mathbf{x}}^t = \mathbf{x}^*$ , for any  $t \geq T^*$ .*

**Proof.** Although our objective function (2) is different than that of the min-cost network flow problem with linear (or PLC) edge costs, we can use the same proof strategy as in [21, theorem 4.1, theorem 6.1]. The difference is that we need to define an appropriate residual graph [20]. Denote by  $G(\mathbf{x})$  the residual graph of  $G(\mathcal{V}, \mathcal{E})$  with respect to the flow  $\mathbf{x} \in \mathbb{R}^{|\mathcal{E}|}$ .  $G(\mathbf{x})$  has the same vertices  $\mathcal{V}$ , while we define its edges and their costs as follows: for any  $e = (i, j) \in \mathcal{E}$ , if  $x_e < u_e$ , then  $e$  is also an edge in  $G(\mathbf{x})$  with capacity  $u_e - x_e$  and cost  $c_e^{\mathbf{x}} = (1 - w)c_e + w \lim_{z \rightarrow 0^+} (\phi_i(y + z) - \phi_i(y)) / z$ , where  $y = \sum_{e' \in \mathcal{E}_i^{\text{out}} \setminus e} x_{e'}$  is the load of node  $i$ ; if  $x_e > 0$ , then  $G(\mathbf{x})$  additionally includes the directed edge  $e' = (j, i)$  with capacity  $x_e$  and cost  $c_{e'}^{\mathbf{x}} = -(1 - w)c_e + w \lim_{z \rightarrow 0^-} (\phi_i(y + z) - \phi_i(y)) / z$ . At the unique optimal solution  $\mathbf{x}^*$ , all the directed cycles of the residual graph  $G(\mathbf{x}^*)$  must have positive costs (according to the negative cycle optimality criterion [20]). The proof relies on this property and follows the same steps as that of [21, theorem 4.1]; therefore we omit the details.  $\square$

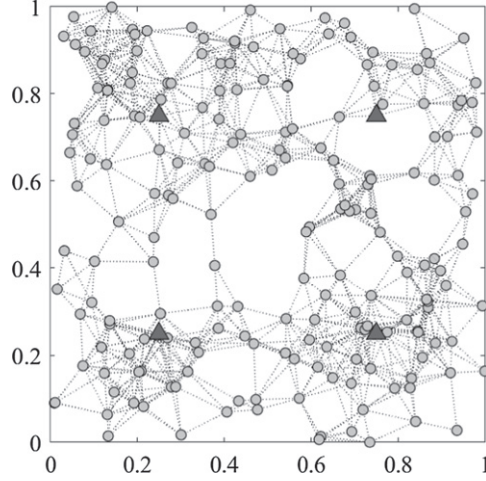
## 5. Numerical results

As an example, we consider a wireless sensor network consisting of  $n = 200$  sensors that send measurements to any of the  $m = 4$  sink nodes (collectors). The wireless network is modelled by a random geometric graph [23]. Specifically, the sensor nodes are independently and uniformly distributed inside the unit square, while the sink nodes are placed to evenly cover the area, as illustrated in figure 1. Any two nodes  $i$  and  $j$  that are spaced by less than  $\sqrt{\frac{2 \ln n}{\pi n}}$ , which is on the order of the critical connection range [24], are connected by the directed edges  $(i, j)$  and  $(j, i)$ . In each trial, a new network realization is generated (we discard the network realizations that are not connected). Then, we randomly select  $k$  sources out of the  $n$  nodes; the sources generate information at unit rate, while the remaining  $n - k$  nodes act as relays. The cost associated with each link is the ETX, which is drawn uniformly at random from the interval  $[1, 3]$ .

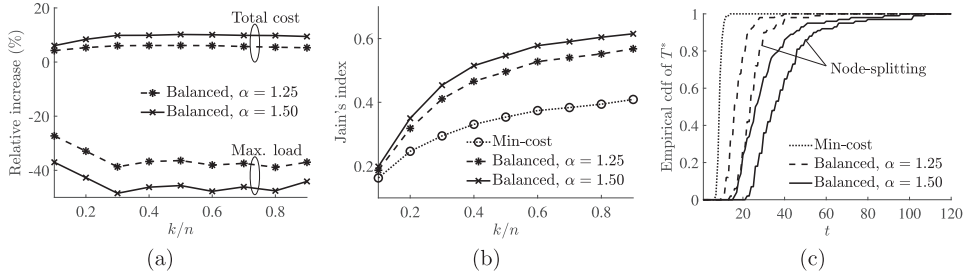
For the proposed balanced routing scheme, algorithm 1, we select the parameter  $w$  such that the two terms of the objective function (2) (representing the routing cost and the node load, respectively) are somewhat balanced. For this, we impose the condition that, when a node

<sup>5</sup> When the costs  $\{c_e\}$  are generic (e.g., random), it is highly likely that (2) has a unique solution. Otherwise, it is possible to add small noise to the costs such that the modified problem has a unique solution which very closely approximates the solution of the original problem [21].





**Figure 1.** A network instance generated from the considered random geometric graph model;  $n = 200$  nodes (circles) and  $m = 4$  sink nodes (triangles); the edges of the graph are represented by dotted lines.



**Figure 2.** Simulation results for  $n = 200$ ,  $m = 4$  and various fractions  $k/n$  of source nodes: (a) Improvement of the total cost and maximum load relative to minimum-cost routing; (b) Jain's fairness index for the node loads; (c) empirical cdf of the minimum number  $T^*$  of iterations required for algorithm 1 to converge when  $k/n = 0.3$ .

has a load of one unit which is forwarded over an edge of maximum cost, the two terms of the objective function that correspond to this node should be equal; i.e.,  $(1 - w)c_{\max} = w\phi(1) = w$ , which gives  $w = 0.75$ .

We evaluate the total cost, the maximum of the node loads  $\{y_i\}_{i=1}^n$ , the Jain's index,  $J = \frac{1}{n} \left( \sum_{i=1}^n y_i \right)^2 / \sum_{i=1}^n y_i^2 \in [1/n, 1]$ , as a measure of fairness in the distribution of the loads and the empirical distribution of the minimum number  $T^*$  of iterations required for algorithm 1 to converge. We compare the results obtained using our algorithm against minimum-cost routing which is instantiated by setting  $w = 0$  in algorithm 1. The results in figure 2 are obtained by averaging from 200 independent trials. In figure 2(a), we observe that balancing with  $\alpha = 1.25$  reduces the maximum load by 25%–40% compared to minimum-cost routing across all fractions of source nodes, while the total cost increases by no more than 6%. Increasing  $\alpha$  to 1.5 brings larger reduction of the maximum load, of about 40%–50%, and a higher relative total

cost of up to about 10%. Figure 2(b) shows that the balanced routing scheme provides significantly fairer load-distributions. As illustrated in figure 2(c), the number of iterations required to find a balanced solution is higher than for min-cost routing and increases with  $\alpha$ . We also evaluated BP on the graph transformed by node splitting (see footnotes 1 and 2) and, while it outputs the same solutions, it requires a higher number of iterations than our method, as shown in figure 2(c).

## 6. Conclusion

We formulated balanced routing in large-scale networks (such as internet of things) as optimization of an objective function that provides a tunable trade-off between total cost efficiency and fairness of the distribution of the node loads. In the proposed decentralized scheme, the nodes collectively find the globally optimal routing solution through low-complexity local computations and exchanges of messages with neighbours. The scheme provides significantly fairer solutions than minimum-cost routing at the expense of slightly increased total cost and higher number of required iterations.

There are several interesting directions to explore further, such as adapting the framework to specific models of energy consumption, including in the design the notions of reliability, trust among nodes and security, but also extending the framework to take into account the scheduling of the transmissions.

## Acknowledgments

This work was supported by the Independent Research Fund Denmark under Grant ID DFF-5054-00212, the Leverhulme Trust Grant No. RPG-2018-092, EPSRC programme Grant No. EP/R035342/1 TRANSNET (DS) and EPSRC Grant No. EP/N002350/1 ('Spatially Embedded Networks'). Most of the research was carried out during an extended visit of M-A Badiu to Aston University, while being affiliated with Aalborg University, Denmark.

## ORCID iDs

Mihai-Alin Badiu  <https://orcid.org/0000-0003-1581-5598>

David Saad  <https://orcid.org/0000-0001-9821-2623>

Justin P Coon  <https://orcid.org/0000-0002-9623-5087>

## References

- [1] Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M and Ayyash M 2015 *IEEE Commun. Surv. Tutorials* **17** 2347–76
- [2] Toh C-K 2001 *IEEE Commun. Mag.* **39** 138–47
- [3] Chang J-H and Tassiulas L 2004 *IEEE/ACM Trans. Networking* **12** 609–19
- [4] Madan R and Lall S 2006 *IEEE Trans. Wireless Commun.* **5** 2185–93
- [5] Vazifehdan J, Prasad R V and Niemegeers I 2014 *IEEE Trans. Mobile Comput.* **13** 434–47
- [6] Pantazis N A, Nikolidakis S A and Vergados D D 2013 *IEEE Commun. Surv. Tutorials* **15** 551–91
- [7] Carey M 1995 *Dynamic Congestion Pricing and the Price of FIFO Urban Traffic Networks* ed N H Gartner and G Improta (Berlin: Springer) pp 333–50
- [8] Bell M G H and Cassir C 2002 *Transp. Res. B* **36** 671–81
- [9] Yang H and Meng Q 1998 *Transp. Res. B* **32** 247–60

- [10] Mezard M and Montanari A 2009 *Information, Physics, and Computation* (New York: Oxford University Press)
- [11] Kabashima Y, Murayama T and Saad D 2000 *Phys. Rev. Lett.* **84** 1355–8
- [12] Mézard M, Parisi G and Zecchina R 2002 *Science* **297** 812–5
- [13] van Mourik J and Saad D 2002 *Phys. Rev. E* **66** 056120
- [14] Zdeborová L and Krzakala F 2007 *Phys. Rev. E* **76** 031131
- [15] Krzakala F, Mézard M, Sausset F, Sun Y F and Zdeborová L 2012 *Phys. Rev. X* **2** 021005
- [16] Yeung C H and Saad D 2012 *Phys. Rev. Lett.* **108** 208701
- [17] Yeung C H, Saad D and Wong K Y M 2013 *Proc. Natl Acad. Sci. USA* **110** 13717–22
- [18] Bacco C D, Franz S, Saad D and Yeung C H 2014 *J. Stat. Mech.* **P07009**
- [19] Altarelli F, Braunstein A, Dall'Asta L, De Bacco C and Franz S 2016 *PLOS ONE* **10** e0145222
- [20] Ahuja R K, Magnanti T L and Orlin J B 1993 *Network Flows: Theory, Algorithms, and Applications* (Englewood Cliffs, NJ: Prentice-Hall)
- [21] Gamarnik D, Shah D and Wei Y 2012 *Oper. Res.* **60** 410–28
- [22] Ruozzi N and Tatikonda S 2008 s-t paths using the min-sum algorithm 2008 *46th Annual Allerton Conference on Communication, Control, and Computing* pp 918–21
- [23] Haenggi M, Andrews J G, Baccelli F, Dousse O and Franceschetti M 2009 *IEEE J. Sel. Areas Commun.* **27** 1029–46
- [24] Gupta P and Kumar P R 1999 *Critical Power for Asymptotic Connectivity in Wireless Networks Stochastic Analysis, Control, Optimization and Applications* (Berlin: Springer) pp 547–66