SIAM J. APPL. MATH.
Vol. 83, No. 5, pp. 1832–1853
© 2023 Society for Industrial and Applied Mathematics

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

MINIMIZING CONGESTION IN SINGLE-SOURCE, SINGLE-SINK QUEUEING NETWORKS*

FABIAN YING†, ALISDAIR O. G. WALLIS‡, MASON A. PORTER§, SAM D. HOWISON†, AND MARIANO BEGUERISSE-DÍAZ†

Abstract. Motivated by the modeling of customer mobility and congestion in supermarkets, we study queueing networks with a single source and a single sink. We assume that walkers traverse a network according to an unbiased random walk, and we analyze how network topology affects the total mean queue size Q, which we use to measure congestion. We examine network topologies that minimize Q and provide proofs of optimality for some cases and numerical evidence of optimality for others. Finally, we present greedy algorithms that add edges to and delete edges from a network to reduce Q, and we apply these algorithms to a network that we construct using a supermarket store layout. We find that these greedy algorithms, which typically tend to add edges to the sink node, are able to significantly reduce Q. Our work helps improve understanding of how to design networks with low congestion and how to amend networks to reduce congestion.

Key words. queueing networks, random walks, congestion, human mobility, complex systems

MSC codes. 90B22, 60K20, 05C82

DOI. 10.1137/21M1457515

1. Introduction.

Understanding the interplay between dynamical processes on networks and the underlying network topology [18] is important for designing safe and efficient communication networks [16, 17], transportation networks [19, 23], power grids [5, 15], and more.

In this paper, we study a simple model of traffic on networks. There has been much work to investigate how the topology of a network affects traffic on it [1, 6]. We examine a scenario that is inspired by customer mobility in supermarkets [21, 22]. In [22], we proposed a model of customer mobility in supermarkets that is based on population-level mobility models; we examined congestion by placing a queueing system at each node of a network. In the present paper, we study a more abstract model of customer mobility. We consider open queueing networks [10] in which the customers are random walkers. We analyze how network topology affects congestion, which we measure using the total mean queue size Q. We focus on stationary queueing networks with a single source node (representing the entrance of a supermarket) and a single sink node (representing the till area and exit). Walkers (i.e., customers) enter a network at the source node and follow an unbiased random walk. At each node,

*Received by the editors November 5, 2021; accepted for publication (in revised form) February 24, 2023; published electronically September 15, 2023.
https://doi.org/10.1137/21M1457515
Funding: The work of the first author was supported by the EPSRC Centre For Doctoral Training in Industrially Focused Mathematical Modelling (grant EP/L015803/1) in collaboration with Tesco PLC. The work of the fifth author was supported by the Oxford–Emirates Data Science Lab.
†Mathematical Institute, University of Oxford, Oxford OX2 6GG, UK (fabian.m.ying@gmail.com, howison@maths.ox.ac.uk, marianob@spotify.com).
‡Tesco PLC, Tesco House, Shire Park, Kestrel Way, Welwyn Garden City, AL7 1GA, UK (Alisdair.Wallis@tesco.com).
§Department of Mathematics, University of California, Los Angeles, CA 90095 USA and Santa Fe Institute, Santa Fe, NM 87501 USA (mason@math.ucla.edu).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

walkers queue up to be served;[1] the service time at each node represents the delay due to congestion at that node. Walkers leave the network when they arrive at the sink node.

Our model is a simplified model of customer mobility in a supermarket; it neglects shopping intent, customer heterogeneity, and many other factors. However, such simple models are important for providing insights into a focal application. They also provide a starting point to explore more complicated models that capture more realistic aspects of a problem. Although we are motivated by customer mobility in supermarkets, one can adapt our model to study other applications (such as human traffic in airports, warehouses, shopping centers, and other public facilities) [7, 8, 12].

We model and measure congestion in our networks. We treat congestion as a condition that increases the journey time of customers and we seek to minimize the mean journey time. By Little's law [11], minimizing the mean journey time is equivalent to minimizing the total mean queue size Q. Because it is typically easier to directly calculate Q than to calculate the mean journey time, we measure congestion using Q.

Our model is a variant of the traffic-dynamics model of Arenas et al. [1, 6]. We consider random walks on networks [13] with a single source and a single sink, whereas every node in the Arenas et al. model is both a source and a sink. Arenas et al. focused primarily on minimizing the traffic capacity ρ_c (which, in our model, is equivalent to the maximum arrival rate of a node) [3] instead of minimizing Q. However, our methods are applicable both to Q and to the maximum arrival rate $\lambda_{\max}$. (See section 3.2 and our Supplementary Materials (supplement-ying.pdf [local/web 3.31MB]).)

We present our model in section 2. In our investigation of our model, we address the following two questions: (1) Which network topologies minimize Q? (2) Given a network, which edges should one add or delete to reduce Q?

We investigate the first question in section 3. We present an optimal directed network topology (which we denote by $G_{\bar{\mathcal{C}}_n}$) that minimizes Q over the set of networks with n nodes that do not have an edge from the source node to the sink node. In $G_{\bar{\mathcal{C}}_n}$, each node other than the source and sink has an incoming edge (i.e., an in-edge) from the source and an outgoing edge (i.e., an out-edge) to the sink. For small values of n, we perform numerical investigations to find optimal networks when there is an edge from the source to the sink. In this situation, we observe that the optimal topology is $G_{\bar{\mathcal{C}}_n}$ along with an edge from the source to the sink.

In section 4, we investigate the second question and present greedy algorithms for edge deletion and edge addition to reduce Q. We apply these algorithms to queueing networks that we generate using random-graph models and to a network from an actual supermarket store layout. We demonstrate that these algorithms are able to significantly reduce Q in all of these networks.

In section 5, we summarize our findings and outline several directions for future study. We give additional details about our work in the Supplementary Materials (supplement-ying.pdf [local/web 3.31MB]).

2. A single-source, single-sink open queueing network with unbiased random walkers. Consider a single-source, single-sink open[2] queueing network [10] that takes the form of a directed, unweighted network (i.e., a graph) $G = (V, E)$,

[1] For example, in the present context, an individual may wait for an opportunity to take an item from a supermarket shelf.

[2] A queueing network is *open* if walkers can enter and leave the network (so there is at least one source node and at least one sink node). By contrast, in a *closed* queueing network, walkers neither enter nor leave.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

where V is the set of nodes (i.e., vertices) and $E \subseteq V \times V$ is the set of edges. This network has $n = |V|$ nodes that we label from 1 (the source node) to n (the sink node). Walkers arrive at node 1 from outside the system at a rate $\lambda_{01} = \lambda$, which we call the *external arrival rate*. We assume that the sink node n has no out-edges, so it is a point of no return. This is sensible for most customers in a supermarket; once customers reach the tills, they do not tend to go back to other parts of a store.

We assume that the topology of G satisfies certain *reachability conditions*. Specifically, we assume that each node $i \in V$ is in both the out-component of node 1 and the in-component of node n. (The in-component and out-component of a node include the node itself.) These conditions ensure that walkers can reach each node and that all walkers eventually leave the system through the sink node. We summarize the reachability conditions with the relation

$$C_{\text{out}}(1) = C_{\text{in}}(n) = \{1, \dots, n\}\,, \tag{2.1}$$

where $C_{\text{in}}(i)$ and $C_{\text{out}}(i)$, respectively, denote the in-component and out-component of i. Therefore, for a fixed number n of nodes, we consider directed networks in the set

$$\mathcal{C}_n = \{G = (V, E) : |V| = n,\, C_{\text{out}}(1) = C_{\text{in}}(n) = \{1, \dots, n\},\, d_n^{\text{out}} = 0\}\,, \tag{2.2}$$

where d_i^{out} is the out-degree of node i. For any graph $G \in \mathcal{C}_n$, all nodes except n have at least one out-edge because they are in the in-component of node n. Therefore, $d_i^{\text{out}} \geq 1$ for all $i \in \{1, \dots, n-1\}$.

We assume that walkers traverse a network according to an unbiased random walk. The associated transition matrix $\boldsymbol{P}$ has entries

$$P_{ij} = \begin{cases} A_{ij}/d_i^{\text{out}}\,, & i \in \{1, \dots, n-1\} \\ 0\,, & i = n\,, \end{cases} \tag{2.3}$$

where A_{ij} is the (i,j)th entry of the adjacency matrix $\boldsymbol{A}$, whose entries are 1 if $(i,j) \in E$ (i.e., if there exists a directed edge from i to j in G) and 0 otherwise.

Each node i is a first-in-first-out (FIFO) single-server node with a fixed service rate μ_i. We assume that μ_i exceeds the arrival rate λ_i of i, as otherwise there is no stationary state. In our computations and in section 3, we assume that the service rates of the nodes are homogeneous, so $\mu_i = \mu$ for all i; however, we present our model in a more general form (i.e., with heterogeneous service rates). We consider the system at stationarity, thereby neglecting temporal variations in the external arrival rate and the service rate. In reality, congestion in supermarkets is more complicated. It can depend on the time of day (e.g., busy times, such as during lunch), the time of the year (e.g., Christmas season), or a combination of many other factors (e.g., during a weekend with good weather, many customers may rush to stock up on barbeque supplies), and so on. However, if the system reaches stationarity sufficiently fast (relative to the time scales of exogenous temporal variations), it is reasonable to describe the system by its stationary state. We summarize our model and its inputs and outputs in Figure 1.

2.1. Traffic equations. We calculate the arrival rate λ_i by solving the traffic equations [10]

$$\lambda_i = \delta_{1i}\lambda + \sum_{j=1}^{n-1} \lambda_j P_{ji}\,, \quad i \in \{1, \dots, n\}\,, \tag{2.4}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

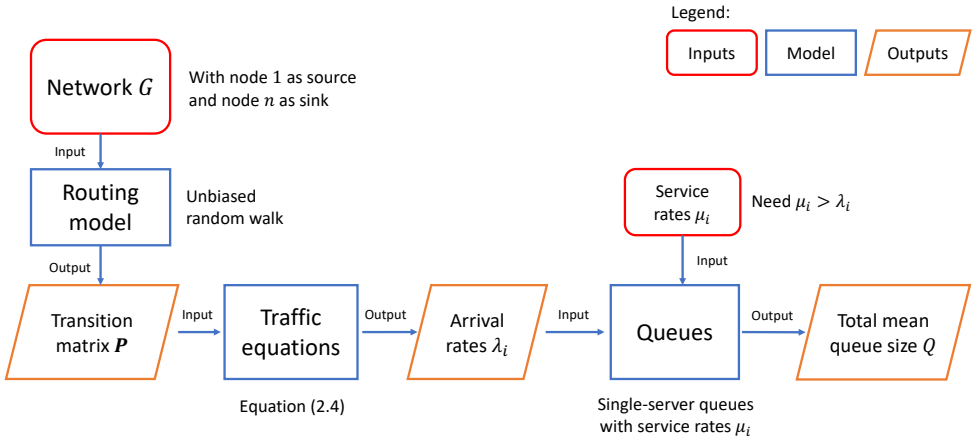


FIG. 1. *Summary of our model of an open queueing network. The model takes a network G and the service rates μ_i as inputs, and it outputs the total mean queue size Q.*

where the Kronecker delta δ_{1i} is 1 if $i=1$ and is 0 for all other values of i. Equation (2.4) is linear in λ_i, so we can rescale the values λ_i and μ_i by dividing by λ. Therefore, without loss of generality, we set $\lambda=1$ so that equation (2.4) becomes

$$\lambda_i = \delta_{1i} + \sum_{j=1}^{n} \lambda_j P_{ji}\,, \quad i \in \{1,\dots,n\}\,. \tag{2.5}$$

In matrix form, we write (2.5) as

$$\left(\boldsymbol{I} - \boldsymbol{P}^T\right)\boldsymbol{l} = \boldsymbol{b}\,, \tag{2.6}$$

where $\boldsymbol{l} = (\lambda_1,\dots,\lambda_n)^T$ and $\boldsymbol{b} = (1,0,\dots,0)^T$. There is a unique, positive solution to (2.5) [10], so $\boldsymbol{I} - \boldsymbol{P}^T$ is invertible and

$$\boldsymbol{l} = (\boldsymbol{I} - \boldsymbol{P}^T)^{-1}\boldsymbol{b}\,, \tag{2.7}$$

where $\lambda_i > 0$ for all $i \in \{1,\dots,n\}$. We know that $\lambda_n = 1$ because the departure rate from the system must equal the normalized external arrival rate (which is equal to 1). The arrival rates λ_i are independent of the service rates μ_i (and they depend only on the network topology) as long as $\mu_i > \lambda_i$ for all i. In other words, the rate at which customers arrive at node i does not depend on how fast it can serve customers (as long as it is sufficiently fast), so it does not depend on the level of congestion (as measured by the queue size) at i.

2.2. Total mean queue size Q. The mean queue size Q_i at each node i at stationarity is [10]

$$Q_i = \frac{\lambda_i}{\mu_i - \lambda_i}\,, \quad i \in \{1,\dots,n\}\,. \tag{2.8}$$

The function Q_i increases with λ_i and decreases with μ_i for $\lambda_i \in [0,\mu_i)$. That is, increasing the arrival rate or decreasing the service rate results in longer queues. The total mean queue size is

$$Q = \sum_{i=1}^{n} Q_i = \sum_{i=1}^{n} \frac{\lambda_i}{\mu_i - \lambda_i}\,. \tag{2.9}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

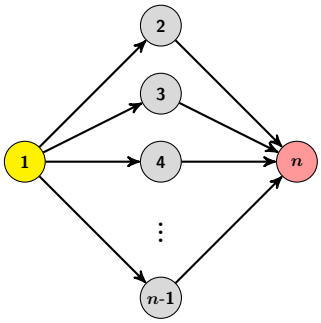


FIG. 2. $G_{\bar{\mathcal{C}}_n}$: *A Q-optimal network over* $\bar{\mathcal{C}}_n$.

3. Network topologies that minimize the total mean queue size Q. We seek network topologies in $\mathcal{C}_n$ (or in some subset of $\mathcal{C}_n$) that minimize the total mean queue size Q when all nodes have the same service rate $\mu > 1$ (i.e., when $\mu_i = \mu > 1$ for all i). As defined in equation (2.2), $\mathcal{C}_n$ is the set of networks such that (1) every node is reachable from the source node, (2) the sink node is reachable from any other node, and (3) the sink node has no out-edges. Given that $\lambda_n = 1$, we only consider service rates $\mu > 1$, because otherwise there is no stationary state for any queueing network. For a fixed $\mu > 1$, we calculate the total mean queue size Q of a network $G \in \mathcal{C}_n$ using the formula

$$Q = Q(G,\mu) = \begin{cases} \sum_{i=1}^{n} \frac{\lambda_i}{\mu - \lambda_i}, & \mu > \lambda_i \text{ for all } i \\ \infty, & \text{otherwise}, \end{cases} \tag{3.1}$$

where $\lambda_i = \lambda_i(G)$ is the arrival rate of node i in G (see equation (2.7)).

For any set $\mathcal{G} \subseteq \mathcal{C}_n$ of graphs, we say that $G \in \mathcal{G}$ is *Q-optimal over* $\mathcal{G}$ if, for any $\mu > 1$ and for all graphs $G' \in \mathcal{G}$, we have $Q(G,\mu) \le Q(G',\mu)$. That is, G is Q-optimal over $\mathcal{G}$ if, for any service rate $\mu > 1$, there is no other graph $G' \in \mathcal{G}$ with a strictly smaller total mean queue size.

We consider the following sets $\mathcal{G}$ of graphs:

- $\mathcal{C}_n$: No additional constraints.
- $\bar{\mathcal{C}}_n = \{G = (V,E) \in \mathcal{C}_n : (1,n) \notin E\}$: No edge from node 1 to node n.
- $\mathcal{U}_n = \{G = (V,E) \in \mathcal{C}_n : (i,j) \in E \text{ with } j \neq n \implies (j,i) \in E\}$: "Undirected" networks in $\mathcal{C}_n$. (All edges are bidirectional except for edges that are incident to n.)

We primarily consider $\bar{\mathcal{C}}_n$ because we are able to prove Q-optimality over $\bar{\mathcal{C}}_n$. We abuse terminology and use the adjective *undirected* for any network in $\mathcal{U}_n$ because such a network is equivalent to an undirected network with an additional constraint that any walker on it leaves after being served at node n. (In other words, node n remains a point of no return.)

A key result of this section is the following theorem.

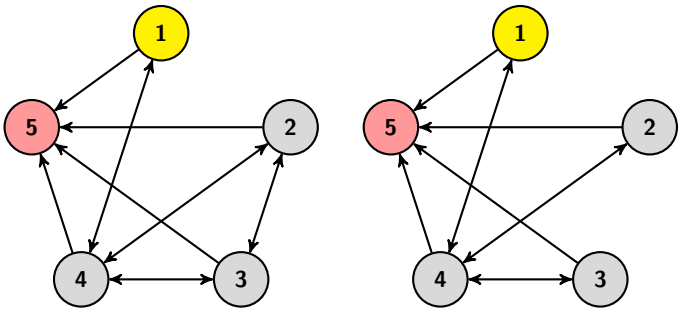
THEOREM 3.1. *For $n \ge 3$, the network $G_{\bar{\mathcal{C}}_n} = (V,E)$ with edge set $E = \{(1,i) : i \in \{2,\dots,n-1\}\} \cup \{(i,n) : i \in \{2,\dots,n-1\}\}$ (see Figure 2) is Q-optimal over $\bar{\mathcal{C}}_n$.*

We give the proof of Theorem 3.1 in Appendix A.

3.1. Properties of Q-optimal networks. In the following two propositions, we give necessary conditions for a network to be Q-optimal. These conditions relate Q to the maximum arrival rate $\lambda_{\max} := \max_i \lambda_i$ and the total arrival rate $\lambda_{\text{total}} := \sum_i \lambda_i$.

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.



(a) *Network* 1, *which has the arrival rates* $\left(\frac{6}{5}, \frac{3}{10}, \frac{3}{10}, \frac{4}{5}, 1\right)$ *for nodes* $(1, 2, 3, 4, 5)$.

(b) *Network* 2, *which has the arrival rates* $\left(\frac{6}{5}, \frac{1}{5}, \frac{1}{5}, \frac{4}{5}, 1\right)$ *for nodes* $(1, 2, 3, 4, 5)$.

FIG. 3. *The two networks that minimize* $\lambda_{\max}$ *over* $\mathcal{U}_5$. *The arrival rate of each node of network* 2 *is no larger than the arrival rate of the corresponding node of network* 1. *Therefore, network* 2 *has a smaller total mean queue size* Q *than network* 1 *for any* $\mu > 6/5$. *This illustrates that a network that minimizes* $\lambda_{\max}$ *does not necessarily minimize* Q.

PROPOSITION 3.2. *Suppose that* G *minimizes* Q *over all graphs in some set* $\mathcal{G}$ *for all values of* μ *with* $\mu \le \mu_{\max}$ *for some* $\mu_{\max} > \lambda_{\max}(G)$, *where* $\lambda_{\max}(G)$ *is the maximum arrival rate of* G. *For all* $G' \in \mathcal{G}$, *we then have*

$$\lambda_{\max}(G) \le \lambda_{\max}(G'). \tag{3.2}$$

That is, G *minimizes the maximum arrival rate* $\lambda_{\max}$ *over* $\mathcal{G}$. *Consequently, if* G *is* Q*-optimal over* $\mathcal{G}$, *then* G *minimizes the maximum arrival rate* $\lambda_{\max}$ *over all graphs in* $\mathcal{G}$.

Proof. We proceed by contradiction. Suppose that G minimizes Q for all values of μ that satisfy $\mu \le \mu_{\max}$ with $\mu_{\max} > \lambda_{\max}(G)$, and suppose that G does not minimize $\lambda_{\max}$ over $\mathcal{G}$. There is then a graph G' with a strictly lower maximum arrival rate $\lambda'_{\max} < \lambda_{\max}$. Choose any $\mu \in (\lambda'_{\max}, \lambda_{\max}]$. Let Q and Q' be the total mean queue sizes of G and G', respectively. Because G minimizes Q when $\mu < \lambda_{\max} \le \mu_{\max}$, it follows that $Q \le Q'$. However, the total mean queue size Q of G is infinite (because $\mu < \lambda_{\max}$), whereas the total mean queue size of G' is finite (because $\mu > \lambda'_{\max}$), so $\infty = Q > Q'$, which is impossible. ∎

The converse of Proposition 3.2 holds only when there is a unique network that minimizes the maximum arrival rate $\lambda_{\max}$. In this case, a network that minimizes Q for sufficiently small[3] values of μ also minimizes $\lambda_{\max}$ and is the unique minimizer of $\lambda_{\max}$. When there are multiple networks that minimize the maximum arrival rate $\lambda_{\max}$, the converse of Proposition 3.2 is not true in general (see Figure 3).

The next proposition gives another necessary condition for a network to be Q-optimal.

PROPOSITION 3.3. *A graph* G *minimizes* Q *over all graphs in some set* $\mathcal{G}$ *for all values of* μ *such that* $\mu > \mu_{\min}$ *for some* $\mu_{\min} > 0$ *if and only if*

$$\lambda_{\text{total}}(G) \le \lambda_{\text{total}}(G') \tag{3.3}$$

[3]The parameter μ is "sufficiently small" when $\mu \le \mu_{\max}$ for some $\mu_{\max} > \lambda_{\max}(G)$. The lower bound $\lambda_{\max}(G)$ of $\mu_{\max}$ is necessary because Q is infinite for all $G \in \mathcal{G}$ if μ is smaller than the minimum of the individual maximum arrival rates of these graphs. When all $G \in \mathcal{G}$ have infinite Q, then any G minimizes Q in the set $\mathcal{G}$, but the maximum arrival rate can be different in different graphs. Therefore, in this case, a network that minimizes Q does not necessarily minimize $\lambda_{\max}$.

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

for all $G' \in \mathcal{G}$, where $\lambda_{\text{total}}(G) = \sum_i \lambda_i$ is the total arrival rate of G. That is, G minimizes Q for all sufficiently large values of μ if and only if G minimizes λ_{total}. Consequently, if G is Q-optimal over $\mathcal{G}$, then G minimizes the total arrival rate λ_{total} over all graphs in $\mathcal{G}$.

Proof. For sufficiently large μ, we have that $\lambda_i/(\mu - \lambda_i) \approx \lambda_i/\mu$ for any node i because the arrival rate λ_i is independent of μ. Consider the objective function μQ. For a fixed value of μ, a network minimizes Q if and only if it minimizes μQ. For sufficiently large μ, we have

$$\mu Q = \mu \sum_i \frac{\lambda_i}{\mu - \lambda_i} = \sum_i \lambda_i + \mathcal{O}\left(\frac{\sum_i \lambda_i^2}{\mu}\right) = \lambda_{\text{total}} + \mathcal{O}\left(\frac{\sum_i \lambda_i^2}{\mu}\right), \tag{3.4}$$

which approaches λ_{total} as $\mu \to \infty$. Therefore, if G does not minimize λ_{total}, there is some graph G' with a smaller value of λ'_{total} and hence a smaller value of μQ (and thus Q) when μ is sufficiently large, which is a contradiction. Conversely, if G does not minimize Q for sufficiently large values of μ, it also does not minimize μQ for sufficiently large values of μ, so equation (3.4) implies that it does not minimize λ_{total}. ∎

Propositions 3.2 and 3.3 describe properties of Q-optimal networks (over some set $\mathcal{G}$ of graphs) for extreme values of μ. To minimize Q when μ is small (i.e., in the *small-μ regime*), a Q-optimal network must have the smallest $\lambda_{\max}$ of all graphs in $\mathcal{G}$. To minimize Q when μ is sufficiently large (i.e., in the *large-μ regime*), a Q-optimal network must have the smallest λ_{total} of all graphs in $\mathcal{G}$.

Propositions 3.2 and 3.3 imply that $G_{\bar{\mathcal{C}}_n}$ (a Q-optimal network over $\bar{\mathcal{C}}_n$) minimizes $\lambda_{\max}$ and λ_{total} over $\bar{\mathcal{C}}_n$. Proposition 3.3 also states that minimizing Q in the large-μ regime is equivalent to minimizing λ_{total}. Although Proposition 3.2 does not state that minimizing Q in the small-μ regime is equivalent to minimizing $\lambda_{\max}$, it allows us to identify the possible candidate networks that minimize Q in the small-μ regime. These candidates are the networks that minimize $\lambda_{\max}$.

3.2. Extensions. We extend Theorem 3.1 to other objective functions and other types of queues in sections SM1 and SM2 of the Supplementary Materials.

3.3. Numerical evidence for the Q-optimality of $G_{\mathcal{C}_n}$ over $\mathcal{C}_n$. When we consider networks in $\mathcal{C}_n$ instead of $\bar{\mathcal{C}}_n$ (i.e., when we allow edges between nodes 1 and n), we conjecture that the graph $G_{\mathcal{C}_n}$ in Figure 4 is Q-optimal over $\mathcal{C}_n$. In section SM3 of the Supplementary Materials, we show that proving that

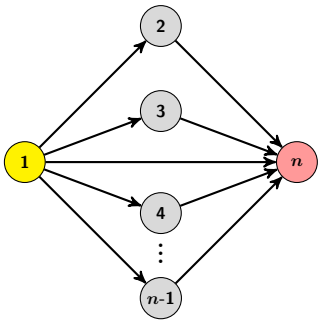


FIG. 4. *The graph $G_{\mathcal{C}_n}$: A conjectured Q-optimal network over $\mathcal{C}_n$.*

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

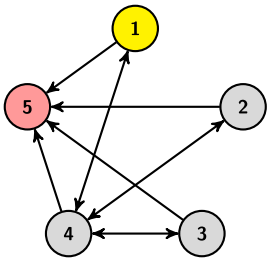
Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

$$\sum_{i=2}^{n-1} \lambda_i \geq 1 - \frac{1}{n-1} \tag{3.5}$$

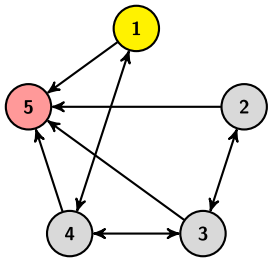
for all $G \in \mathcal{C}_n$ is sufficient to guarantee the Q-optimality of $G_{\mathcal{C}_n}$ over $\mathcal{C}_n$. The inequality (3.5) depends only on the arrival rates of the graphs $G \in \mathcal{C}_n$.

We do not have a proof of (3.5), but we have verified by exhaustive enumeration that all networks with 7 or fewer nodes satisfy (3.5). For larger network sizes (up to $n = 100$ nodes), we employ a simulated-annealing (SA) algorithm to find networks with minimal $\sum_{i=2}^{n-1} \lambda_i$. The SA algorithm did not find any networks with smaller values of $\sum_{i=2}^{n-1} \lambda_i$, and the values of $\sum_{i=2}^{n-1} \lambda_i$ of the networks that we obtained using SA are close to $1 - 1/(n-1)$. For more details about the SA algorithm and our results of using it, see section SM4.

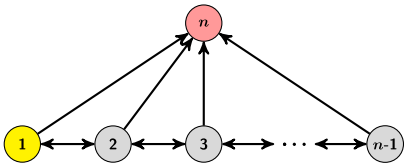
3.4. Q-optimality of undirected networks. When we consider the set $\mathcal{U}_n$ of undirected networks with $n \geq 5$, we observe behavior that is different than that for directed networks. For $n \in \{5, 6, 7\}$, we find using exhaustive enumeration that there are no Q-optimal networks over $\mathcal{U}_n$. Instead, different networks minimize Q for different values of the homogeneous service rate μ. For example, when $n = 5$, the network that minimizes Q over $\mathcal{U}_5$ when $\mu = 2$ (see Figure 5a) is different than the network that minimizes Q when $\mu = 2.5$ (see Figure 5b). When $n = 6$ and $n = 7$, we again find that which network minimizes Q depends on the service rate μ. For networks with 3 or 4 nodes, there exist Q-optimal networks (see section SM5).



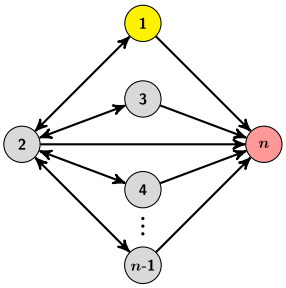
(a) *Minimizes Q when $\mu = 2$.*



(b) *Minimizes Q when $\mu = 2.5$.*



(c) $G_{\mathcal{U}_n}^{\lambda_{\text{total}}}$*: A conjectured undirected network (with $n \geq 3$ nodes) that uniquely minimizes λ_{total} over $\mathcal{U}_n$.*



(d) $G_{\mathcal{U}_n}^{\lambda_{\max}}$*: A conjectured undirected network (with $n \geq 3$ nodes) that minimizes $\lambda_{\max}$ over $\mathcal{U}_n$.*

FIG. 5. (a, b) *We show networks that minimize Q over $\mathcal{U}_5$ for* (a) $\mu = 2$ *and* (b) $\mu = 2.5$. (c, d) *We show conjectured undirected networks,* (c) $G_{\mathcal{U}_n}^{\lambda_{\text{total}}}$ *and* (d) $G_{\mathcal{U}_n}^{\lambda_{\max}}$*, that minimize λ_{total} and $\lambda_{\max}$, respectively, over $\mathcal{U}_n$.*

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

To understand why there may not exist a Q-optimal network over $\mathcal{U}_n$ for $n \geq 5$, we note that Propositions 3.2 and 3.3 imply that a Q-optimal network over $\mathcal{G}$ can exist only if it minimizes both $\lambda_{\max}$ and λ_{total}. When $\mathcal{G} = \mathcal{U}_n$ with $n \geq 5$, we conjecture that there does not exist a network that minimizes both $\lambda_{\max}$ and λ_{total}. For $\mathcal{U}_5$, we find by exhaustive enumeration that the network in Figure 5b minimizes λ_{total} (and is the unique minimizer), but it does not minimize $\lambda_{\max}$. In Figure 5a, we show a network[4] that minimizes $\lambda_{\max}$. For general n, we postulate two conjectures:

1. The network $G_{\mathcal{U}_n}^{\lambda_{\text{total}}}$ (see Figure 5c) uniquely minimizes λ_{total} over $\mathcal{U}_n$ for $n \geq 3$.
2. The network $G_{\mathcal{U}_n}^{\lambda_{\max}}$ (see Figure 5d) is a network with minimum $\lambda_{\max}$ over $\mathcal{U}_n$ for $n \geq 3$.

In section SM6, we show that $G_{\mathcal{U}_n}^{\lambda_{\text{total}}}$ does not minimize $\lambda_{\max}$ over $\mathcal{U}_n$ for $n \geq 5$. Therefore, if both of these conjectures are true, there does not exist a network that minimizes both $\lambda_{\max}$ and λ_{total}, which in turn implies that there are no Q-optimal networks over $\mathcal{U}_n$ for $n \geq 5$. We have verified these conjectures for all $n \in \{3, \ldots, 7\}$. In section SM4.2.2, we present numerical evidence of their validity for larger values of n.

4. Reducing Q in undirected networks by adding or deleting edges. We now examine algorithms to reduce the total mean queue size Q in networks. Because supermarket store networks are (predominantly) undirected, we present our results on queueing networks only for networks in the set $\mathcal{U}_n$ (i.e., networks that are undirected, except for edges that are incident to the sink node). However, it is straightforward to adapt our approach to directed networks.

In most real-world situations, it is infeasible to rewire a network into an optimal one (or a conjectured optimal one) like the ones that we presented in section 3. In practice, one typically can make only local perturbations to a network. The simplest examples of network perturbations are additions or deletions of individual edges or nodes [3]. For example, in an internet network (at the router level), adding connections between routers (which are the nodes of the network) is costly for internet service providers (ISPs), so it is useful for ISPs to identify the best edges that they can add to improve traffic capacity [4, 9]. In street networks, some streets are closed (corresponding to edge deletions) at rush hours to alleviate congestion [3, 14].

We fix the number of nodes and examine *edge perturbations*, which consist of adding or deleting edges. We consider the following question: Given an unbiased random walk on a single-source, single-sink open queueing network $G = (V, E)$, how do we add and delete edges to reduce Q? For the application to navigation in supermarkets, large changes in a store network are generally undesirable and disruptive. We do not consider perturbations that consist of adding or removing nodes, because it is not easy to expand most supermarkets and typically it is undesirable to remove shelf space.

We propose a greedy algorithm (which we call the GREEDY algorithm) that adds and deletes edges from a graph to reduce Q. We also consider two variants of the GREEDY algorithm: the GREEDY-ADD algorithm (which considers only edge additions) and the GREEDY-DELETE algorithm (which considers only edge deletions). We test the performance of the three algorithms on queueing networks with a topology that we draw from random-graph models with source and sink nodes that we choose at random (see section 4.3). We also test their performance on an actual store network. When we generate random graphs, we choose parameters so that the models typically generate sparse networks, in which the number of edges is much smaller than

[4]This network (see Figure 5a) is not the unique network that minimizes $\lambda_{\max}$ over $\mathcal{U}_n$. There are two distinct networks in $\mathcal{U}_5$ that achieve the minimum $\lambda_{\max}$ value of 1.2. We show them in Figure 3.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

the number of nonedges. We do this because supermarket store networks (and most other spatial networks [2]) are sparse. We demonstrate numerically that our three greedy algorithms are able to significantly reduce Q for all considered random-graph models and for the store network. The GREEDY algorithm has the best performance of these three algorithms.

4.1. Mathematical setup. Given an undirected[5] and unweighted network $G = (V, E)$, we define an *edge perturbation* $\Delta e = \Delta(i, j)$ on G to be the process of adding an undirected edge $e = (i, j)$ to G if $e \notin E$ or the process of deleting an undirected edge $e = (i, j)$ if $e \in E$. In the former case, Δe is an *edge addition*; in the latter case, Δe is an *edge deletion*. We say that an edge perturbation Δe to G is *valid* if the graph that we obtain by applying Δe to G satisfies our reachability conditions. Otherwise, we say that that edge perturbation is *invalid*. In a graph G that satisfies the reachability conditions, all edge additions are valid because adding edges cannot lead to a violation of them.

4.2. The GREEDY algorithm. The intuition behind the GREEDY algorithm is to try to minimize Q by repeatedly applying particular edge perturbations, which we select to give the best "local" decreases in Q, to a queueing network G. For each possible perturbation Δe, we compute the total mean queue size Q' of the graph G' that we obtain after performing the edge perturbation Δe on G. We rank each perturbation in increasing order of $\Delta Q = Q' - Q$ or, equivalently, in increasing order of Q'. (We break ties uniformly at random.) For any G' that does not satisfy the reachability conditions, we define Q' to be infinity. An undirected network G satisfies the reachability conditions if and only if both the graph G and the subgraph of G that is induced on the node set $\{1, \ldots, n-1\}$ are connected. Therefore, edge deletions are the only edge perturbations that may cause G' to violate the reachability conditions (if the original network G satisfies the reachability conditions).

At each iteration k of the GREEDY algorithm, we perform K_k edge perturbations for some integer K_k. We apply edge perturbations one at a time according to the ranking (in terms of ΔQ) of the edge perturbations; we skip any invalid perturbations.[6] We allow edge perturbations that increase Q, so Q does not necessarily decrease monotonically throughout the optimization procedure. After performing K_k edge perturbations, we recompute the ranking of all edge perturbations for the current graph G. We then apply K_{k+1} edge perturbations in iteration $k+1$. We repeat this procedure of applying edge perturbations and recalculating edge-perturbation rankings until we have performed T edge perturbations (for some predetermined integer $T \geq 1$).[7] We summarize the GREEDY algorithm in Algorithm 4.1.

When $K_k = 1$ for all k, we apply, at each iteration, an edge perturbation that results in the largest decrease in Q (or the smallest increase in Q, if all perturbations increase Q); we recalculate the ranking after each perturbation. Calculating the rankings of the permissible edge perturbation is computationally expensive, as it requires solving a linear system to obtain ΔQ for each permissible edge perturbation in $C_p(G)$. Larger values of K_k reduce the number of such calculations and thereby

[5] In our usage of the word "undirected", recall (see section 3) that the sink node n has no out-edges and that all other edges are bidirectional.

[6] When $K_k \geq 2$, we recompute the ranking only after K_k edge perturbations. After applying the first edge perturbation, subsequent edge perturbations may result in a graph that violates our reachability conditions, so we skip any that do so. In this case, we consider additional edge perturbations so that we perform K_k edge perturbations in total, provided that there are at least K_k valid edge perturbations.

[7] Therefore, the final iteration can have fewer than K_k edge perturbations.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Algorithm 4.1. The GREEDY algorithm for reducing Q using edge perturbations.

procedure GREEDY
Input: Network $G = (V, E)$, service rates μ_i, total number T of edge perturbations, number K_k of edge perturbations between recalculating rankings for each iteration k
Output: Network G with up to T edge perturbations
Initialize:
    Calculate ΔQ for each edge perturbation Δe
        ▷ (There are $\binom{V}{2}$ possible edge perturbations.)
    Rank all edge perturbations in increasing order of ΔQ and save them in the list R
    num_edges_perturbed $= 0$
    iteration_number $= 1$
Algorithm:
    while `True` **do**
        num_edges_perturbed_in_loop $= 0$
        Set $K = K_k$, where $k =$ *iteration_number*
        while *num_edges_perturbed_in_loop* $< K$ **do**
            Take first edge perturbation $\Delta e = \Delta(i, j)$ in R and remove it from R
            if edge $e = (i, j)$ already exists in G **then**
                $G' \leftarrow G$ with e removed
            else
                $G' \leftarrow G$ with e added
            if G' satisfies the reachability conditions **then**
                $G \leftarrow G'$
                Increment *num_edges_perturbed_in_loop* by 1
                Increment *num_edges_perturbed* by 1
            else
                continue ▷ Skip because it is not a valid graph
            if *num_edges_perturbed* $\geq T$ or R is empty **then**
                return G
        Recalculate ΔQ for each edge perturbation Δe
        Rank all edge perturbations in increasing order of ΔQ and save them in the list R
        Increment *iteration_number* by 1

reduce the computational cost. However, when $K_k \geq 2$, the second edge perturbation (and any subsequent ones) in each iteration may not give the largest decrease in Q, because the ranking of the edges is based on the network before we apply the first edge perturbation. We expect that the GREEDY algorithm performs worse (i.e., decreases Q less) when we use larger values of K_k and that there is a trade-off between performance and computation time. One can also calculate the value of ΔQ only for a fraction f_{ep} of all edge perturbations. Suppose that we choose these edge perturbations uniformly at random whenever we determine a ranking of perturbations. Smaller values of f_{ep} yield faster computation times (which may be desirable for large networks) in exchange for potentially worse performance (i.e., larger final values of Q). In the present paper, we rank all edge perturbations (i.e., $f_{\text{ep}} = 1$).

To compare the relative effectiveness of edge additions and edge deletions, we also consider two variants—the GREEDY-ADD and GREEDY-DELETE algorithms—

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

TABLE 1
Our edge-perturbation algorithms.

Algorithm	Description
GREEDY	At each iteration k, it performs K_k valid edge perturbations that most reduce Q.
GREEDY-ADD	At each iteration k, it performs K_k edge additions (which are always valid) that most reduce Q.
GREEDY-DELETE	At each iteration k, it performs K_k valid edge deletions that most reduce Q.

of Algorithm 4.1. The GREEDY-ADD algorithm allows only edge additions and the GREEDY-DELETE algorithm allows only edge deletions. See section SM7 for complete specifications of these algorithms. We summarize the GREEDY algorithm and its two variants in Table 1.

4.3. Random network topologies for queueing networks. We apply the GREEDY algorithm and its two variants to queueing networks that we generate from the following six random-graph models (see section SM8 for their definitions):

1. Barabási–Albert (BA) graphs with $n = 100$ nodes and a mean degree of $m_{\mathrm{BA}} = 3$.
2. Erdős–Rényi (ER) $G(n,p)$ graphs with $n = 100$ nodes and a connection probability of $p = 0.06$.
3. Random regular graphs (RRGs) with $n = 100$ nodes and a node degree of $d = 6$.
4. Watts–Strogatz (WS) graphs with $n = 100$ nodes, $2k_{\mathrm{WS}} = 6$ neighbors for each node of the initial ring network, and a rewiring probability of $p = 0.5$.
5. Random geometric graphs (RGGs) on a unit square with $n = 100$ nodes and a connection radius of $r = 0.19$.
6. Chung–Lu (CL) graphs with $n = 100$ nodes and a degree distribution that we determine from the degree sequence of a BA network with $n = 100$ nodes and $m_{\mathrm{BA}} = 3$.

Each of the models generates a set $\{G_{\mathrm{original}}\}$ of undirected networks. The parameters of the random-graph models ensure that each network G_{original} has the same number n of nodes ($n = 100$) and the same expected number $\mathbb{E}[m]$ of undirected edges ($\mathbb{E}[m] = 300$). We generate 100 networks from each random-graph model and perform the following steps for each network G_{original} to convert it to a queueing network G that satisfies our reachability conditions. To ensure that G is connected, we remove all nodes (and their associated incident edges) from the original network G_{original} that do not belong to the largest connected component. Therefore, the number of nodes of a network G may be smaller than 100 and the mean number of edges across all G typically is smaller than 300. We choose a source node s uniformly at random from the nodes of G. We then also choose a sink node uniformly at random from all sink-node candidates. A node k is a *sink-node candidate* if $k \neq s$ and removing k and its incident edges does not disconnect the subgraph of $G = (V,E)$ that is induced on $V \setminus \{k\}$. For $n \geq 2$, there is at least one sink-node candidate[8] for any choice of s. We must choose

[8]To see this, consider a node k whose shortest path to node s is of maximal length (i.e., no nodes are farther than node k from s). Node k is a sink-node candidate because a shortest path from any other node j to s does not go through k. (Otherwise, such a shortest path from j to s is longer than a shortest path from k to s.) Therefore, the subgraph of $G = (V,E)$ that is induced on $V \setminus \{k\}$ is connected, as required.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

a sink node from the sink-node candidates to ensure that our reachability conditions are satisfied in the queueing network. We then relabel the nodes so that node 1 is the source node and node n is the sink node (where n is the number of nodes of G).

We set the homogeneous service rate to $\mu = 3\lambda_{\max}$, where $\lambda_{\max}$ is the maximum arrival rate in the network, and we perform our edge-perturbation algorithms on each graph G. For each graph G, we set the number T of edge perturbations to $\lfloor 0.48\, m \rceil$, where m is the number of edges of G and $\lfloor \cdot \rceil$ denotes rounding to the nearest integer. We select K_k so that we recalculate the ranking of the edge perturbations after making $k\lfloor F_{\text{iter}} m \rceil$ total edge perturbations, where $F_{\text{iter}} = 0.02$ denotes the fraction of a network's edges that we change between successive iterations of an algorithm. In section SM9, we present our results when minimizing $\lambda_{\max}$ and minimizing λ_{total}. As we explained in section 3.1, minimizing λ_{total} is equivalent to minimizing Q in the large-μ regime. A network that minimizes Q in the small-μ regime also minimizes $\lambda_{\max}$, so minimizing $\lambda_{\max}$ is a proxy[9] for minimizing Q in the small-μ regime. The results that we present in section 4.4 are consistent with our results when using the GREEDY algorithm to minimize λ_{total}. This suggests that a service rate of $\mu = 3\lambda_{\max}$ is in the large-μ regime for the networks that we consider. When we minimize $\lambda_{\max}$ (see section SM9.2), we obtain qualitatively similar results to the ones that we present in section 4.4.

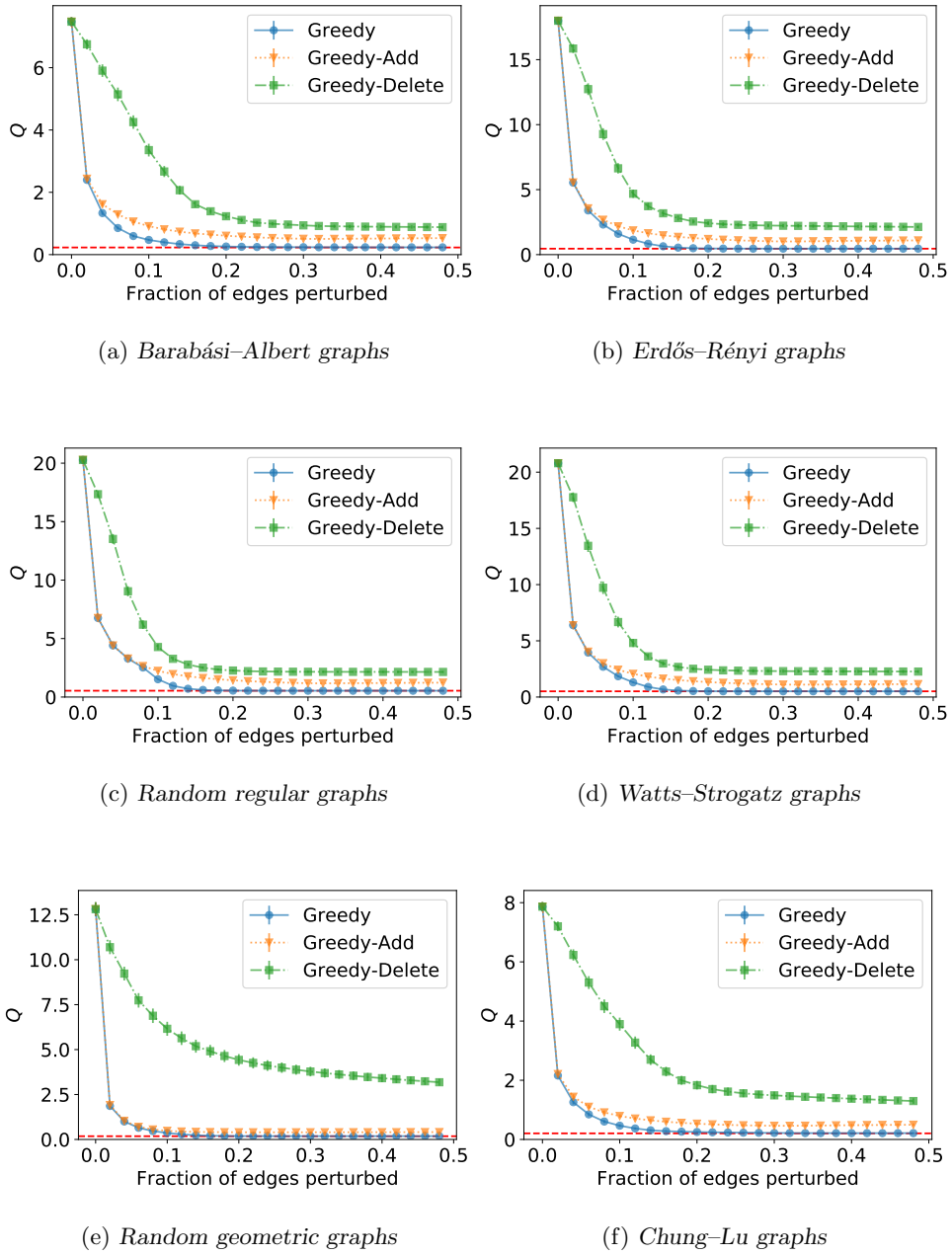
4.4. Results of applying our greedy algorithms to random networks. In our numerical computations, we find that edge additions tend to decrease Q more effectively than edge deletions, but we decrease Q the most by allowing both types of edge perturbations. We are then able to achieve values of Q that are close to the conjectured minimum value. Additionally, the vast majority of added edges that achieve the largest reductions in Q are incident to the sink node.

In Figure 6, we show the mean value of Q (and the associated standard errors) as a function of the fraction F of edges that we perturb for each of the three algorithms. Specifically, F is the number of edges that we perturb divided by m, which is the number of edges of the original network. The original GREEDY algorithm and its two variants (see Table 1) are able to reduce Q using edge perturbations for all six of our random-graph models (see Figure 6). Unsurprisingly, we achieve the largest reduction in Q using the original GREEDY algorithm, which can either add edges or delete them. The GREEDY-ADD algorithm reduces Q slightly less than the GREEDY algorithm. The GREEDY-DELETE algorithm yields the smallest reduction in Q.

In each simulation, we start with a connected graph and perturb it using one of our greedy algorithms. For a given μ, we estimate a lower bound of Q using the total mean queue size Q_{opt} of $G_{\mathcal{U}_n}^{\lambda_{\text{total}}}$ (see Figure 5c), which is our conjectured optimal undirected network for large values of μ. The value $\mu = 3\lambda_{\max}$, where $\lambda_{\max}$ is the maximum arrival rate of the initial graph of each simulation, is typically much larger (by a factor of at least 6.5–42, depending on the random-graph model) than the arrival rates of the optimized networks. Additionally, the results in the present subsection are qualitatively similar to our results when we use the greedy algorithms to reduce λ_{total} (see section SM9.1). Therefore, we are likely in the large-μ regime. To see how close the values of Q that we obtain using our algorithms are to Q_{opt}, we calculate the ratio $R_Q = Q/Q_{\text{opt}}$ for each simulation. For each random-graph model, we report the mean value of R_Q for the GREEDY algorithm and its two variants (see Table 2).

[9]It is an imperfect proxy because minimizing $\lambda_{\max}$ does not guarantee that we can find a network that minimizes Q in the small-μ regime.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.



(a) *Barabási–Albert graphs* (b) *Erdős–Rényi graphs*

(c) *Random regular graphs* (d) *Watts–Strogatz graphs*

(e) *Random geometric graphs* (f) *Chung–Lu graphs*

FIG. 6. *Comparison of the performance of our* GREEDY *algorithm and its two variants (*GREEDY-ADD *and* GREEDY-DELETE*) for decreasing the total mean queue size Q with edge perturbations. We plot the mean and standard error of Q as a function of the fraction F of perturbed edges (i.e., the number of edge perturbations divided by the number of edges of the original network). In most of the curves, the standard error is smaller than the marker size. For each $F \in [0, 0.48]$, we take the mean of all simulations that yield finite Q after perturbing a fraction F of the edges. The red dashed line indicates the mean value of Q_{opt} (which we average over* 100 *networks), which is our conjectured lower bound of Q.*

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

TABLE 2

The mean value of $R_Q = Q/Q_{\text{opt}}$ for our original GREEDY *algorithm and its two variants (*GREEDY-ADD *and* GREEDY-DELETE*) for our six random-graph models. We generate* 100 *graphs for each random-graph model and run the three algorithms on each graph.*

Model	GREEDY	GREEDY-ADD	GREEDY-DELETE
BA	1.009	2.241	5.641
ER	1.007	2.224	5.041
RRG	1.000	2.169	4.004
WS	1.000	2.184	4.569
RGG	1.034	2.148	61.580
Chung–Lu	1.030	2.252	7.692

TABLE 3

The fraction F_{sink} of edge additions (of the first $\lfloor 0.2\,m \rceil$ edge perturbations) that are incident to the sink node for the GREEDY *and* GREEDY-ADD *algorithms for our six random-graph models. We generate* 100 *graphs for each random-graph model and run the two algorithms on each graph. We show the minimum, mean, and maximum values of F_{sink} of the* 100 *simulations for each random-graph model.*

	GREEDY			GREEDY-ADD		
Model	Min	Mean	Max	Min	Mean	Max
BA	1.000	1.000	1.000	0.948	0.998	1.000
ER	1.000	1.000	1.000	1.000	1.000	1.000
RRG	1.000	1.000	1.000	1.000	1.000	1.000
WS	1.000	1.000	1.000	1.000	1.000	1.000
RGG	1.000	1.000	1.000	0.754	0.955	1.000
Chung–Lu	1.000	1.000	1.000	0.929	0.994	1.000

The original GREEDY algorithm achieves mean values of R_Q that are close to 1, which suggests that the algorithm is able to perturb the networks to obtain networks whose total mean queue sizes are close to (or even equal to) Q_{opt}. The mean values of R_Q for GREEDY-ADD range between about 2 and about 4. GREEDY-DELETE yields much larger mean values of R_Q; they are between 4.1 (for RRGs) and 47.5 (for RGGs).

In our simulations, we observe that the vast majority of edge additions yield edges that are incident to the sink. To quantitatively measure this observation, in each simulation, we record all edge additions until the fraction F of perturbed edges reaches 0.2. (We use the value 0.2 because we obtain the largest reduction in Q when $F \in [0, 0.2]$ (see Figure 6).) We then calculate the fraction F_{sink} of edge additions (of the first $\lfloor 0.2\,m \rceil$ edge perturbations) that yield edges that are incident to the sink. For all six random-graph models, we find that the mean value of F_{sink} is close to or equal to 1 for both the GREEDY and GREEDY-ADD algorithms. In other words, almost all edge additions (of the first $\lfloor 0.2\,m \rceil$ edge perturbations) yield edges that are incident to the sink node for both of these algorithms (see Table 3).

4.5. Results of applying the GREEDY algorithm to a supermarket store network. We now apply the GREEDY algorithm and its two variants to a supermarket store network (see Figure 7) with $n = 179$ nodes and $m = 384$ edges. We construct the store network by manually dividing the floor area into rectangular zones (i.e., nodes) and connecting contiguous zones by edges [20, 22]. Deleting an edge (i, j) corresponds to blocking the direct walkway between zones i and j, such as by adding an extra

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

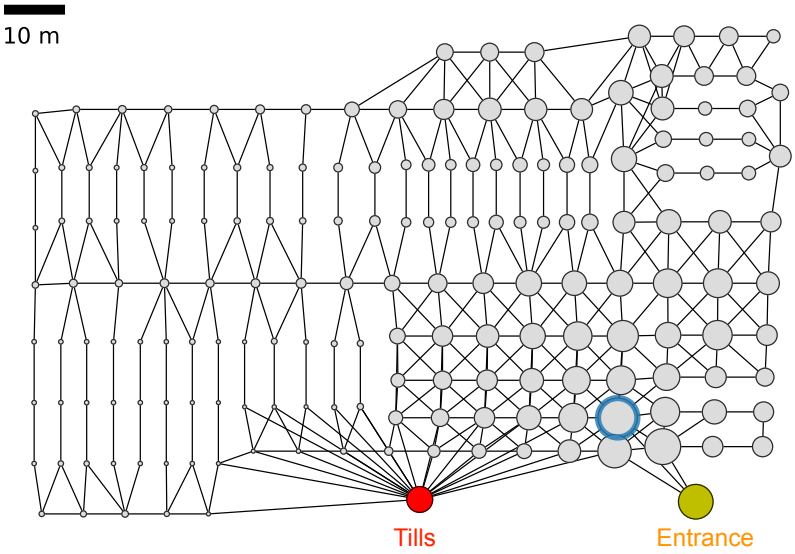


FIG. 7. *A supermarket store network with $n = 179$ nodes and $m = 384$ edges. We color the source node (i.e., the entrance) in greenish yellow and the sink node (i.e., the tills) in red. The size of each node is proportional to its arrival rate in the store's associated queueing network. We circle the node with the highest arrival rate in blue.*

shelf. Adding an edge (i, j) corresponds to creating a direct walkway between zones i and j. This is possible[10] only if zones i and j are next to each other but are separated by shelves. Automatically identifying which edges are possible to add is a difficult task and may not be possible without manual oversight. In this subsection, we use the following procedure to identify possible edge additions. We represent each edge (i, j) as a line between the centroids of zones i and j. We allow an edge addition $\Delta(i, j)$ if the edge does not intersect any other edges. We call this the *planarity constraint* because it ensures that new edges do not violate the planarity of the store network.[11] In practice, it may not be possible to add every edge that we identify in this way to a store network. One should check manually whether or not any perturbed store network is realizable in practice.

We apply our three greedy algorithms, with the planarity constraint whenever we allow edge additions, to a store network G from a large United Kingdom supermarket chain. In our simulation, we take $T = \lfloor 0.18\, m \rceil = 69$ and determine K_k as described in section 4.3. The service rate is $3\lambda_{\max} \approx 6.68$, where $\lambda_{\max}$ is the maximum arrival rate of G. With these parameters, the original value of Q is 11.1.

First, we apply the GREEDY-DELETE algorithm to the store network G. This algorithm decreases Q significantly and yields a final value of $Q \approx 1.02$ (see Figure 8a). In the final network (see the left part of Figure 8a), the random walkers in the network are effectively "directed" towards the sink node (i.e., the tills) because we have deleted edges that lead them further away from the sink.

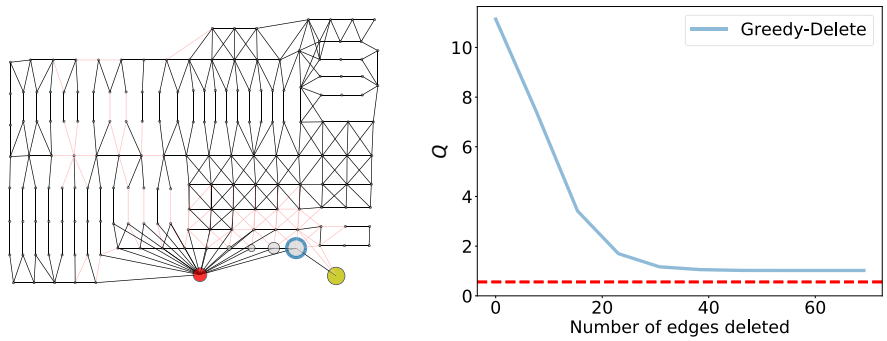
Second, we apply the GREEDY-ADD algorithm (with the planarity constraint) to G. The algorithm decreases Q to a final value of $Q \approx 6.30$. The minimum value of

[10] We assume that we cannot build bridges over the shelves in a store.

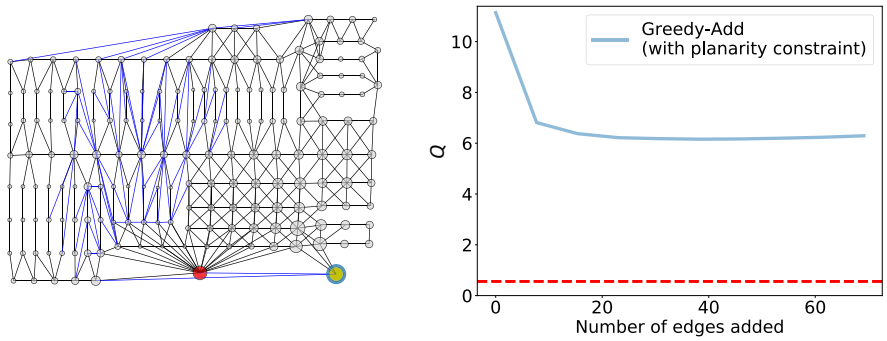
[11] Although the original store network is not planar, it is approximately planar because we only need to delete a small number of edges to obtain a planar graph.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

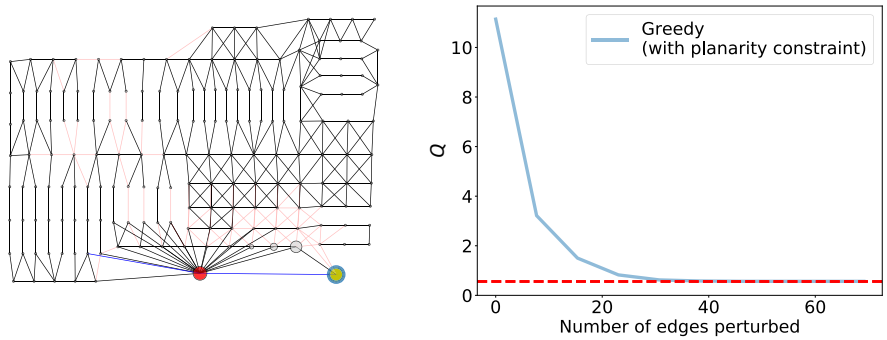
Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



(a) *Edge deletion*



(b) *Edge addition*



(c) *Edge deletion and addition*

FIG. 8. *Application of the* GREEDY *algorithm and its two variants (*GREEDY-ADD *and* GREEDY-DELETE*) to a store network. The left part of each panel shows the final network after we apply the indicated greedy algorithm. We color the source node (i.e., the entrance) in greenish yellow and the sink node (i.e., the tills) in red. We color deleted edges in pink and added edges in blue. The size of each node is proportional to its arrival rate; we circle the node with the highest arrival rate in blue. The right part of each panel shows the total mean queue size Q as a function of the number of perturbed edges. The red dashed line indicates our conjectured theoretical minimum value of Q.*

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Q, which we obtain after 38 edge additions, is $Q \approx 6.16$. In contrast to our results for our six random-graph models (see section 4.4), the decrease in Q is much smaller for the GREEDY-ADD algorithm than for the GREEDY-DELETE algorithm. This is the case primarily because we are using the more restrictive GREEDY-ADD algorithm with the planarity constraint for the store network. This limits the possible edges that the algorithm can add. Without the planarity constraint, we obtain a (much smaller) final value of $Q \approx 1.25$. However, this value is still larger than the total mean queue size Q that we obtain using the GREEDY-DELETE algorithm.

Finally, we apply the GREEDY algorithm (with the planarity constraint) to G. This algorithm yields the largest decrease in Q and results in a final value of $Q \approx 0.565$ (see Figure 8b). This value of Q is close to the total mean queue size $Q_{\text{opt}} \approx 0.559$ of the conjectured optimal network $G_{\mathcal{U}_n}^{\lambda_{\text{total}}}$ for large μ. When we apply the GREEDY algorithm to the store network, most edge perturbations are edge deletions; only two perturbations are edge additions. The perturbed network directs random walkers to the sink node and thereby reduces congestion in the store. Unfortunately, the resulting store layout is not very useful in practice. In a supermarket store network, customers will not buy much if they immediately proceed to the tills and leave. This reveals a shortcoming of our model. In our optimizations, we perturb networks in a way that directs walkers as quickly as possible to a store exit. This strategy ignores the shopping intentions of customers. In section 5, we briefly discuss a variety of possible ways to improve our model to address this shortcoming.

5. Conclusions and discussion. Inspired by the modeling of customer mobility and congestion in supermarkets, we studied an unbiased random walk on a single-source, single-sink queueing network. We used the total mean queue size Q as our congestion measure because (by Little's law) minimizing Q is equivalent to minimizing the mean journey time of the random walkers.

We examined which network topologies minimize Q for a homogeneous service rate μ. One of our main results is that the network $G_{\bar{\mathcal{C}}_n}$ (see Figure 2) minimizes Q (for any value of μ) of the graphs in $\bar{\mathcal{C}}_n$, which is the set of all networks with n nodes (with $n \geq 3$) that satisfy certain reachability conditions and do not have a directed edge from the source node to the sink node. We also explored what occurs (1) when we allow that directed edge and (2) when we impose an undirected network structure. We found numerical evidence that $G_{\bar{\mathcal{C}}_n}$ minimizes Q for all values of μ in the first case and that different networks minimize Q for different values of μ in the second case. We also established relationships (1) between minimizing Q and minimizing the maximum arrival rate $\lambda_{\max}$ and (2) between minimizing Q and minimizing the total arrival rate λ_{total}. For sufficiently small μ (i.e., in the small-μ regime), minimizing Q implies that one is also minimizing $\lambda_{\max}$. For sufficiently large μ (i.e., in the large-μ regime), minimizing Q is equivalent to minimizing λ_{total}. Therefore, when minimizing Q in one of these two regimes, one can instead minimize the quantities $\lambda_{\max}$ or λ_{total}, which are easier to calculate and are independent of μ.

We also examined the use of edge perturbations—in the form of edge additions, edge deletions, or both—of an existing network to decrease the total mean queue size Q. We introduced a greedy algorithm that, at each step, performs the edge perturbation that decreases Q the most. Because supermarket store networks are undirected, we only considered undirected networks, but it is straightforward to adapt our greedy algorithm to directed networks. We applied the greedy algorithm and two variants of it—one that allows only edge additions and another that allows only edge

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

deletions—to networks from six random-graph models and to a store network from a real supermarket. We found that the greedy algorithm and its two variants are able to reduce Q on all six types of random graphs. The greedy algorithm with both edge additions and edge deletions yielded the largest reduction of Q and achieved a value that is close to our conjectured minimum value of Q. The variant with only edge addition performed slightly worse, although it was much better than the variant with only edge deletion. In our numerical experiments on random graphs, most edge additions yielded edges that are incident to the sink node. For the supermarket store network, to ensure that our perturbations yield realistic (or at least plausible) network structures, the only edges that are permissible to add are ones that satisfy a planarity constraint (by not intersecting existing edges). We found that the greedy algorithm with edge additions, edge deletions, and the planarity constraint reduced Q to a value that is close to our conjectured minimum. However, the resulting network is not practical for supermarkets because it directs customers towards the tills (and thus towards a store's exit without shopping).

One potential way to address some of the shortcomings of our model is to incorporate shopping intentions. For example, each walker can have a shopping list of nodes to visit before leaving a store [21]. Another possible improvement is to incorporate constraints, such as a minimum visitation probability for each node, to ensure certain elements of realism in our models and thereby yield more practical supermarket store layouts through optimization.

Our work gives insight into how network topology affects the total mean queue size Q in supermarket stores. We examined this question by studying unbiased random walks on open queueing networks with a single source and a single sink. We explored optimal network topologies and three greedy algorithms for perturbing an existing graph towards an optimal network topology. The queueing networks that we studied have a single source, a single sink, and homogeneous service rates. This is a special situation, and extending our analysis to queueing networks with more realistic scenarios, such as by examining more realistic mobility models (e.g., by supposing that each walker has a shopping list or by considering congestion-biased random walks in which walkers tend to avoid overly congested nodes) and heterogeneities (e.g., in service rates or mobility) are important avenues to explore in future work.

Appendix A. Proof of Theorem 3.1. To prove Theorem 3.1, we first prove two lemmas. These lemmas specify inequalities that are satisfied by the arrival rates λ_i of the nodes of any open queueing network $G \in \bar{\mathcal{C}}_n$.

LEMMA A.1. *For any open queueing network $G \in \bar{\mathcal{C}}_n$, the arrival rates λ_i of its nodes satisfy the following relations:*

$$\lambda_1 \geq 1\,, \tag{A.1}$$

$$\lambda_i > 0\,, \quad i \in \{2, \ldots, n-1\}\,, \tag{A.2}$$

$$\lambda_n = 1\,. \tag{A.3}$$

Proof. The inequality (A.1) follows from equation (2.5) and the nonnegativity of $\boldsymbol{P}$ and λ_i. We verified the inequality (A.2) and equation (A.3) in section 2.1 (see equation (2.7)). ∎

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

LEMMA A.2. *The arrival rates λ_i of the nodes of any network $G \in \bar{\mathcal{C}}_n$ satisfy*

$$\sum_{i=2}^{n-1} \lambda_i \geq 1\,. \tag{A.4}$$

Proof. Our network does not include the edge $(1,n)$, so walkers can visit only nodes $2,\dots,n-1$ (and not node n) from node 1 in a single step. Therefore,

$$\sum_{i=2}^{n-1} P_{1i} = \sum_{i=2}^{n} P_{1i} = 1\,. \tag{A.5}$$

The arrival rate λ_1 of node 1 is equal to the rate of departure at stationarity. Each walker that departs from node 1 goes to one of the interior nodes $2,\dots,n-1$, so the sum of the arrival rates of all interior nodes $2,\dots,n-1$ must be at least λ_1. Finally, $\lambda_1 \geq 1$ because of (A.1); this gives the desired result. ∎

The arrival rates $\bar{\lambda}_i^{(\mathrm{opt})}$ of the nodes of $G_{\bar{\mathcal{C}}_n}$ are

$$\bar{\lambda}_i^{(\mathrm{opt})} = \begin{cases} 1\,, & i = 1 \text{ or } i = n \\ 1/(n-2)\,, & i \in \{2,\dots,n-1\}\,, \end{cases} \tag{A.6}$$

so the total mean queue size is

$$Q(G_{\bar{\mathcal{C}}_n}, \mu) = \bar{Q}_{\mathrm{opt}} = \frac{2}{\mu - 1} + \frac{1}{\mu - 1/(n-2)}\,. \tag{A.7}$$

We will show that $\bar{\boldsymbol{\lambda}}^{(\mathrm{opt})} = (\bar{\lambda}_1^{(\mathrm{opt})},\dots,\bar{\lambda}_n^{(\mathrm{opt})})$ minimizes Q over the space of all possible arrival rates λ_i for graphs in $\bar{\mathcal{C}}_n$. To do this, we prove a more general statement.

PROPOSITION A.3. *Fix $\mu > 1$ and let $C \in [0,(n-2)\mu)$ be a constant. Let $\Omega_{C,\mu}$ be the set of vectors $\boldsymbol{\lambda} = (\lambda_1,\dots,\lambda_n) \in \mathbb{R}^n$ that satisfy*

$$1 \leq \lambda_1 < \mu\,, \tag{A.8}$$

$$0 < \lambda_i < \mu\,, \quad i \in \{2,\dots,n-1\}\,, \tag{A.9}$$

$$\sum_{i=2}^{n-1} \lambda_i \geq C\,, \tag{A.10}$$

$$\lambda_n = 1\,. \tag{A.11}$$

Let $f(\boldsymbol{\lambda})\colon \Omega_{C,\mu} \to \mathbb{R}$, and suppose that we have a nondecreasing function $g\colon [1,\mu) \to \mathbb{R}$, a nondecreasing, convex, and differentiable function $h\colon (0,\mu) \to \mathbb{R}$, and a constant $c \in \mathbb{R}$ such that

$$f(\boldsymbol{\lambda}) = g(\lambda_1) + \sum_{i=2}^{n-1} h(\lambda_i) + c\,. \tag{A.12}$$

It then follows that f is minimized on $\Omega_{C,\mu}$ when

$$\lambda_i = \begin{cases} 1\,, & i \in \{1,n\} \\ C/(n-2)\,, & i \in \{2,\dots,n-1\}\,. \end{cases} \tag{A.13}$$

That is, for any $\boldsymbol{\lambda} \in \Omega_{C,\mu}$, we have that

$$f(\boldsymbol{\lambda}) \geq g(1) + (n-2)h\left(\frac{C}{n-2}\right) + c\,. \tag{A.14}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Proof. Without loss of generality, we assume that $c = 0$, as adding a constant to f does not change its minimizers. Because g is nondecreasing,

$$g(\lambda_1) \geq g(1) \tag{A.15}$$

for all $\lambda_1 \geq 1$ by the inequality (A.8).

By the convexity of h, for any $\lambda, a \in (0, \mu)$, we have

$$h(\lambda) \geq h(a) + h'(a)(\lambda - a)\,. \tag{A.16}$$

Using $a = \lambda_2^{(\mathrm{opt})} = C/(n-2) \in (0, \mu)$ and $\lambda = \lambda_i$ in the inequality (A.16) and summing over all $i \in \{2, \ldots, n-1\}$ yields

$$\begin{aligned} \sum_{i=2}^{n-1} h(\lambda_i) &\geq (n-2)h\left(\frac{C}{n-2}\right) + h'\left(\frac{C}{n-2}\right)\left(\sum_{i=2}^{n-1}\lambda_i - C\right) \\ &\geq (n-2)h\left(\frac{C}{n-2}\right) \end{aligned} \tag{A.17}$$

because $h'(C/(n-2)) \geq 0$ (recall that h is nondecreasing) and $\sum_{i=2}^{n-1} \lambda_i \geq C$ (by the inequality (A.10)).

Combining the inequalities (A.15) and (A.17) yields

$$f(\boldsymbol{\lambda}) = g(\lambda_1) + \sum_{i=2}^{n-1} h(\lambda_i) \geq g(1) + (n-2)h\left(\frac{C}{n-2}\right), \quad \boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n) \in \Omega_{C,\mu}\,, \tag{A.18}$$

as required. ∎

We are now ready to prove Theorem 3.1.

Proof of Theorem 3.1. Fix $\mu > 1$. To ensure that Q is bounded, we consider a network $G \in \bar{\mathcal{C}}_n$ with arrival rates λ_i that satisfy $\lambda_i < \mu$. By Lemmas A.1 and A.2, the arrival rates λ_i of the nodes of G satisfy $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n) \in \Omega_{1,\mu}$ (i.e., $\Omega_{C,\mu}$ with $C = 1$).

The total mean queue size Q of G as a function of the arrival rates λ_i is

$$Q(G, \mu) = f(\boldsymbol{\lambda}) + \frac{1}{\mu - 1}\,, \tag{A.19}$$

where $f(\boldsymbol{\lambda})$ is given by equation (A.12) with $g(\lambda) = h(\lambda) = \lambda/(\mu - \lambda)$. Note that g is nondecreasing, convex, and differentiable on $(0, \mu)$.

By Proposition A.3,

$$\begin{aligned} f(\boldsymbol{\lambda}) &\geq g(1) + (n-2)h\left(C/(n-2)\right) \\ &= \frac{1}{\mu - 1} + \frac{1}{\mu - 1/(n-2)}\,. \end{aligned} \tag{A.20}$$

Therefore, the total mean queue size Q of any graph $G \in \bar{\mathcal{C}}_n$ satisfies

$$\begin{aligned} Q(G, \mu) &\geq \frac{1}{\mu - 1} + \frac{1}{\mu - 1/(n-2)} + \frac{1}{\mu - 1} \\ &= \bar{Q}_{\mathrm{opt}}\,, \end{aligned} \tag{A.21}$$

so $G_{\bar{\mathcal{C}}_n}$ is Q-optimal over $\bar{\mathcal{C}}_n$. ∎

REFERENCES

[1] A. Arenas, A. Díaz-Guilera, and R. Guimerà, *Communication in networks with hierarchical branching*, Phys. Rev. Lett., 86 (2001), pp. 3196–3199.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

[2] M. Barthelemy, *Spatial Networks: A Complete Introduction: From Graph Theory and Statistical Physics to Real-World Applications*, Springer International Publishing, Cham, Switzerland, 2022.
[3] S. Chen, W. Huang, C. Cattani, and G. Altieri, *Traffic dynamics on complex networks: A survey*, Math. Probl. Eng., 2012 (2011), 732698.
[4] V.-A. Darvariu, S. Hailes, and M. Musolesi, *Planning spatial networks with Monte Carlo tree search*, Proc. R. Soc. Math. Phys. Eng. Sci., 479 (2023).
[5] F. Dörfler, M. Chertkov, and F. Bullo, *Synchronization in complex oscillator networks and smart grids*, Proc. Natl. Acad. Sci. USA, 110, (2013), pp. 2005–2010.
[6] R. Guimerà, A. Díaz-Guilera, F. Vega-Redondo, A. Cabrales, and A. Arenas, *Optimal network topologies for local search with congestion*, Phys. Rev. Lett., 89 (2002), 248701.
[7] D. Helbing, *A fluid dynamic model for the movement of pedestrians*, Complex Syst., 6 (1992), pp. 391–415.
[8] S. P. Hoogendoorn and W. Daamen, *Pedestrian behavior at bottlenecks*, Transport. Sci., 39 (2005), pp. 147–159.
[9] W. Huang and T. W. Chow, *Effective strategy of adding nodes and links for maximizing the traffic capacity of scale-free network*, Chaos, 20 (2010), 033123.
[10] F. Kelly and E. Yudovina, *Stochastic Networks*, Cambridge University Press, Cambridge, UK, 2014.
[11] J. D. C. Little, *A proof for the queuing formula:* $L = \lambda W$, Oper. Res., 9 (1961), pp. 383–387.
[12] G. G. Løvås, *Modeling and simulation of pedestrian traffic flow*, Transp. Res. Part B Methodol., 28 (1994), pp. 429–443.
[13] N. Masuda, M. A. Porter, and R. Lambiotte, *Random walks and diffusion on networks*, Phys. Rep., 716–717 (2017), pp. 1–58.
[14] A. E. Motter, *Cascade control and defense in complex networks*, Phys. Rev. Lett., 93 (2004), 098701.
[15] A. E. Motter, S. A. Myers, M. Anghel, and T. Nishikawa, *Spontaneous synchrony in power-grid networks*, Nat. Phys., 9 (2013), pp. 191–197.
[16] R. Pastor-Satorras, A. Vázquez, and A. Vespignani, *Dynamical and correlation properties of the Internet*, Phys. Rev. Lett., 87 (2001), 258701.
[17] R. Pastor-Satorras and A. Vespignani, *Evolution and Structure of the Internet: A Statistical Physics Approach*, Cambridge University Press, Cambridge, UK, 2007.
[18] M. A. Porter and J. P. Gleeson, *Dynamical Systems on Networks: A Tutorial*, Frontiers in Applied Dynamical Systems: Reviews and Tutorials, volume 4, Springer International Publishing, Cham, Switzerland, 2016.
[19] A. Solé-Ribalta, S. Gómez, and A. Arenas, *A model to identify urban traffic congestion hotspots in complex networks*, Roy. Soc. Open Sci., 3 (2016), 160098.
[20] F. Ying, *Customer Mobility and Congestion in Supermarkets*, D.Phil. thesis, University of Oxford, 2019.
[21] F. Ying and N. O'Clery, *Modelling COVID-19 transmission in supermarkets using an agent-based model*, PLoS ONE, 16 (2021), e0249821.
[22] F. Ying, A. O. G. Wallis, M. Beguerisse-Díaz, M. A. Porter, and S. D. Howison, *Customer mobility and congestion in supermarkets*, Phys. Rev. E, 100 (2019), 062304.
[23] X. Zhan, S. V. Ukkusuri, and P. S. C. Rao, *Dynamics of functional failures and recovery in complex road networks*, Phys. Rev. E, 96 (2017), 052301.

Downloaded 09/15/23 to 131.179.220.28 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

SUPPLEMENTARY MATERIALS: MINIMIZING CONGESTION IN SINGLE-SOURCE, SINGLE-SINK QUEUEING NETWORKS*

FABIAN YING†, ALISDAIR O. G. WALLIS‡, MASON A. PORTER§,
SAM D. HOWISON†, AND MARIANO BEGUERISSE-DÍAZ†

SM1. Optimality of the graph $G_{\bar{\mathcal{C}}_n}$ for other objective functions. Equation Proposition A.3 implies that the graph $G_{\bar{\mathcal{C}}_n}$ is a minimizer for any objective function $f(\boldsymbol{\lambda})$, with $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^n$, of the form

$$f(\boldsymbol{\lambda}) = g(\lambda_1) + \sum_{i=2}^{n-1} h(\lambda_i) + c\,, \tag{SM1.1}$$

where g is a non-decreasing function on $[1, \mu)$, the function h is non-decreasing, convex, and differentiable on $(0, \mu)$, and c is a constant. Recall that $\lambda_n = 1$, so the last term in (SM1.1) is $c\lambda_n = c$.

Examples of objective functions that satisfy Equation (SM1.1) include the total arrival rate λ_{total} (i.e., $g(\lambda) = h(\lambda) = \lambda$ and $c = 1$), the total arrival rates $\sum_{i=2}^{n-1} \lambda_i$ of the interior nodes (i.e., $g(\lambda) = 0$, $h(\lambda) = \lambda$, and $c = 0$), and the total mean queue size Q when node 1 has a different service rate μ_1 than the other nodes (i.e., $g(\lambda) = \lambda/(\mu_1 - \lambda)$, $h(\lambda) = \lambda/(\mu - \lambda)$, and $c = 1/(\mu - 1)$).

The graph $G_{\bar{\mathcal{C}}_n}$ minimizes both $\lambda_{\max}$ and $\max_{i=2}^{n-1} \lambda_i$ over $\bar{\mathcal{C}}_n$. We proved the former in section 3.1. To see the latter, note that the arrival rates λ_i of any graph $G \in \bar{\mathcal{C}}_n$ satisfy $\sum_{i=2}^{n-1} \lambda_i \geq 1$ by Lemma A.2. Because $\lambda_i \geq 0$, it follows that $\max_{i=2}^{n-1} \lambda_i \geq 1/(n-2)$. The arrival rates of $G_{\bar{\mathcal{C}}_n}$ achieve this lower bound. We denote the vector of these arrival rates by $\bar{\boldsymbol{\lambda}}^{(\text{opt})}$.

SM2. Proof of optimality of the graph $G_{\bar{\mathcal{C}}_n}$ for other types of queues. When we use other types of queues, we can also show that, under certain conditions, the arrival rates $\bar{\boldsymbol{\lambda}}^{(\text{opt})}$ of $G_{\bar{\mathcal{C}}_n}$ minimize the total mean queue size Q over $\bar{\mathcal{C}}_n$. We no longer have a single service rate μ_i for each node i; instead, there is an infinite-dimensional vector of service rates $\{\mu_{ik}\}_{k=1,2,\ldots}$ for each node i, where μ_{ik} is the service rate of node i when there are k customers at node i. As in our consideration of single-server queues, we assume that each node has the same service rates, so

$$\mu_{ik} = \mu_{1k} \tag{SM2.1}$$

for all nodes i, all positive integers k, and some constants $\mu_{1k} > 0$.

Our definition of Q-optimality for homogeneous single-server queues in the main text (see section 3) entails a homogeneous service rate μ, so we need to extend our definition of Q-optimality to more general queues. To do this, we start with the quantity

*Supplementary material for SIAP MS#M145751.
https://doi.org/10.1137/21M1457515

†Mathematical Institute, University of Oxford, Oxford OX2 6GG, UK (fabian.m.ying@gmail.com, howison@maths.ox.ac.uk, marianob@spotify.com).

‡Tesco PLC, Tesco House, Shire Park, Kestrel Way, Welwyn Garden City, AL7 1GA, UK (Alisdair.Wallis@tesco.com).

§Department of Mathematics, University of California, Los Angeles, CA 90095 USA and Santa Fe Institute, Santa Fe, NM 87501 USA (mason@math.ucla.edu).

$$U = \sup\left\{ x \geq 0 \colon \sum_{k=1}^{\infty} \frac{x^k}{\prod_{l=1}^{k} \mu_{1l}} < \infty \right\}, \tag{SM2.2}$$

which depends only on the service rates μ_{1k}. A stationary state exists for a queueing network that satisfies our reachability conditions (see Equation (2.1)) if the arrival rates λ_i satisfy $\lambda_i < U$ for all i [SM6]. Therefore, given service rates μ_{1k}, the quantity U is the supremum of the arrival rates λ_i to ensure that a stationary state exists. One can verify that $U = \mu$ for single-server queues with service rate μ. We assume that the service rates are sufficiently large to ensure that $U > 1$, as otherwise there exists no stationary state for any network $G \in \bar{\mathcal{C}}_n$. We thereby write the following extended definition for Q-optimality: A network $G \in \mathcal{G}$ is *Q-optimal* over a set $\mathcal{G}$ of graphs if, for any service rates μ_{1k} such that $U > 1$, the total mean queue size Q of G does not exceed the total mean queue size for any graph $G' \in \bar{\mathcal{C}}_n$. As before, we say that a queueing network $G \in \bar{\mathcal{C}}_n$ without a stationary state has total mean queue size Q of infinity.

Provided the service rates are sufficiently large (to ensure that there exists a stationary state), the arrival rates λ_i of each node i in a queueing network depend only on the network topology. In particular, they are independent of the type of queue. Therefore, using different types of queues does not change the arrival rates, so the arrival rates still satisfy Lemmas A.1 and A.2. Furthermore, for any network $G \in \bar{\mathcal{C}}_n$ with finite Q, we have $\lambda_i < U$ (as otherwise no stationary state exists, which would then imply that $Q = \infty$). Consequently, the arrival rates λ_i of any queueing network $G \in \bar{\mathcal{C}}_n$ satisfy $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n) \in \Omega_{1,U}$ where $\lambda_n = 1$ and $\Omega_{1,U}$ is defined in Equations (A.8)–(A.10).

We can extend Proposition A.3 to apply to any type of queue that satisfies Equation (SM2.1) by replacing μ with U everywhere. Therefore, it follows that $G_{\bar{\mathcal{C}}_n}$ is Q-optimal over $\bar{\mathcal{C}}_n$ for any such queue. For example, the proof of Theorem 3.1 extends to queueing networks in which each node is a two-server queue with equal service rates (of $\mu/2$) at each queue. In this case, $U = \mu$, which is the same as in a single-server queue.

We choose $g(\lambda) = h(\lambda) = 2\mu\lambda/(\mu^2 - \lambda^2)$, so the total mean queue size is

$$Q = \sum_{i=1}^{n} \frac{2\mu\lambda_i}{\mu^2 - \lambda_i^2}\,. \tag{SM2.3}$$

Therefore, $G_{\bar{\mathcal{C}}_n}$ is a network in $\bar{\mathcal{C}}_n$ that minimizes Q when all queues are two-server queues.

SM3. Q-optimality over $G_{\mathcal{C}_n}$. When we consider networks in $\mathcal{C}_n$ (instead of $\bar{\mathcal{C}}_n$), the bound on $\sum_{i=2}^{n-1} \lambda_i$ from inequality (A.4) in Lemma A.2 does not hold. Therefore, to prove Q-optimality of $G_{\mathcal{C}_n}$ over $\mathcal{C}_n$ using the same approach that we used for $\bar{\mathcal{C}}_n$, we need to prove a different bound on $\sum_{i=2}^{n-1} \lambda_i$. We wrote our conjectured bound in (3.5).

If (3.5) is true, we can show that $G_{\mathcal{C}_n}$ is Q-optimal by following the same steps as in the proof of Theorem 3.1, except that we use $C = 1 - 1/(n-1)$ instead of $C = 1$. The arrival rates $\lambda_i^{(\mathrm{opt})}$ of $G_{\mathcal{C}_n}$ are

$$\lambda_i^{(\mathrm{opt})} = \begin{cases} 1\,, & i = 1 \text{ or } i = n \\ \frac{1}{n-1} = \frac{C}{n-2}\,, & i \in \{2, \ldots, n-1\}\,, \end{cases} \tag{SM3.1}$$

where $C = 1 - 1/(n-1)$. The arrival rates satisfy $\sum_i \lambda_i^{(\mathrm{opt})} = C$, as in the case for $\bar{\mathcal{C}}_n$.

SM4. Simulated-annealing algorithm for determining optimal network topologies.

SM4.1. Description of the simulated-annealing (SA) algorithm. We use a simulated-annealing (SA) algorithm to try to find (directed or undirected) networks with a minimum value of an objective function. We use objective functions $f(\boldsymbol{\lambda})$ (specifically, $\sum_{i=2}^{n-1} \lambda_i$, $\lambda_{\max}$, and λ_{total}) that depend only on the arrival rates λ_i (with $i \in \{1, \ldots, n\}$) of the nodes of an unweighted network G. We describe the SA algorithm in Algorithm SM4.1.

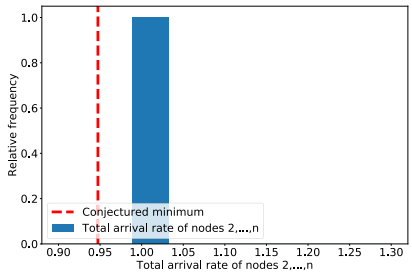
Algorithm SM4.1 Simulated-annealing (SA) algorithm for finding networks that minimize the objective function f

procedure SIMULATEDANNEALING
Input: Number n of nodes, objective function $f(\boldsymbol{\lambda})$, number L of iterations
Output: A network G that has been optimized according to f
Initialize:
    Let G_0 (the initial network) be the fully connected network with n nodes
    Set the computational temperature to $T_c = 1$
Algorithm:
    for iteration $k \in \{1, \ldots, L\}$ **do**
        Step A: Choose an ordered node pair (i, j) in the graph $G_{k-1} = (V, E)$ uniformly at random from all possible pairs of nodes
        Step B: Perform one of the following actions (depending on whether $(i, j) \in E$ or $(i, j) \notin E$):
        if $(i, j) \notin E$ **then**
            $G_k \leftarrow G_{k-1}$ with (i, j) added
        else
            $G_k \leftarrow G_{k-1}$ with (i, j) removed
            if G_k does not satisfy the reachability conditions **then**
                Repeat Steps A and B until one selects a node pair (i, j) for which either $(i, j) \in E$ or G_k satisfies the reachability conditions
        Step C: Compute the change Δf in the objective function f between the networks G_k and G_{k-1}
        if $\Delta f \geq 0$ (i.e., G_k has a larger or equal objective-function value) **then**
            With probability $1 - \exp(-\Delta f / T_c)$, reject the change and set
            $G_k \leftarrow G_{k-1}$
        Step E: Reduce the computational temperature T_c by $8.3 \times 10^{-6}\, T_c$
    Assign $G \leftarrow G_L$
    return G

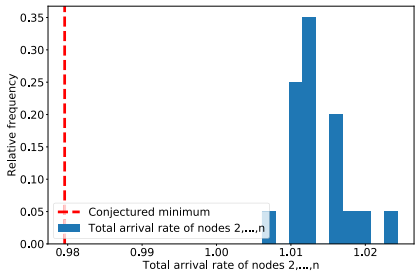
We run 10^5 iterations in total. As the SA algorithm progresses, we decrease the computational temperature and the algorithm accepts progressively fewer changes that increase the objective function.

We use the SA algorithm to attempt to find the following optimal n-node networks:

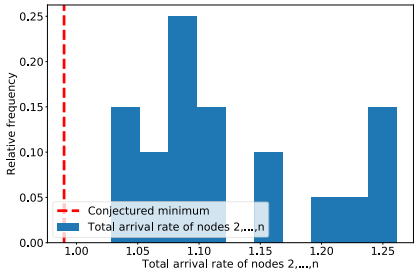
(1) a directed network with minimal $\sum_{i=2}^{n-1} \lambda_i$;
(2) an undirected network with minimal $\lambda_{\max} = \max_{i=1}^{n} \lambda_i$; and
(3) an undirected network with minimal $\lambda_{\text{total}} = \sum_{i=1}^{n} \lambda_i$.



(a) $n = 20$ *nodes*



(b) $n = 50$ *nodes*



(c) $n = 100$ *nodes*

FIG. SM1. *Histograms of the minimum values of* $\sum_{i=2}^{n-1} \lambda_i$ *that we find using an SA algorithm. We indicate our conjectured minimum value using the dashed red line. Note that the horizontal scales are different in the different figure panels.*

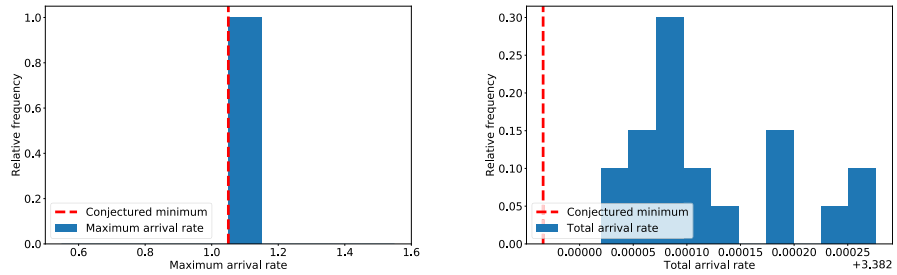
SM4.2. Results.

SM4.2.1. Directed n-node networks with minimal $\sum_{i=2}^{n-1} \lambda_i$. We conjecture that for any directed network $G \in \mathcal{C}_n$ with $n \geq 3$, the arrival rates λ_i of the nodes of G satisfy $\sum_{i=2}^{n-1} \lambda_i \geq 1 - 1/(n-1)$ (see (3.5)). We have verified this conjecture for all $n \in \{3, \ldots, 7\}$ by exhaustive enumeration.

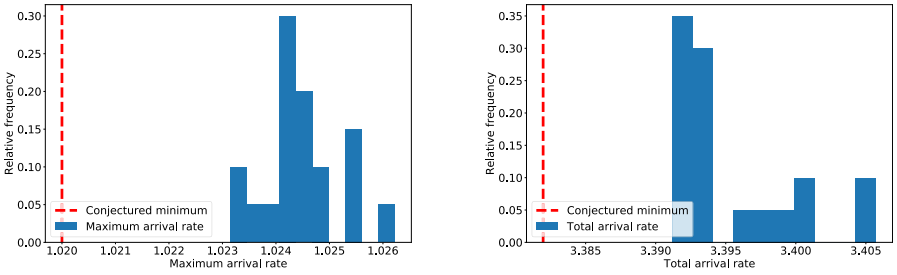
We use the SA algorithm to find larger networks with minimal $\sum_{i=2}^{n-1} \lambda_i$. For each n, we run the SA algorithm 20 times and record the value of $\sum_{i=2}^{n-1} \lambda_i$ of the optimized network that we obtain in each run. For $n = 20$, $n = 50$, and $n = 100$ nodes, the SA algorithm did not find any networks with $\sum_{i=2}^{n-1} \lambda_i$ that are smaller than $1 - 1/(n-1)$, which is our conjectured minimum (see Figure SM1). The values of $\sum_{i=2}^{n-1} \lambda_i$ from our computations are close to our conjectured minimum value. Therefore, our computational results support our conjecture.

SM4.2.2. Undirected n-node networks with minimal $\lambda_{\max}$ or λ_{total}. We conjecture for any network with $n \geq 3$ nodes that the network $G_{\mathcal{U}_n}^{\lambda_{\text{total}}}$ (see Figure 5c) uniquely minimizes λ_{total} over $\mathcal{U}_n$. If this conjecture holds, then for any $n \geq 5$, there are no Q-optimal networks over the space $\mathcal{U}_n$ of undirected networks that satisfy our reachability conditions (see section 3). We also conjecture that for any network with $n \geq 3$ nodes, the graph $G_{\mathcal{U}_n}^{\lambda_{\max}}$ (see Figure 5d) minimizes $\lambda_{\max}$.

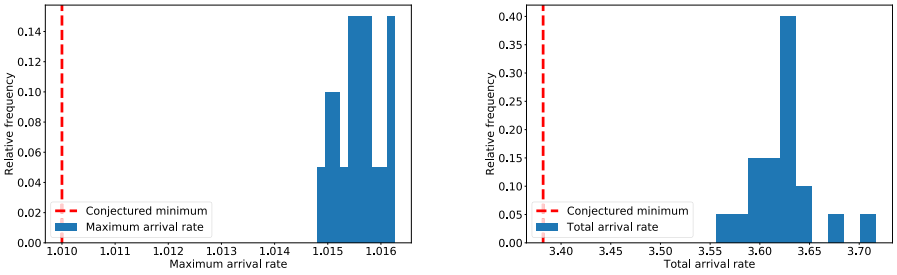
For all $n \in \{3, \ldots, 7\}$, we have verified both conjectures by exhaustive enumeration. For larger networks, we use the SA algorithm that we described in Appendix SM4.1 to obtain undirected networks with small values of $\lambda_{\max}$ or λ_{total}.



(a) $n = 20$ *nodes* (b) $n = 20$ *nodes*



(c) $n = 50$ (d) $n = 50$



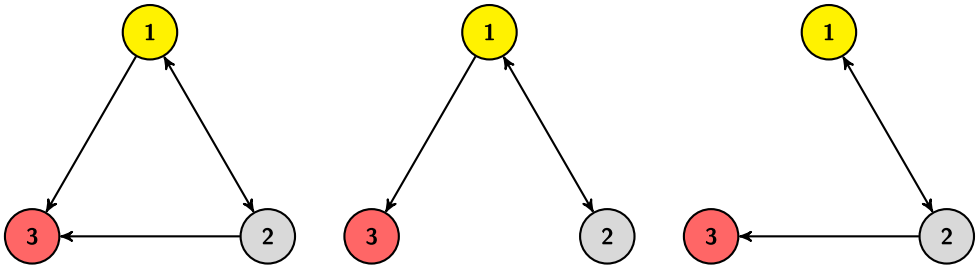
(e) $n = 100$ *nodes* (f) $n = 100$

FIG. SM2. *Histograms of the minimum values of (left panels)* $\lambda_{\max}$ *and (right panels)* λ_{total} *that we obtain using an SA algorithm. In each case, we indicate our conjectured minimum value using a dashed red line. Note that the horizontal scales are different in different panels.*

For each n, we run the SA algorithm 20 times and record the minimum objective-function value from each run. For both objective functions, the SA algorithm yields networks with objective-function values that are close to (but above) the conjectured minimum values for $n = 20$, $n = 50$, and $n = 100$ (see Figure SM2). Therefore, our results support both conjectures.

SM5. Q-optimal networks over $\mathcal{U}_n$ for $n = 3$ and $n = 4$. We show that $G_{\mathcal{U}_n}^{\lambda_{\max}}$ is the unique Q-optimal network (which we defined in section 3) over $\mathcal{U}_n$ for $n = 3$ and $n = 4$. Note that $G_{\mathcal{U}_n}^{\lambda_{\max}}$ is identical to $G_{\mathcal{U}_n}^{\lambda_{\text{total}}}$ for $n = 3$ and $n = 4$.

There are only 3 different networks in $\mathcal{U}_3$ (see Figure SM3). Of these 3 networks, the graph $G_{\mathcal{U}_3}^{\lambda_{\max}}$ has the smallest arrival rates λ_i for each node i. We show the arrival rates in the captions of Figures SM3a to SM3c. Because $Q = \sum_i \lambda_i/(\mu - \lambda_i)$ is an increasing function of λ_i for fixed values of the other arrival rates, the network $G_{\mathcal{U}_3}^{\lambda_{\max}}$ minimizes Q for all values of μ, so it is Q-optimal over $\mathcal{U}_3$.



(a) $G_{\mathcal{U}_3}^{\lambda_{\max}}$ *with arrival rates* $(\frac{4}{3}, \frac{2}{3}, 1)$ (b) *The 3-node network with arrival rates* $(2, 1, 1)$ (c) *The 3-node network with arrival rates* $(2, 2, 1)$

FIG. SM3. *The* 3 *networks in* $\mathcal{U}_3$ *and their associated arrival rates* $(\lambda_1, \lambda_2, \lambda_3)$.

There are 28 different networks in $\mathcal{U}_4$. Of these networks, there are 11 pairs (G_1, G_2) of distinct networks (i.e., $G_1 \neq G_2$) such that G_1 is the same as G_2 except that the labels of interior nodes (i.e., the nodes that are neither a source nor a sink) are swapped. For each pair (G_1, G_2), the network G_2 has the same arrival rates as G_1, except that the arrival rates of interior nodes (i.e., nodes 2 and 3) are swapped. Therefore, the total mean queue sizes of G_1 and G_2 are identical for any value of μ. Consequently, we only need to consider one network from each of these pairs, so there are 17 different networks to examine. We show these 17 networks in Figure SM4 and give their arrival rates in Table SM1. For each node i, the arrival rate λ_i is smaller in $G_{\mathcal{U}_4}^{\lambda_{\max}}$ than in all other networks except for networks 2 and 6. (In networks 2 and 6, node 3 has a smaller arrival rate than in $G_{\mathcal{U}_4}^{\lambda_{\max}}$.) Using the same argument as with $n = 3$, we see that the network $G_{\mathcal{U}_4}^{\lambda_{\max}}$ has smaller values of Q than the other networks, except for networks 2 and 6, for all values of μ.

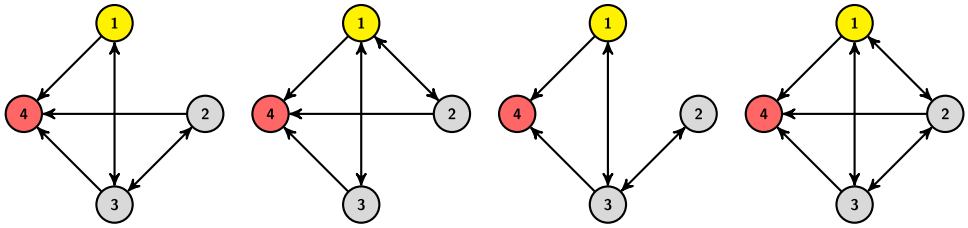
We now show that $G_{\mathcal{U}_4}^{\lambda_{\max}}$ also has smaller or equal values of Q than networks 2 and 6 for all values of μ. Let $\lambda_i^{(\mathrm{opt})}$, $\lambda_i^{(3)}$, and $\lambda_i^{(6)}$ be the arrival rates of node i in the network $G_{\mathcal{U}_4}^{\lambda_{\max}}$, network 3, and network 6, respectively. They take the respective values

$$\begin{aligned} \left(\lambda_1^{(\mathrm{opt})}, \lambda_2^{(\mathrm{opt})}, \lambda_3^{(\mathrm{opt})}, \lambda_4^{(\mathrm{opt})}\right) &= (1.25, 0.25, 0.75, 1)\,, \\ \left(\lambda_1^{(3)}, \lambda_2^{(3)}, \lambda_3^{(3)}, \lambda_4^{(3)}\right) &= (1.5, 0.5, 0.5, 1)\,, \\ \left(\lambda_1^{(6)}, \lambda_2^{(6)}, \lambda_3^{(6)}, \lambda_4^{(6)}\right) &= \left(2, \frac{2}{3}, \frac{2}{3}, 1\right)\,. \end{aligned} \tag{SM5.1}$$

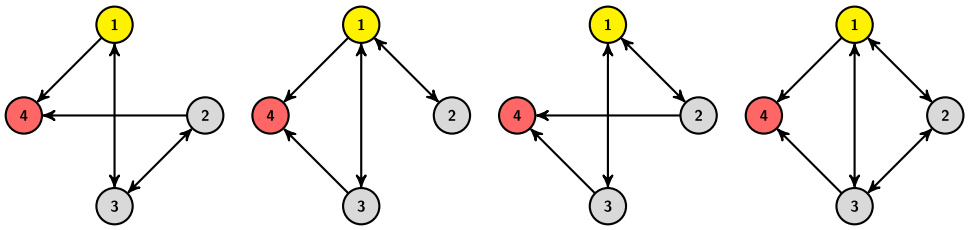
For a given service rate μ, let $f(x) = x/(\mu - x)$. The function f gives the mean queue size of a node with arrival rate $x \in [0, \mu)$. To show that $G_{\mathcal{U}_4}^{\lambda_{\max}}$ has smaller values of Q than networks 2 and 6, we need to show that

$$f(\lambda_1^{(\mathrm{opt})}) + f(\lambda_2^{(\mathrm{opt})}) + f(\lambda_3^{(\mathrm{opt})}) + f(\lambda_4^{(\mathrm{opt})}) \leq f(\lambda_1^{(l)}) + f(\lambda_2^{(l)}) + f(\lambda_3^{(l)}) + f(\lambda_4^{(l)})\,, \tag{SM5.2}$$

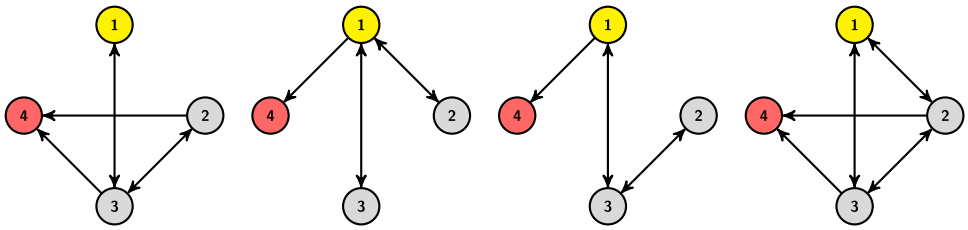
for $l = 3$ and $l = 6$ and all sufficiently large values of μ (to ensure that a stationary state exists). Specifically, we need to verify Equation (SM5.2) for all values of μ such that $\mu > \lambda_i^l$ for $l \in \{3, 6\}$ and for all i. (When $l = 3$, we require that $\mu > 1.5$; when $l = 6$, we require that $\mu > 2$.) Because f is increasing and $\lambda_2^{(\mathrm{opt})} \leq \lambda_2^{(l)}$ and $\lambda_4^{(\mathrm{opt})} = \lambda_4^{(l)}$ for $l = 3$ and $l = 6$, it suffices to show that



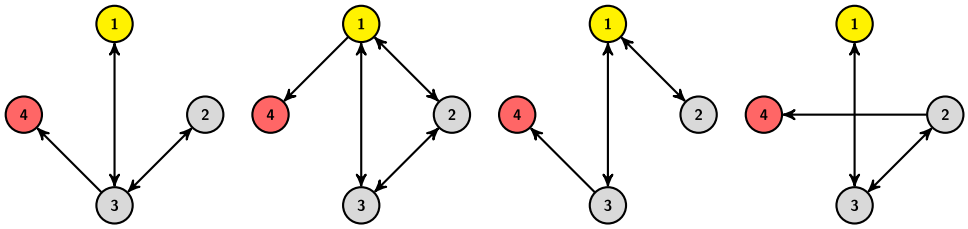
(a) $G_{\mathcal{U}_4}^{\lambda_{\max}}$ (b) Network 2 (c) Network 3 (d) Network 4



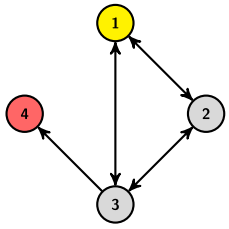
(e) Network 5 (f) Network 6 (g) Network 7 (h) Network 8



(i) Network 9 (j) Network 10 (k) Network 11 (l) Network 12



(m) Network 13 (n) Network 14 (o) Network 15 (p) Network 16



(q) Network 17

FIG. SM4. *The* 17 *networks in* $\mathcal{U}_4$ *with different sets of arrival rates (see Table* SM1*).*

TABLE SM1

Arrival rates (to 2 decimal places) of the 17 networks in $\mathcal{U}_4$ with different sets of arrival rates. (There are 28 networks in total in $\mathcal{U}_4$. As we explain in the text, each of the 11 networks that we do not list in this table has the same arrival rates (up to relabeling of the interior nodes) as one of these 17 networks.)

	λ_1	λ_2	λ_3	λ_4
$G_{\mathcal{U}_4}^{\lambda_{\max}}$	1.25	0.25	0.75	1.00
Network 2	1.50	0.50	0.50	1.00
Network 3	1.33	0.33	1.00	1.00
Network 4	1.50	0.75	0.75	1.00
Network 5	1.50	0.50	1.00	1.00
Network 6	2.00	0.67	0.67	1.00
Network 7	2.00	1.00	1.00	1.00
Network 8	1.88	1.00	1.12	1.00
Network 9	1.67	0.67	2.00	1.00
Network 10	3.00	1.00	1.00	1.00
Network 11	2.00	1.00	2.00	1.00
Network 12	2.00	1.50	1.50	1.00
Network 13	2.00	1.00	3.00	1.00
Network 14	3.00	2.00	2.00	1.00
Network 15	4.00	2.00	2.00	1.00
Network 16	3.00	2.00	4.00	1.00
Network 17	3.33	2.67	3.00	1.00

$$f(\lambda_1^{(\mathrm{opt})}) + f(\lambda_3^{(\mathrm{opt})}) \leq f(\lambda_1^{(l)}) + f(\lambda_3^{(l)})\,, \tag{SM5.3}$$

for $l = 3$ and $l = 6$ and all values of μ such that $\mu > \lambda_i^l$ for $l \in \{3, 6\}$ and all i.

We first show Equation (SM5.3) for $l = 3$. The function f is convex (i.e., concave down) on $[0, \mu)$, so it satisfies

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y) \tag{SM5.4}$$

for all $t \in [0, 1]$ and $x, y \in [0, \mu)$. Setting $t = 0.25$ and $t = 0.75$ in Equation (SM5.4) yields

$$\begin{aligned} f(0.25x + 0.75y) &\leq 0.25f(x) + 0.75f(y)\,, \\ f(0.75x + 0.25y) &\leq 0.75f(x) + 0.25f(y)\,. \end{aligned} \tag{SM5.5}$$

We sum both sides of the inequalities in (SM5.5) to obtain

$$f(0.75x + 0.25y) + f(0.75x + 0.25y) \leq f(x) + f(y)\,. \tag{SM5.6}$$

Using $x = \lambda_1^{(3)} = 1.5$ and $y = \lambda_3^{(3)} = 0.5$ then gives

$$f(0.25 \times 1.5 + 0.75 \times 0.5) + f(0.75 \times 1.5 + 0.25 \times 0.5) \leq f(1.5) + f(0.5)\,. \tag{SM5.7}$$

That is,

$$f(1.25) + f(0.75) \leq f(1.5) + f(0.5)\,, \tag{SM5.8}$$

which verifies Equation (SM5.3) with $(\lambda_1^{(\mathrm{opt})}, \lambda_3^{(\mathrm{opt})}) = (1.25, 0.75)$ and $(\lambda_1^{(3)}, \lambda_3^{(3)}) = (1.5, 0.5)$, as required.

For $l = 6$, we sum the inequalities in (SM5.4) and insert $t = 0.875$ and $t = 0.125$ to obtain

$$f(0.875x + 0.125y) + f(0.125x + 0.875y) \le f(x) + f(y)\,. \tag{SM5.9}$$

Substituting $x = 4/3$ and $y = \lambda_3^{(6)} = 2/3$ into Equation (SM5.9) yields

$$f(1.25) + f(0.75) \le f(4/3) + f(2/3)\,. \tag{SM5.10}$$

Therefore,

$$\begin{aligned} f(\lambda_1^{(\mathrm{opt})} + f(\lambda_3^{(\mathrm{opt})} &\le f(4/3) + f(\lambda_3^{(6)}) \\ &\le f(\lambda_1^{(6)}) + f(\lambda_3^{(6)})\,, \end{aligned} \tag{SM5.11}$$

where the last inequality holds because f is increasing and $\lambda_1^{(6)} = 2 \ge 4/3$. Consequently, $G_{\mathcal{U}_4}^{\lambda_{\max}}$ has smaller values of Q than networks 2–17, so it is Q-optimal over $\mathcal{U}_4$.

SM6. Maximum arrival rates of $G_{\mathcal{U}_n}^{\lambda_{\max}}$ and $G_{\mathcal{U}_n}^{\lambda_{\mathrm{total}}}$. We show that $G_{\mathcal{U}_n}^{\lambda_{\mathrm{total}}}$ (see Figure 5c) does not minimize the maximum arrival rate $\lambda_{\max}$ (see Figure 5d) over $\mathcal{U}_n$ for $n \ge 5$. We do this by showing that the maximum arrival rate of $G_{\mathcal{U}_n}^{\lambda_{\mathrm{total}}}$ is larger than the maximum arrival rate of $G_{\mathcal{U}_n}^{\lambda_{\max}}$.

We first examine the arrival rates λ_i of the nodes of $G_{\mathcal{U}_n}^{\lambda_{\max}}$. These arrival rates satisfy the following traffic equations (2.5):

$$\begin{aligned} \lambda_1 &= \frac{1}{n-1}\lambda_2 + 1\,, \\ \lambda_2 &= \frac{1}{2}\lambda_1 + \sum_{i=3}^{n-1}\frac{1}{2}\lambda_i\,, \\ \lambda_i &= \frac{1}{n-1}\lambda_2\,, \quad i \in \{3,\dots,n-1\}\,, \\ \lambda_n &= \frac{1}{2}\lambda_1 + \frac{1}{n-1}\lambda_2 + \sum_{i=3}^{n-1}\frac{1}{2}\lambda_i\,. \end{aligned} \tag{SM6.1}$$

The arrival rates of $G_{\mathcal{U}_n}^{\lambda_{\max}}$ for $n \ge 5$ are

$$\begin{aligned} \lambda_1 &= 1 + \frac{1}{n}\,, \\ \lambda_2 &= 1 - \frac{1}{n}\,, \\ \lambda_i &= \frac{1}{n}\,, \quad i \in \{3,\dots,n-1\}\,, \\ \lambda_n &= 1\,, \end{aligned} \tag{SM6.2}$$

which satisfy Equation (SM6.1). Therefore, we see that $G_{\mathcal{U}_n}^{\lambda_{\max}}$ has a maximum arrival rate of $\lambda_{\max} = \lambda_1 = 1 + 1/n$.

We now examine the arrival rates λ_i of $G_{\mathcal{U}_n}^{\lambda_{\text{total}}}$. They satisfy the following traffic equations (2.5):

$$\begin{aligned}
\lambda_1 &= \frac{1}{3}\lambda_2 + 1\,, \\
\lambda_2 &= \frac{1}{2}\lambda_1 + \frac{1}{3}\lambda_3\,, \\
\lambda_i &= \frac{1}{3}\lambda_{i-1} + \frac{1}{3}\lambda_{i+1}\,, \quad i \in \{3, \dots, n-3\}\,, \\
\lambda_{n-2} &= \frac{1}{3}\lambda_{n-3} + \frac{1}{2}\lambda_{n-1}\,, \\
\lambda_{n-1} &= \frac{1}{3}\lambda_{n-2}\,, \\
\lambda_n &= \frac{1}{2}\lambda_1 + \sum_{i=2}^{n-2}\frac{1}{3}\lambda_i + \frac{1}{2}\lambda_{n-1}\,.
\end{aligned} \tag{SM6.3}$$

We establish a lower bound for λ_1 (and thus a lower bound for $\lambda_{\max}$) by substituting the second equation of Equation (SM6.3) into the first equation of Equation (SM6.3) to obtain

$$\lambda_1 = \frac{1}{3}\left(\frac{1}{2}\lambda_1 + \frac{1}{3}\lambda_3\right) + 1\,. \tag{SM6.4}$$

Rearranging Equation (SM6.4) yields

$$\begin{aligned}
\frac{5}{6}\lambda_1 &= \frac{1}{9}\lambda_3 + 1 \\
&> 1\,,
\end{aligned} \tag{SM6.5}$$

because $\lambda_3 > 0$ (as node 3 is reachable by a walker). Consequently, $\lambda_1 > 6/5 = 1+1/5$, so $\lambda_{\max} > 1 + 1/5 \geq 1 + 1/n$ for all $n \geq 5$. This shows that $G_{\mathcal{U}_n}^{\lambda_{\text{total}}}$ has a smaller maximum arrival rate than $G_{\mathcal{U}_n}^{\lambda_{\max}}$ for $n \geq 5$ and cannot minimize $\lambda_{\max}$ over $\mathcal{U}_n$ for $n \geq 5$.

SM7. Variants of our greedy algorithm for reducing Q. In Algorithms SM7.1 and SM7.2, we summarize the two variants of our GREEDY algorithm. The first variant only allows edge additions, and the second variant only allows edge deletions. Our original algorithm allows both edge additions and edge deletions.

SM8. Random-graph models. In this section, we present the definitions of the six random-graph models that we used in section 4.

SM8.1. Barabási–Albert graphs. A *Barabási–Albert* (BA) graph [SM1] is an undirected graph that we construct as follows. Given the parameters $n \in \mathbb{Z}_+$ and $m_{\text{BA}} \in \mathbb{Z}_+$ and an initial undirected graph with $m_0 \geq m_{\text{BA}}$ nodes, we add one node at a time to the graph until it has n nodes. Each new node j attaches to m_{BA} distinct existing nodes, where we choose each node i with a probability that is proportional to its degree d_i. We set the initial graph to be the graph with $m_0 = m_{\text{BA}}$ nodes and no edges. The number of edges of each BA graph is

$$m = (n - m_{\text{BA}})m_{\text{BA}}\,. \tag{SM8.1}$$

Algorithm SM7.1 Greedy algorithm for reducing Q using only edge additions.

procedure GREEDY-ADD
Input: Network $G = (V, E)$, service rates μ_i, total number T of edge additions, number K_k of edge additions between recalculating rankings for each iteration k
Output: Network G with up to T edge additions
Initialize:
    Calculate ΔQ for each edge addition Δe
    Rank all edge additions in increasing order of ΔQ and save them in the list R
    num_edges_added $= 0$
    iteration_number $= 1$
Algorithm:
    while `True` **do**
        num_edges_added_in_loop $= 0$
        Set $K = K_k$, where $k =$ *iteration_number*
        while *num_edges_added_in_loop* $< K$ **do**
            Take the first edge addition $\Delta e = \Delta(i, j)$ in R and remove it from R
            $G \leftarrow G$ with e added
            Increment *num_edges_added_in_loop* by 1
            Increment *num_edges_added* by 1
            if *num_edges_added* $\geq T$ or R is empty **then**
                return G
            if R is empty **then** ▷ No further edges can be added
                return G
        Recalculate ΔQ for each edge addition Δe
        Rank all edge additions in increasing order of ΔQ and save them in the list R
        Increment *iteration_number* by 1

SM8.2. Erdős–Rényi graphs. An *Erdős–Rényi* (ER) $G(n, p)$ graph [SM4] (which is also called a *Bernoulli random graph*) with $p \in [0, 1]$ is an undirected graph with n nodes in which any two nodes i and j are connected by an edge with a homogeneous, independent probability p. The number of edges of a $G(n, p)$ graph follows a binomial distribution with mean

$$\mathbb{E}[m] = \frac{n(n-1)p}{2}\,. \tag{SM8.2}$$

SM8.3. Random regular graphs. Given the parameters $n \in \mathbb{Z}_+$ and $d \in \mathbb{Z}_+$ such that nd is even, a *random regular graph* (RRG) is an n-node d-regular graph that we choose uniformly at random from all n-node d-regular graphs. We use the algorithm by Kim and Vu [SM7] that samples a d-regular graph in an asymptotically uniform way when $d = \mathcal{O}(n^{1/3-\epsilon})$ for any $\epsilon < 1/3$ as $n \to \infty$.

The number of edges of an RRG is

$$m = \frac{nd}{2}\,. \tag{SM8.3}$$

SM8.4. Watts–Strogatz graphs. Given the parameters $n \in \mathbb{Z}_+$, $k_{\mathrm{WS}} \in \mathbb{Z}_+$, and $p \in [0, 1]$, we generate a *Watts–Strogatz* (WS) graph as follows. We start with a graph with n nodes that are arranged as a regular n-gon, and we add undirected edges such that each node is adjacent to its nearest $2k_{\mathrm{WS}}$ nodes. For each node i, we

Algorithm SM7.2 Greedy algorithm for reducing Q using only edge deletions.

procedure GREEDY-DELETE
Input: Network $G = (V, E)$, service rates μ_i, total number T of edge deletions, number K_k of edge deletions between recalculating rankings for each iteration k
Output: Network G with up to T edge deletions
Initialize:
    Calculate ΔQ for each edge deletion Δe
    Rank all edge deletions in increasing order of ΔQ and save them in the list R
    num_edges_deleted $= 0$
    iteration_number $= 1$
Algorithm:
    while `True` **do**
        num_edges_deleted_in_loop $= 0$
        Set $K = K_k$, where $k =$ *iteration_number*
        while *num_edges_deleted_in_loop* $< K$ **do**
            Take first edge deletion $\Delta e = \Delta(i, j)$ in R and remove it from R
            $G' \leftarrow G$ with e removed
            if G' satisfies reachability conditions **then**
                $G \leftarrow G'$
                Increment *num_edges_deleted_in_loop* by 1
                Increment *num_edges_deleted* by 1
            else
                continue ▷ Skip because it is not a valid graph
            if *num_edges_deleted* $\geq T$ or R is empty **then**
                return G
            if R is empty **then** ▷ No further edges can be removed
                return G
        Recalculate ΔQ for each edge deletion Δe
        Rank all edge deletions in increasing order of ΔQ and save them in the list R
        Increment *iteration_number* by 1

then consider the edges that connect i with its k_{WS} rightmost neighbors. We rewire each of the associated k_{WS} edges with independent probability p as follows. For each edge (i, j_{old}) that we rewire, we choose a node j_{new} uniformly at random from all nodes that are not neighbors of i and replace (i, j_{old}) with (i, j_{new}).

In the present paper, we produce connected WS graphs by repeatedly sampling WS graphs using the procedure above until it produces a connected graph. The number of edges of each WS graph is

$$m = nk_{\mathrm{WS}}\,. \tag{SM8.4}$$

SM8.5. Random geometric graphs. Given the parameters $n \in \mathbb{Z}_+$ and $r \in [0, \sqrt{2}]$, we define[1] a *random geometric graph* (RGG) [SM2] to be a spatial, undirected graph in $\mathbb{R}^2$ in which we place n nodes uniformly at random in the unit square in $\mathbb{R}^2$ such that each node is adjacent to all nodes within a Euclidean distance of r.

[1] Our definition is the simplest, traditional version of a RGG. See [SM8] for more general versions of RGGs.

For small r, the mean number of edges of an RGG is

$$\mathbb{E}[m] \approx \frac{n(n-1)\pi r^2}{2}\,. \tag{SM8.5}$$

Because of the boundary, the precise value of $\mathbb{E}[m]$ is smaller than the right-hand side of (SM8.5). The formula (SM8.5) becomes exact as $n \to \infty$ and $r \to 0$ with nr fixed.

SM8.6. Chung–Lu graphs. The *Chung–Lu* model [SM3] is a variant of a configuration model [SM5]. Given $n \in \mathbb{Z}_+$ and a sequence $w_1, \ldots, w_n$ of positive weights (to encode the expected degree sequence of a graph), we generate a Chung–Lu graph as follows. For each pair of distinct nodes, i and j, we place an edge (i, j) between them with independent probability $w_i w_j / \sum_k w_k$. The expected degree $\mathbb{E}[d_i]$ of node i is then

$$\mathbb{E}[d_i] = w_i \left(1 - \frac{w_i}{\sum_k w_k}\right)\,, \tag{SM8.6}$$

which tends to w_i as $n \to \infty$.

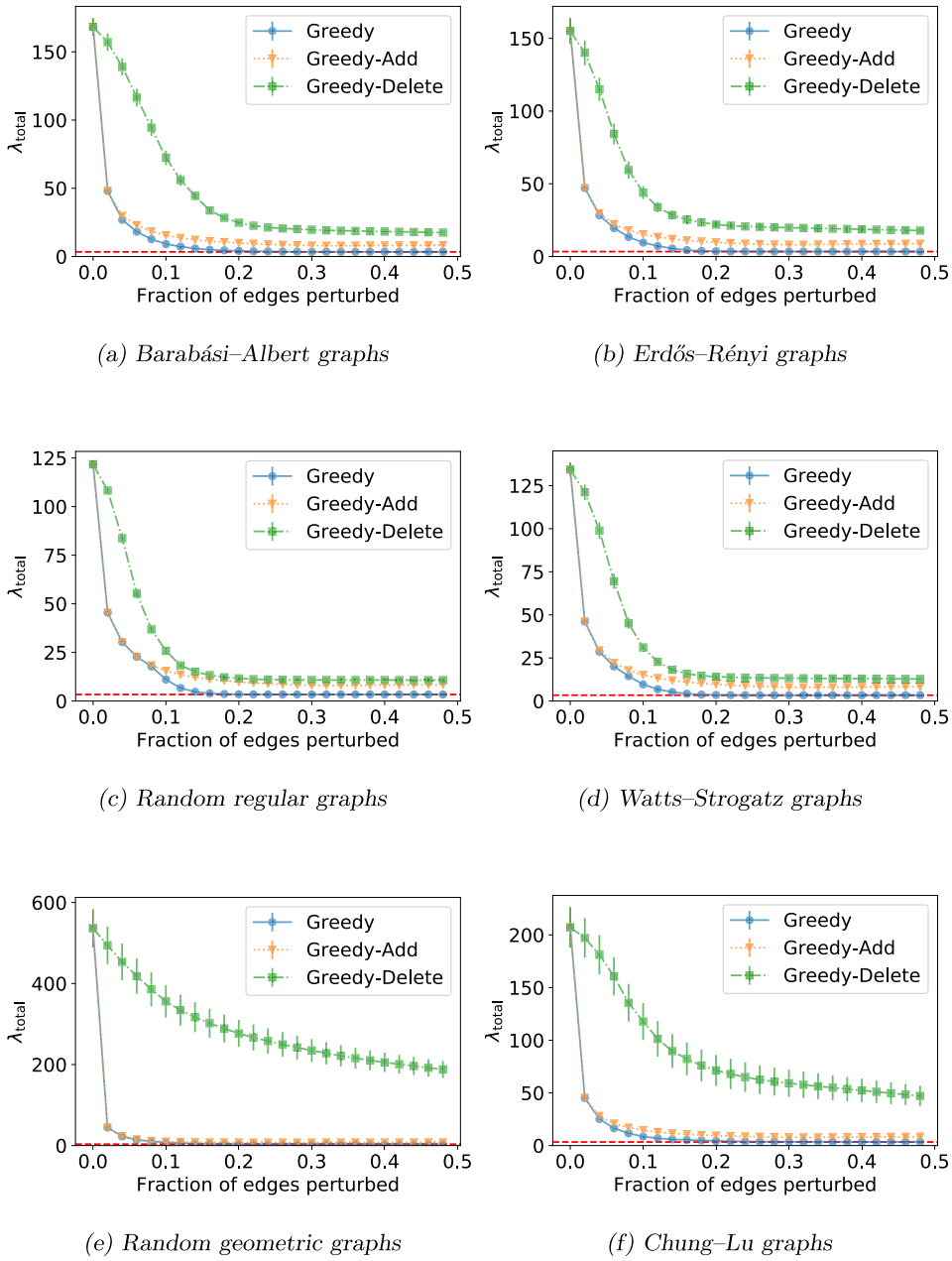
The expected number of edges is

$$\mathbb{E}[m] = \frac{\sum_i \mathbb{E}[d_i]}{2}\,. \tag{SM8.7}$$

SM9. Reducing $\lambda_{\max}$ or λ_{total} by adding or deleting edges. We use the Greedy algorithm and its two variants (Greedy-Add and Greedy-Delete) to reduce the maximum arrival rate $\lambda_{\max}$ and total arrival rate λ_{total}. To do this, we amend Algorithm 4.1 and replace any calculation of ΔQ by the changes in $\lambda_{\max}$ and λ_{total}, respectively. In other words, we change the objective function in our greedy algorithms. We apply the three algorithm variants to 100 networks from each of the six random-graph models that we mentioned in section 4. (See section SM8 for the definitions of these models.) We use the same model parameters as in section 4; we listed them in section 4.3.

SM9.1. Reducing the value of the total arrival rate λ_{total}. In Figure SM5, we plot the mean value of λ_{total} as a function of the fraction of perturbed edges for the three algorithm variants. The curves in Figure SM5 are qualitatively similar to those in Figure 6 (where we used these algorithms to reduce Q). Analogously to our definition of R_Q (see section 4.4), we define $R_{\lambda_{\text{total}}}$ to be the ratio of the achieved minimum value of λ_{total} to the conjectured minimum value. (The conjectured minimum value of λ_{total} is the total arrival rate of $G_{\mathcal{U}_n}^{\lambda_{\text{total}}}$ (see section 3.4).) For each of the random-graph models, we observe that the mean value of $R_{\lambda_{\text{total}}}$ is close to the corresponding mean value of R_Q (see Table SM2 and Table 2). The similarity of the mean values of R_Q and $R_{\lambda_{\text{total}}}$ and of the curves in Figure SM5 and Figure 6 suggest that reducing Q (with $\mu = 3\lambda_{\max}$) is approximately equivalent to reducing λ_{total}. In other words, we are in the large-μ regime with $\mu = 3\lambda_{\max}$ for our random-graph models with the parameter values in section 4.3.

SM9.2. Reducing the value of the maximum arrival rate $\lambda_{\max}$. When we use the Greedy, Greedy-Add, and Greedy-Delete algorithms to reduce $\lambda_{\max}$, the mean minimum achieved values of $\lambda_{\max}$ for all three algorithms are significantly larger than the conjectured minimum value of $\lambda_{\max}$ (see Figure SM6). (The conjectured minimum value of $\lambda_{\max}$ is the maximum arrival rate of $G_{\mathcal{U}_n}^{\lambda_{\max}}$ (see section 3.4).) We calculate the ratio $R_{\lambda_{\max}}$ of the minimum achieved value of $\lambda_{\max}$ to the

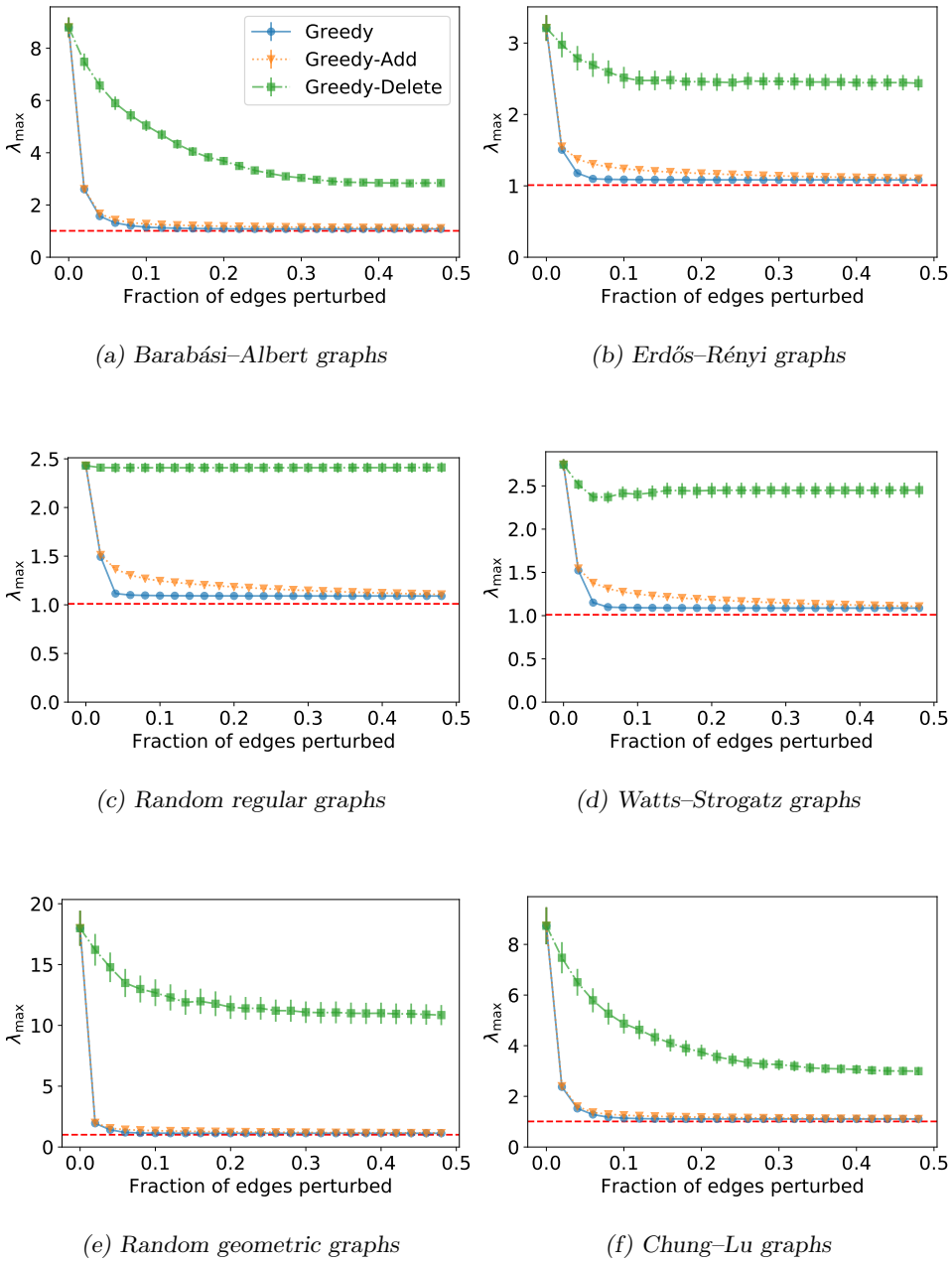


(a) Barabási–Albert graphs *(b) Erdős–Rényi graphs*

(c) Random regular graphs *(d) Watts–Strogatz graphs*

(e) Random geometric graphs *(f) Chung–Lu graphs*

FIG. SM5. *Comparison of the performance of the* GREEDY *algorithm and its two variants (*GREEDY-ADD *and* GREEDY-DELETE*) for decreasing the total arrival rate λ_{total} with edge perturbations. We plot the mean and standard error of λ_{total} as a function of the fraction F of perturbed edges (i.e., the number of edge perturbations divided by the number of edges of the original graph). For each $F \in [0, 0.48]$, we take the mean over all of the simulations that yield finite λ_{total} after perturbing a fraction F of the edges. The red dashed line indicates our conjectured lower bound of λ_{total}. (For most of the curves, the standard error is smaller than the marker size.)*



(a) Barabási–Albert graphs

(b) Erdős–Rényi graphs

(c) Random regular graphs

(d) Watts–Strogatz graphs

(e) Random geometric graphs

(f) Chung–Lu graphs

FIG. SM6. *Comparison of the performance of the* GREEDY *algorithm and its two variants (*GREEDY-ADD *and* GREEDY-DELETE*) for decreasing the maximum arrival rate $\lambda_{\max}$ with edge perturbations. We plot the mean and standard error of $\lambda_{\max}$ as a function of the fraction F of perturbed edges (i.e., the number of edge perturbations divided by the number of edges of the original graph). For each $F \in [0, 0.48]$, we take the mean over all of the simulations that yield finite $\lambda_{\max}$ after perturbing a fraction F of the edges. The red dashed line indicates our conjectured lower bound of $\lambda_{\max}$. (For most of the curves, the standard error is smaller than the marker size.)*

TABLE SM2

The mean value of $R_{\lambda_{\text{total}}}$ for the GREEDY *algorithm and its two variants (*GREEDY-ADD *and* GREEDY-DELETE*) for our six random-graph models. We generate* 100 *networks for each random-graph model and run the three algorithms on those networks.*

Model	GREEDY	GREEDY-ADD	GREEDY-DELETE
BA	1.017	2.329	5.198
ER	1.007	2.367	5.274
RRG	1.000	2.354	3.178
WS	1.000	2.345	3.781
RGG	1.030	2.180	55.444
Chung–Lu	1.034	2.279	13.932

TABLE SM3

The mean value of $R_{\lambda_{\max}}$ for the original GREEDY *algorithm and its two variants (*GREEDY-ADD *and* GREEDY-DELETE*) for our six random-graph models. We generate* 100 *networks for each random-graph model and run the three variants of the algorithms on those networks.*

Model	GREEDY	GREEDY-ADD	GREEDY-DELETE
BA	1.080	1.064	3.102
ER	1.073	1.094	2.120
RRG	1.080	1.096	2.201
WS	1.083	1.095	2.115
RGG	1.106	1.126	8.532
Chung–Lu	1.083	1.072	3.780

conjectured minimum value of $\lambda_{\max}$ in Table SM3 and find that the mean minimum achieved value of $\lambda_{\max}$ is at least 8–10% larger than the conjectured minimum. By contrast, when we use the GREEDY algorithm to reduce Q (with $\mu = 3\lambda_{\max}$) or λ_{total}, the means of the achieved objective-function values are within 3% of the conjectured minima.

The GREEDY-ADD algorithm performs similarly to the original GREEDY algorithm at reducing $\lambda_{\max}$ (see Table SM3). For the BA and Chung–Lu models, it even yields a larger mean reduction in $\lambda_{\max}$ than the GREEDY algorithm.

REFERENCES

[SM1] R. ALBERT AND A.-L. BARABÁSI, *Statistical mechanics of complex networks*, Reviews of Modern Physics, 74 (2002), pp. 47–97.

[SM2] M. BARTHELEMY, *Spatial Networks: A Complete Introduction: From Graph Theory and Statistical Physics to Real-World Applications*, Springer International Publishing, Cham, Switzerland, 2022.

[SM3] F. CHUNG AND L. LU, *Connected components in random graphs with given expected degree sequences*, Annals of Combinatorics, 6 (2002), pp. 125–145.

[SM4] P. ERDŐS AND A. RÉNYI, *On random graphs i*, Publ. Math. Debrecen, 6 (1959), pp. 290–297.

[SM5] B. K. FOSDICK, D. B. LARREMORE, J. NISHIMURA, AND J. UGANDER, *Configuring random graph models with fixed degree sequences*, SIAM Review, 60 (2018), pp. 315–355.

[SM6] F. P. KELLY, *Reversibility and Stochastic Networks*, Cambridge University Press, Cambridge, UK, 2011.

[SM7] J. H. KIM AND V. H. VU, *Generating random regular graphs*, in Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, ACM, 2003, pp. 213–222.

[SM8] M. PENROSE, *Random Geometric Graphs*, Oxford University Press, Oxford, UK, 2003.