https://pubsonline.informs.org/journal/mnsc

MANAGEMENT SCIENCE
Vol. 70, No. 12, December 2024, pp. 8988–9013
ISSN 0025-1909 (print), ISSN 1526-5501 (online)

Designing Sparse Graphs for Stochastic Matching with an Application to Middle-Mile Transportation Management

Yifan Feng,[a] René Caldentey,[b] Linwei Xin,[b,*] Yuan Zhong,[b] Bing Wang,[c] Haoyuan Hu[c]

[a] NUS Business School, National University of Singapore, Singapore 119245; [b] Booth School of Business, University of Chicago, Chicago, Illinois 60637; [c] Zhejiang Cainiao Supply Chain Management Co., Ltd, Hangzhou 310000, China

*Corresponding author

Contact: yifan.feng@nus.edu.sg, https://orcid.org/0000-0002-1695-9668 (YF); rene.caldentey@chicagobooth.edu, https://orcid.org/0000-0002-6767-9770 (RC); Linwei.Xin@chicagobooth.edu, https://orcid.org/0000-0002-8160-6877 (LX); Yuan.Zhong@chicagobooth.edu, https://orcid.org/0000-0002-8601-968X (YZ); lingfeng.wb@cainiao.com (BW); haoyuan.huhy@cainiao.com (HH)

Received: May 30, 2022
Revised: May 30, 2023
Accepted: July 10, 2023
Published Online in Articles in Advance: March 19, 2024

https://doi.org/10.1287/mnsc.2022.01588

Abstract. Given an input graph $G^{\text{in}} = (V, E^{\text{in}})$, we consider the problem of designing a sparse subgraph $G = (V, E)$ with $E \subseteq E^{\text{in}}$ that supports a large matching after some nodes in V are randomly deleted. We study four families of sparse graph designs (namely, clusters, rings, chains, and Erdős–Rényi graphs) and show both theoretically and numerically that their performance is close to the optimal one achieved by a complete graph. Our interest in the stochastic sparse graph design problem is primarily motivated by a collaboration with a leading e-commerce retailer in the context of its middle-mile delivery operations. We test our theoretical results using real data from our industry partner and conclude that adding a little flexibility to the routing network can significantly reduce transportation costs.

History: Accepted by David Simchi-Levi, optimization.
Funding: This work was supported by the University of Chicago Booth School of Business, an Alibaba Cainiao Research Grant, and the Singapore Ministry of Education [NUS Startup Grant WBS A-0003856-00-00].
Supplemental Material: Data and the online appendix are available at https://doi.org/10.1287/mnsc.2022.01588.

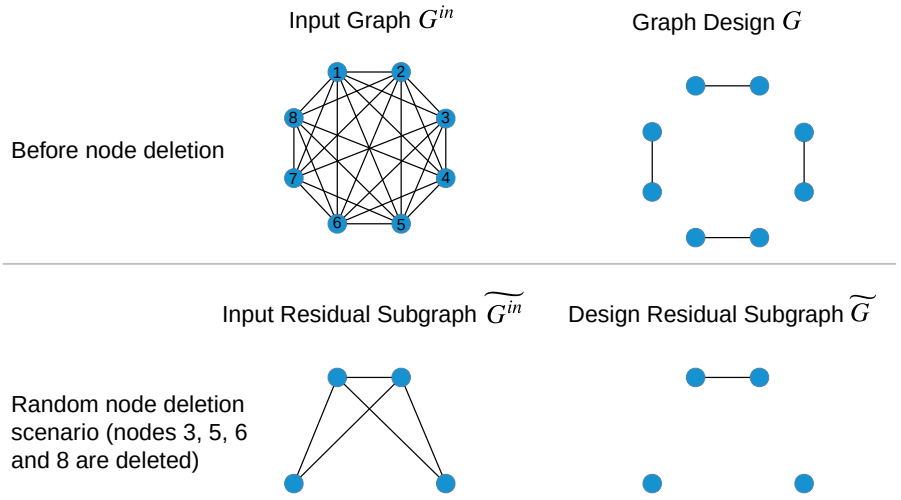
Keywords: transportation • middle-mile • flexibility • stochastic matching • long chain • e-commerce

1. Introduction

We study the problem of identifying a sparse subgraph for a given graph, such that the expected size of its maximum matching is close to that of the original graph, when nodes are randomly deleted. More specifically, we are given as input a graph $G^{\text{in}} = (V, E^{\text{in}})$, with node set V and edge set E^{in} (the superscript "in" is mnemonic for "input"), and a *node survival* probability $p \in (0, 1)$. Nodes in V are randomly and independently deleted with probability $q := 1 - p$, leaving a random subset of surviving nodes $\widetilde{V} \subseteq V$ and corresponding *residual* graph $\widetilde{G}^{\text{in}} = (\widetilde{V}, \widetilde{E}^{\text{in}})$, where $\widetilde{E}^{\text{in}} \subseteq E^{\text{in}}$ is the set of surviving edges. Denote by $\mu(\widetilde{G}^{\text{in}})$ the random cardinality of a maximum matching in $\widetilde{G}^{\text{in}}$, and by $\mathbb{M}(G^{\text{in}}) = \mathbb{E}[\mu(\widetilde{G}^{\text{in}})]$ its expected value. In this setting, we are interested in finding a sparse subset of edges $E \subseteq E^{\text{in}}$ such that the resulting subgraph $G = (V, E)$ has an expected maximum matching of similar size as $G^{\text{in}} = (V, E^{\text{in}})$ after node deletion, that is, $\mathbb{M}(G)$ is close to $\mathbb{M}(G^{\text{in}})$. See Section 2 for a more detailed problem description, and Figure 1 for an illustration.

Our interest in this problem is primarily motivated by a collaboration with a leading e-commerce retailer in the context of its middle-mile delivery operations. The middle-mile transportation system is responsible for moving parcels from large regional warehouses to last-mile delivery stations (these correspond to the nodes V in the graph G^{in}), which act as transfer centers between regional warehouses and local customers. Although middle-mile delivery is less known than last-mile delivery (i.e., transportation from last-mile delivery stations to customers) as it is less visible to the end customers, the middle-mile network can be the most expensive part of the trip (Naughton and Boyle 2019). The distribution of parcels from warehouses to delivery stations involves the coordination of transportation and other logistics resources, mainly trailer truck capacity. To this end, the controller constructs service routes by first linking delivery stations that can, in principle, be served by a single truck during a trip (links between stations correspond to the edges E^{in} in the graph G^{in}), while taking into account delivery lead times and other operational constraints. Then, to minimize transportation costs, the controller looks for ways to maximize the number of delivery stations served by a single truck. However, because

Figure 1. (Color online) Illustration of the Stochastic Matching Problem with Random Node Deletions



Notes. In this example, the input graph G^{in} has eight nodes and all possible edges except {8, 3} and {7, 4}. The graph design G has four edges. Under the random scenario in which nodes 3, 5, 6, and 8 are deleted, the size of the maximum matching in the input residual subgraph $\widetilde{G}^{\text{in}}$ is two, and that in the design residual subgraph $\widetilde{G}$ is one.

demands at stations are random, and truck loading capacities are limited, most trucks end up serving either one or two stations per trip in practice (only in rare occasions does a truck serve three or more stations in a trip). Thus, delivery stations are essentially partitioned into two groups based on their demand level:

i. High Demand: These are delivery stations whose demand is sufficiently large that a full truck is used to serve them.

ii. Low Demand: These are delivery stations whose demand is relatively low and need only a fraction of a truck capacity to serve them.

Thus, a truck can serve two stations simultaneously only if they both belong to group ii. Furthermore, because station demands are random, so are the sets of stations in groups i and ii in a given replenishment cycle. In our model, we assume that a station belongs to group ii with fixed probability p (this corresponds to the node survival probability for the graph G^{in}). Given the operating characteristics described above, it is not hard to see that the controller can minimize the number of trucks needed to serve the demand in all the delivery stations by finding a maximum cardinality matching on the random subgraph that results from eliminating all stations with high demand in group i (this is the residual graph $\widetilde{G}^{\text{in}}$).

A major challenge for the controller to implement service routes using the residual graph $\widetilde{G}^{\text{in}}$ is that $\widetilde{G}^{\text{in}}$ itself, as well as its maximum matching, can change significantly from one replenishment cycle to another, because of the stochastic nature of the demand. In other words, delivery routes of the trucks are not necessarily stable, creating additional complexity in the scheduling and coordination of drivers, most of whom operate as independent contractors. This is in contrast to a setting where demands are deterministic or relatively stable and where routes can be planned and fixed over different replenishment cycles, for example, a so-called milk-run delivery strategy. Thus, besides the goal of minimizing the number of trucks needed, the controller also wishes to create predictable delivery routes for truck drivers. One possible way to accomplish the latter is by reducing the number of feasible routes. (This corresponds to eliminating some of the edges in E^{in} to create the sparse subgraph G.) However, by reducing the number of feasible routes, the controller will also be needing more trucks to serve demand. It is precisely this trade-off that motivates us to investigate conditions under which one can reduce the number of delivery routes, without significantly increasingly the number of additional trucks needed. In the context of our graph-theoretic problem formulation, this corresponds to the problem of identifying a sparse subgraph G, such that $\mathbb{M}(G)$, the expected size of a maximum matching in its residual graph, is close to $\mathbb{M}(G^{\text{in}})$.

The sparse random matching problem that we study in this paper has also applications beyond the aforementioned truck scheduling problem. For example, in the context of kidney exchange, living donors are often incompatible with their intended recipients, and it becomes increasingly more common to coordinate and match among multiple donor–recipient pairs simultaneously. Blum et al. (2020) formulated a stochastic matching problem with random edge deletions to help reduce the number of pairwise compatibility tests while identifying almost as many compatible patient–donor pairs as exhaustive testing does. We refer to Roth et al. (2005) and

papers thereafter (e.g., Ashlagi et al. 2012, Dickerson et al. 2012, Ding et al. 2018) for more discussion of applying maximum matchings to kidney exchange. Maximum matchings also find their ways into other important contexts such as two-processor scheduling (Fujii et al. 1969), computational chemistry (May 2015), and online labor markets (Behnezhad et al. 2019a).

2. Model Formulation

2.1. Notation and Conventions

For a set S, we use $|S|$ to denote its cardinality, and use $\binom{S}{2}$ to denote the family of subsets of size two of S. A graph $G=(V,E)$ is defined by its node set V and edge set $E \subseteq \binom{V}{2}$. We say a graph G has size $n \in \mathbb{N}$ if $|V|=n$. With a slight abuse of notation, we will also use $|G|$ to denote the size of G. Without loss of generality, we label the nodes of G from 1 to n, and write $V=[n]:=\{1,\dots,n\}$ in shorthand. If $\{u,v\} \in E$, then nodes u, v are called *adjacent*, u is called a *neighbor* of v, and the edge $\{u, v\}$ is called *incident* to u (and to v). The *neighborhood* $\mathcal{N}(v):=\{u \in E : \{u,v\} \in E\}$ of a node $v \in E$ is the set of all of its neighbors. The *degree* $d(v):=|\mathcal{N}(v)|$ of v is the size of its neighborhood. We will denote by $d(G)$ the *average degree* or *density* of a graph $G=(V,E)$, that is, $d(G):=\sum_{v\in E} d(v)/|G| = 2|E|/|G|$. A graph $H=(V',E')$ is a *subgraph* of $G=(V,E)$ if $V' \subseteq V$ and $E' \subseteq E$. For a subset of nodes $V' \subseteq V$, the subgraph of G *induced* by V' is the subgraph $H=(V',E')$ with $E'=E \cap \binom{V'}{2}$. For a graph $G=(V,E)$, we use the tilde (~) notation $\widetilde{G}=(\widetilde{V},\widetilde{E})$ to denote the *residual* graph obtained from G after a random set of nodes are deleted. The sets $\widetilde{V}$ and $\widetilde{E}$ are the surviving nodes and edges, respectively.

A *matching* in G is a collection of disjoint edges in E, and a *maximum matching* is a matching that contains the largest possible number of edges. We use $\mu(G)$ to denote the cardinality of a maximum matching in G.

The following are special classes of graphs that we will use throughout this paper (see Figure 2 for an illustration):

- Regular: A graph is called *d-regular* if all of its nodes have the same degree d.
- Complete (K_n): This is a graph of size n in which every pair of distinct nodes is connected by a unique edge; that is, the entire graph is a single *clique* of size n.
- K-Cluster ($G_{n,K}$): This is a graph of size $n=Km$ for $K,m \in \mathbb{N}$, which is the disjoint union of m cliques of size K. (We refer to these cliques as *clusters*.) Note that $K_n = G_{n,n}$.
- K-Ring ($R_{n,K}$): This is a graph of size $n=Km$ for $K,m \in \mathbb{N}$ in which the set of nodes V can be partitioned into m disjoint subsets $\{V_1,V_2,\dots V_m\}$ with $|V_\ell|=K$ for all $\ell \in [m]$ and such that the set of edges E satisfies that $\{u,v\} \in E$ if and only if there exists $\ell \in [m]$ such that either (i) $u,v \in V_\ell$ or (ii) $u \in V_\ell$ and $v \in V_{\ell+1}$ (with the convention $V_{m+1}=V_1$).
- K-Chain ($C_{n,K}$): This is a graph of size n, where $n > K$, in which a pair of distinct nodes $i,j \in [n]$ are connected if $i-j \bmod n \le K$.[1] A motivation for the K-chain graph design comes from the operational flexibility literature (e.g., see Jordan and Graves 1995).
- Erdős–Rényi ($\mathrm{ER}_{n,\alpha}$): This is a random graph of size n in which an edge $\{u, v\}$ between two arbitrary nodes $u,v \in V$ is included with fixed probability $\alpha \in (0,1]$, independently over every other edge.

It is worth noticing that $G_{n,K}$, $R_{n,K}$, and $C_{n,K}$ are all regular graphs, whereas $\mathrm{ER}_{n,\alpha}$ is regular in expectation. Also, note that $d(G_{n,K})=K-1$, $d(R_{n,K})=\min\{3K-1,n-1\}$, $d(C_{n,K})=\min\{2K,n-1\}$, and $\mathbb{E}[d(\mathrm{ER}_{n,\alpha})]=(n-1)\alpha$.

Finally, we will make extensive use of the following asymptotic notation. For every pair of functions $f(\cdot)$, $g(\cdot):\mathbb{Z}_+ \to \mathbb{R}$, we write

1. $f(n)=O(g(n))$ if there exists $M<\infty$ and $N_0 \in \mathbb{Z}_+$ such that $|f(n)| \le Mg(n)$ for all $n \ge N_0$;
2. $f(n)=\Omega(g(n))$ if there exists $\delta>0$ and $N_0 \in \mathbb{Z}_+$ such that $f(n) \ge \delta|g(n)|$ for all $n \ge N_0$;
3. $f(n)=\Theta(g(n))$ if both $f(n)=O(g(n))$ and $f(n)=\Omega(g(n))$;
4. $f(n)=o(g(n))$ if for all $k>0$, there exists $N_0 \in \mathbb{Z}_+$ such that $|f(n)| \le kg(n)$ for all $n \ge N_0$;
5. $f(n)=\omega(g(n))$ if for all $k>0$, there exists $N_0 \in \mathbb{Z}_+$ such that $f(n) \ge k|g(n)|$ for all $n \ge N_0$;
6. $f(n) \sim g(n)$ if $f(n)/g(n) \to 1$ as $n \to \infty$.

2.2. Problem Formulation and Research Questions

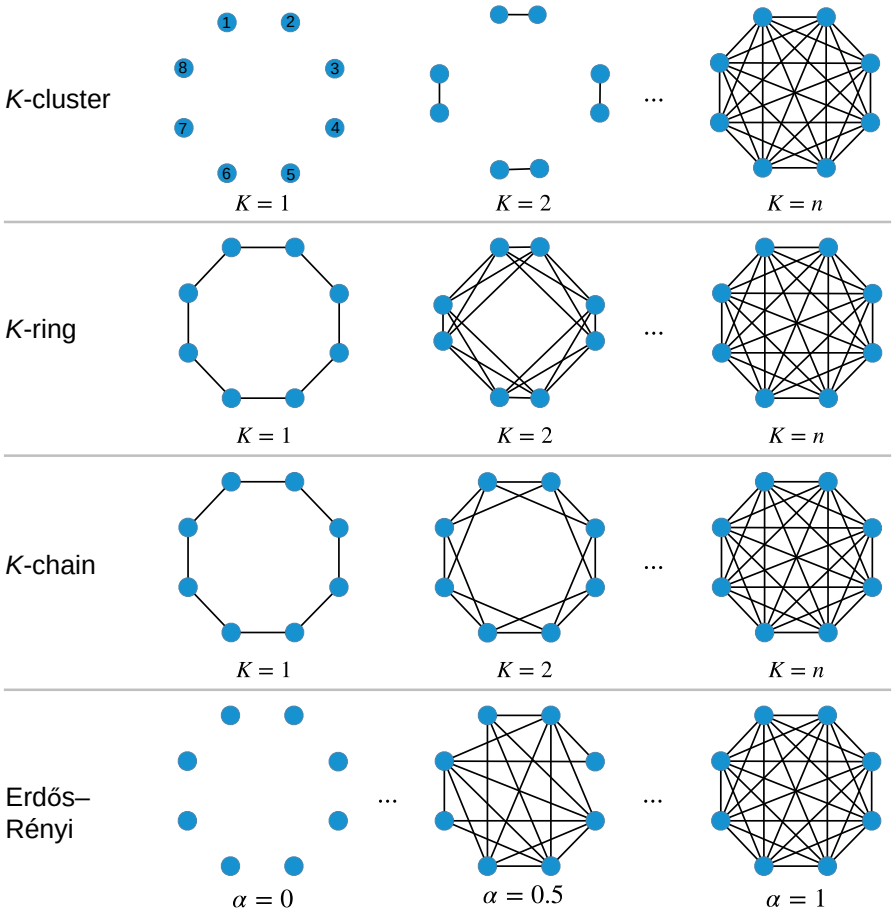
We are given an input graph $G^{\mathrm{in}}=(V,E^{\mathrm{in}})$ and a node survival probability $p \in (0,1)$. We also define $q:=1-p$ to be the *node deletion* probability. Let $\mathbb{G}:=\{G=(V,E): E \subseteq E^{\mathrm{in}}\}$ be the collection of subgraphs of G^{in} defined on the same node set V. After each node in V is deleted with probability q independently, a random subset of nodes, $\widetilde{V}$, remains, so that $\Pr(i \in \widetilde{V})=p$ for all $i \in V$. For each subgraph $G \in \mathbb{G}$, $\widetilde{V}$ induces a residual random subgraph $\widetilde{G}$. Denote by $\mathbb{M}(G)$ the expected cardinality of a maximum matching of $\widetilde{G}$, that is,

$$\mathbb{M}(G):=\mathbb{E}[\mu(\widetilde{G})]. \tag{1}$$

Our main research goal is to identify a *sparse* subgraph $G \in \mathbb{G}$ such that $\mathbb{M}(G)$ is *close* to $\mathbb{M}(G^{\mathrm{in}})$.

To measure sparsity, we use the *density* $d(G)$ of graph G as defined in Section 2.1. When the context is clear, we will often use d instead of $d(G)$. For example, a graph with no edges has $d=0$ and is the most sparse, whereas the complete graph K_n of size n has

Figure 2. (Color online) An illustration of *K*-Clusters, *K*-Rings, *K*-Chains, and ER Graphs



Note. The number of nodes is $n = 8$.

$d = n - 1$ and is the most dense. To compare $\mathbb{M}(G)$ with $\mathbb{M}(G^{\text{in}})$, we define $L(G)$, the *matching loss* function given by

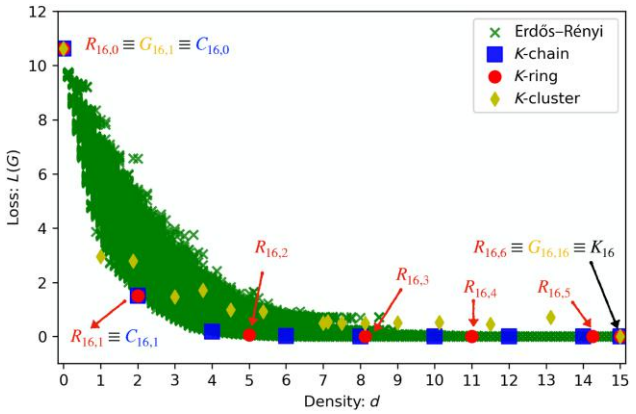
$$L(G) := 2[\mathbb{M}(G^{\text{in}}) - \mathbb{M}(G)], \tag{2}$$

which measures the expected number of additional unmatched nodes in $\widetilde{G}$, compared with $\widetilde{G}^{\text{in}}$.

It is not difficult to see that the density function $d(G)$ and loss function $L(G)$ are monotone with respect to graph G in the following sense: If $G_1, G_2 \in \mathbb{G}$ are both subgraphs of G^{in}, and G_1 is a subgraph of G_2, then $d(G_1) \le d(G_2)$, whereas $L(G_1) \ge L(G_2)$. Thus, there is an inherent trade-off between selecting a subgraph $G \in \mathbb{G}$ with a achieving small $d(G)$ and a small $L(G)$ simultaneously. We are interested in understanding this trade-off, as well as characterizing the entire density-loss spectrum and efficient frontier; that is, for a given G^{in} and a given a density budget, what is the smallest achievable loss?

Figure 3 illustrates the density versus loss trade-off for the case in which the input graph is K_{16}, a complete graph with size $n = 16$. The crosses correspond to 1,000,000 randomly generated instances of $\text{ER}_{16,\alpha}$, where α ranges from 0 to 1; the 16 diamonds depict the *K*-clusters $G_{16,K}$ for $K = 1, 2, \ldots, 16$; the squares represent the *K*-chains for $K = 0, 1, 2, \ldots, 8$; and the dots correspond to the *K*-rings $R_{16,K}$ for $K = 0, 1, \ldots, 6$. The value of the matching loss for each of these graphs is

Figure 3. (Color online) An Illustration of the Trade-Off Between Graph Density and Matching Loss for the Case in Which the Input Graph Is a Complete Graph of Size 16 (K_{16}) and the Node Survival Probability Is $p = 0.7$



computed over an average of 1,200 simulations using a node survival probability of $p = 0.7$. Figure 3 reveals two interesting features that we will formalize in later sections: (1) the performance of the K-ring and K-chain designs are consistently good across density levels and (2) random Erdős–Rényi graphs perform particularly well for medium to large density levels.

Mathematically, the problem of characterizing the efficient frontier between graph density and matching loss can be formulated as a stochastic program. Indeed, recall that for an arbitrary graph $G = (V, E)$, the problem of determining the cardinality of a maximum matching can be formulated as the integer program (see Wolsey 1998):

$$\mu(G) = \max\{\mathbf{1}y : A_G y \le \mathbf{1}, y \in \mathbb{Z}_+^{|E|}\},$$

where $A_G \in \{0,1\}^{|V|\times|E|}$ is the 0–1 node–edge incidence matrix of graph G, and $\mathbf{1}$ denotes a $|V|$-dimensional vector of all ones. Thus, for a given density d, the problem of determining a maximum cardinality matching subgraph $G \in \mathbb{G}$ with degree less than or equal to d can be formulated as the following two-stage stochastic program:

$$\max_{G\in\mathbb{G}:\, d(G)\le d} \{\mathbb{E}_{\boldsymbol{\delta}}[\max\{\mathbf{1}y : A_G y \le \mathbf{1}, y \le \boldsymbol{\delta}, y \in \mathbb{Z}_+^{|E|}\}], \tag{3}$$

where the expectation $E_{\boldsymbol{\delta}}[\cdot]$ is taken over the random vector $\boldsymbol{\delta} \in \{0,1\}^{|E|}$ such that for $\{u,v\} \in E$, we have $\delta_{\{u,v\}} = 1$ if and only if both nodes u and v survive node deletion.

The problem of computing a maximum cardinality matching (i.e., solving the inner maximization in (3)) can be done efficiently in order $O(\sqrt{|V|}|E|)$; see Micali and Vazirani (1980). On the other hand, we have not been able to leverage this knowledge to develop an efficient approach to solve Problem (3). For this reason, we will instead investigate the performance of four special classes of graph designs, K-clusters, K-rings, K-chains, and Erdős–Rényi graphs introduced in Section 2.1, and show that they have good performance both theoretically and numerically.

Let us also remark up front that the main questions of interest in this paper are markedly different from those typically addressed in the closely related stochastic matching literature (see Section 4 for a detailed comparison) in theoretical computer science. In works on stochastic matching, the goal is often to devise a computationally efficient procedure for constructing a sparse subgraph G, for any arbitrary input graph G^{in}, with a performance guarantee that $L(G)$ is at most a constant fraction of $\mathbb{M}(G^{\text{in}})$. In other words, the interest is in an algorithm for computing a single point on the density-loss spectrum, one for each G^{in}. In contrast, we are interested in the *entire* spectrum. It is a daunting challenge to fully characterize the spectrum over arbitrary input graphs G^{in}, so in this paper, we mainly focus on the case where G^{in} is a complete graph. In this case, we are able to obtain order-tight lower bounds on the density-loss efficient frontier, as well as tight characterizations of the density-loss trade-offs for classes of simple and interpretable graph designs (see Sections 3.1 and 5 for details). Another reason for focusing on this case is that it also appears to be the most practically relevant, based on our experiences in the industry collaboration (see Section 7 for details).

In this paper, we establish a variety of nonasymptotic results for input graphs of a given size, as well as asymptotic results where the graph sizes grow large. As we will see, asymptotic properties of the density-loss spectrum for graph designs provide many useful insights. More precisely, we consider a sequence of problem instances indexed by n, where n grows large. For each problem instance, we are given an input graph G^{in} with size n and a target loss level $l = l(n)$. For each n, we wish to identify a graph design G with degree $d = d(n)$, so that $L(G) \le l(n)$. We are particularly interested in two asymptotic regimes regarding $l = l(n)$:

Regime 1 (Relatively Negligible Losses): When $l = o(\mathbb{M}(G^{\text{in}}))$, the loss only constitutes a diminishing fraction of $\mathbb{M}(G^{\text{in}})$ for a large G^{in}. In other words, $\mathbb{M}(G)/\mathbb{M}(G^{\text{in}})$ can be arbitrarily close to one.

Regime 2 (Negligible Losses): When $l = O(1)$, the loss is bounded by a constant independent of the size of G^{in}.

With an asymptotic analysis, we wish to address the following question: What is the minimal level of $d = d(n)$ in the aforementioned two regimes?

2.3. Discussion of Model Assumptions

There are two key assumptions that we make in this paper that deserve further discussion.

2.3.1. Possible Graph Designs. A clear limitation of this paper is the decision to restrict the graph design problem to the specific classes of K-clusters, K-rings, K-chains, and Erdős–Rényi graphs. Our motivation for considering K-clusters, K-rings, and K-chains is twofold. First, these are straightforward and intuitive graph designs that can be easily implemented in practice (whereas a general solution to (3) might be difficult to interpret and execute). In fact, our interest in these graph designs is partly driven by our industry partner's current operations. Second, these graph designs are justified by their asymptotic optimality properties; see Section 3.1 for additional details. For instance, under both the K-ring and the K-chain designs, the matching losses decay *exponentially* with respect to the density. This property implies that K-rings and K-chains are *order optimal* designs, as our lower bound demonstrates that the decay rate cannot be

superexponential. Conversely, our interest in Erdős–Rényi graphs is primarily theoretical, as they can offer higher-order optimality in one specific limiting regime: namely, the exponent of the decay rate can be asymptotically matched in the *negligible loss* regime ($l = O(1)$) when the node deletion probability is high.

It is also worth noting that there is a clear parallel between our graph designs (i.e., *K*-clusters, *K*-rings, *K*-chains, and complete graphs) and the types of configurations that have been studied in the process flexibility literature (i.e., dedicated systems, *K*-chains, or full flexibility systems); see Section 4 for specific references. In particular, our *K*-clusters resemble dedicated systems, our *K*-rings and *K*-chains mimic *K*-chains in the process flexibility literature (e.g., a 1-ring emulates a long chain), and complete graphs correspond to full flexibility systems. At the same time, there is a significant difference between our problem and those studied in the flexibility literature; namely, our system is represented by a general graph rather than a bipartite one.

2.3.2. Two Stops per Truck.

Another possible valid critique of our model concerning the middle-mile logistics problem is that we consider only cases where a truck can visit a maximum of two delivery stations per trip. Although this is undoubtedly a limitation, this assumption is partly driven by the actual operations of our industry partner, where fewer than 5% of current routes involve more than two stops. In practice, the loading and unloading of parcels at a station is a time-consuming process that restricts the number of stops that can be scheduled on a truck's route. Simultaneously, we believe that generalizing our model to accommodate multistop routes presents an intriguing theoretical extension that could be tackled by reformulating the problem as a stochastic maximum matching problem on a random hypergraph, resulting in a significantly more complex problem. In fact, if trucks could serve an arbitrary number of delivery stations (provided their combined demand is less than the truck's capacity), then determining the minimum number of trucks needed to serve all delivery stations is equivalent to a bin packing problem, which is strongly NP-hard. Conversely, our constrained problem, which allows for a maximum of two stops per truck, admits a polynomial-time algorithm (see Section 4 for details).

3. Summary of Results and Takeaways

In this section, we provide a brief summary of our main contributions both in terms of (i) the theoretical results that we obtain for the random matching problem and (ii) the takeaways that our analysis provides in the context of the middle-mile logistics operation.

3.1. Theoretical Results

As mentioned earlier, our paper mainly focuses on the case in which $G^{\text{in}} = K_n$, the complete graph of size n. Under this case, we establish the following nonasymptotic results:

1. Theorem 1, which provides a lower bound of $npq^d - 1$, or, equivalently, $npe^{-\gamma_1 d} - 1$, where $\gamma_1 := \log(1/q)$, on the loss $L(G)$ incurred by any graph design G with density d;
2. Theorem 2, which provides an upper bound of $n/(d+1)$ on the loss $L(G_{n,K})$ for the *K*-cluster graph design $G_{n,K}$;
3. Theorem 3, which provides an upper bound of $\frac{3n}{d+1}q^{(d+1)/3}$, or, equivalently, $3ne^{-\gamma_2(d+1)-\log(d+1)}$, where $\gamma_2 := \gamma_1/3$, on the loss $L(R_{n,K})$ for a *K*-ring design $R_{n,K}$;
4. Corollary 1, which provides an upper bound of $\frac{4n}{d}q^{d/4}$, or, equivalently, $4ne^{-\gamma_3 d-\log d}$, where $\gamma_3 := \gamma_1/4$, on the loss $L(C_{n,K})$ for a *K*-chain design $C_{n,K}$.

For ease of reference, we summarize our nonasymptotic results in Table 1. These results have several important performance implications. First, *K*-rings and *K*-chains significantly improve the density-loss frontier compared with *K*-clusters: the loss decays *exponentially* in d under *K*-rings and under *K*-chains, compared with *reciprocally* under *K*-clusters. Second, the (leading) exponent in the loss upper bound under *K*-rings (*K*-chains, respectively) is $-\gamma_2 d = -\gamma_1 d/3$ ($-\gamma_3 d = -\gamma_1 d/4$, respectively), one-third (one-quarter, respectively) of $-\gamma_1 d$, the exponent in the universal loss lower bound. This means that to achieve any given loss target, the density required under a *K*-ring design (under a *K*-chain design, respectively) is at most three times (four times, respectively) the optimal density (ignoring lower-order terms), so *K*-rings and *K*-chains are families of *order-optimal* graph designs. Furthermore, our nonasymptotic results have a number of asymptotic implications:

1. First, the lower bound in Theorem 1 implies that we need an arbitrarily large density $d = \omega(1)$ to achieve a relatively negligible loss. In other words, if we fix the graph density, the loss will inevitably grow linearly in n. It also implies that to achieve a negligible loss, we need a density level of $d = \frac{1}{\gamma_1}\log n - O(1)$.
2. Second, if we only require the loss to be relatively negligible, *K*-clusters, *K*-rings, and *K*-chains all need a

Table 1. Summary of Nonasymptotic Results

	$L(G)$	Note
Lower bound	$npe^{-\gamma_1 d} - 1$ (Theorem 1)[a]	$\gamma_1 = \log\left(\frac{1}{q}\right)$
K-cluster ($G_{n,K}$)	$\frac{n}{d+1}$ (Theorem 2)	–
K-ring ($R_{n,K}$)	$3ne^{-\gamma_2(d+1)-\log(d+1)}$ (Theorem 3)[b]	$\gamma_2 = \frac{\gamma_1}{3}$
K-chain ($C_{n,K}$)	$4ne^{-\gamma_3 d-\log d}$ (Corollary 1)[c]	$\gamma_3 = \frac{\gamma_1}{4}$

[a]Note that this cell can be rewritten as $npq^d - 1$.
[b]Note that this cell can be rewritten as $\frac{3n}{d+1}q^{(d+1)/3}$.
[c]Note that this cell can be rewritten as $\frac{4n}{d}q^{d/4}$.

density of $d = \omega(1)$, an *order optimal* amount to meet the requirement. In other words, if we are less sensitive to the precise growth rate of L, both designs lead to effective sparse structures.

3. Third, if we require the loss to be negligible, K-clusters needs a density of $\Theta(n)$, whereas a density of $\frac{1}{\gamma_2}\log n$ is sufficient for K-rings ($\frac{1}{\gamma_3}\log n$, respectively, for K-chains).

In other words, if we are more sensitive to L, there is a salient improvement from K-clusters to K-rings and K-chains.

A natural question that arises from our nonasymptotic analysis is whether the constant factor gap between the universal loss lower bound and the upper bounds on the K-ring and K-chain losses can be closed. Although we cannot close this gap in full generality, we are able to do so in the asymptotic regime of negligible loss, and when the node deletion probability is large. More specifically, to achieve negligible loss, the minimal density required is at least $\frac{1}{\gamma_1}\log n - O(1)$, whereas the K-ring (K-chain, respectively) design needs a density of $\frac{1}{\gamma_2}\log n$, three times the lower bound ($\frac{1}{\gamma_3}\log n$, four times the lower bound, respectively). To close these gaps, we study the class of Erdős–Rényi graphs and find that an expected density of $\frac{1+\varepsilon}{\gamma_4}\log n$ for $\gamma_4 = 1 - q$ and any $\varepsilon > 0$ are sufficient to make the expected loss negligible (see Theorem 4). Because $\gamma_1 \sim \gamma_4$ as $q \to 1$, we conclude that the ER graph is *asymptotically optimal* as the most sparse design (with a constant-matching property) when $n \to \infty$, $q \to 1$, and we require the target loss level $l = O(1)$. Our analysis of the ER graph not only demonstrates its promising performance in a large graph with a large node-deletion probability, but also sheds light on the asymptotic tightness of our lower bound. We summarize our results for asymptotic analysis in Table 2.

As an extension to our main results, we also attempt to solve the problem in the general form where G^{in} is arbitrary. By adapting the proof techniques of Assadi and Bernstein (2019) (which focus more on random edge rather than node deletions), our solution enjoys the following desired properties: (i) for every G^{in} and $\varepsilon > 0$, we can compute a (deterministic) graph G in polynomial time; (ii) the maximum degree of G is at most $O(\log\frac{1}{\varepsilon p}/(\varepsilon^2 p))$; and (iii) $L(G) \le \frac{1+4\varepsilon}{3+2\varepsilon}\mathbb{M}(G^{\text{in}})$ (see Theorem 5 for more details). Our solution could be useful when one is interested in highly sparse (i.e., d is bounded by a constant) graphs with some performance guarantee on the loss, for instance, when G^{in} is sparse in the first place. We leave for future research a more comprehensive study of the density-loss spectrum for an arbitrary G^{in}.

Table 2. Summary of Asymptotic Results: Density $d(n)$ Required to Achieve the Target Loss $l(n)$[a]

	$l = o(n)$	$l = O(1)$	Note
Lower bound	$\omega(1)$	$\frac{1}{\gamma_1}\log n - O(1)$	—
K-cluster ($G_{n,K}$)	$\omega(1)$	$\Theta(n)$	—
K-ring ($R_{n,K}$)	$\omega(1)$	$\frac{1}{\gamma_2}\log n$	—
K-chain ($C_{n,K}$)	$\omega(1)$	$\frac{1}{\gamma_3}\log n$	—
ER graph ($\text{ER}_{n,\alpha}$)	—	$\frac{1+\varepsilon}{\gamma_4}\log n$ (Theorem 4)[b]	$\gamma_4 = (1-q) \sim \gamma_1$ as $q \to 1$

[a]The cells for K-cluster, K-ring, K-chain, and ER graph contain $d(n)$ values sufficiently high for the target loss $l(n)$.

[b]Note that Theorem 4 implies a stronger result: the average density of $\frac{1+\varepsilon}{\gamma_4}\log n$ can achieve $o(1)$ loss.

3.2. Takeaways from Simulation and Case Studies

In Section 6, we test the robustness of our theoretical results using simulated data. We try out various parameter settings of our model, as well as relax some modeling assumptions by incorporating correlation in node deletions. We find that our insights from the theoretical analysis are qualitatively robust. In particular, although many of our theoretical results involve the asymptotic regime where the graph size n is large, the main insights still hold when n is relatively small (e.g., $n = 30$).[2] In addition, we carry out comparative statics on the problem parameters to obtain some further guidance in the sparse graph design problem.

In Section 7, we showcase how to apply our theory to a middle-mile transportation problem. Based on a real data set from our industry partner, we demonstrate how the proposed solutions to the sparse graph design problem can help reduce the e-commerce retailer's middle-mile transportation costs. We evaluate the performance of our graph designs using two different approaches. In the (binary) model-based approach, we calibrate the best-fitting parameter values for the demand model. In the fully empirical approach, we relax the model assumptions and instead investigate the benefits of our graph design *retrospectively*. The results consistently suggest that adding a little flexibility to the transportation network can significantly reduce transportation costs.

4. Related Literature

Our paper is related to two streams of literature. Methodologically, we contribute to the literature on maximum matching in (random) graphs, which dates back to the 1930s (for comprehensive reviews, see Bollobás and Béla 2001, Frieze and Karoński 2016). In terms of applications, we contribute to the recent and growing literature on flexible operations management and e-commerce middle-mile transportation.

4.1. Matching in (Random) Graphs

The classic literature on maximum matching largely concerns two questions: For a given (random) graph,

what is the size of its maximal matching? Also, how does one efficiently find one? Regarding the first question, a pioneering result is by Berge (1958), who generalizes the results of Hall (1935) and Tutte (1947) and characterizes the size of a maximum matching of a general deterministic graph, known as the Tutte–Berge formula. Sharper characterizations can be obtained if we add randomness to the graph. For representative examples, Erdős and Rényi (1966) show that if we randomly pick a graph with average degree $d \geq \log n + \omega(1)$,[3] where n is the number of nodes, a perfect matching exists with high probability. Here, a *perfect* matching is a matching with a size of $\lfloor n/2 \rfloor$, where $\lfloor \cdot \rfloor$ is the floor function. Karp and Sipser (1981) (and also Aronson et al. 1998) consider essentially the same type of random graph, known as the Erdős–Rényi graph, but for a sparser regime where expected degree is a constant, that is, $\mathbb{E}[d] = c$. They construct a greedy algorithm and find that, with high precision and high probability, the expected size of a maximum matching can be approximated by $nr(c)$, where $r(c)$ is an explicit function of c.

Regarding the computational question, Edmonds (1965) constructs the first polynomial-time algorithm for an arbitrary deterministic graph with a running time of $O(n^4)$. Improvements in the running time were subsequently derived by Hopcroft and Karp (1973) for bipartite graphs, and Kameda and Munro (1974), Even and Kariv (1975), Micali and Vazirani (1980), and Blum (1990) for general graphs (see also Galil 1986 for a review). It is perhaps not surprising that more efficient algorithms can be developed for random graphs. Angluin and Valiant (1979) construct an algorithm that finds in $O(n \log n)$ time a perfect matching in a (sufficiently dense) Erdős–Rényi graph with high probability. Chebolu et al. (2010) augment the greedy matching in Karp and Sipser (1981) and show that for sparse Erdős–Rényi graphs, only $O(n)$ time is needed to find an exact maximum matching.

Our work is closely related to the stochastic matching literature, a more modern development of the maximum matching problem (see Assadi et al. 2016, Behnezhad et al. 2019b, Behnezhad et al. 2020, Blum et al. 2020, and references therein). Stochastic matching concerns how to "sparsify" a given graph that also supports a large matching under random graph disruptions. The prototypical problem studied in this stream of literature consists of selecting a subgraph of a given (possibly edge-weighted) input graph that contains a large matching after edges are removed independently at random. A state-of-the-art result is obtained by Behnezhad et al. (2020): Only a subgraph with maximum degree $d_{max} = O(1)$ (independent of the graph size) is needed to achieve a $(1-\varepsilon)$-ratio approximation of the input graph.

There are several notable and important differences between the current stochastic matching literature and our paper:

1. First, in our setting, the randomness in the graph comes from node deletion, so edges are no longer independently deleted. This is in direct contrast to the typical setting that assumes that edges are independently removed. Some ideas are transferable. For example, we utilize the techniques from Assadi and Bernstein (2019) and obtain a $(2/3-\varepsilon)$ approximation ratio in Theorem 5. However, stronger results such as those by Behnezhad et al. (2020) heavily exploit the independence of edge deletions, and it is not clear how to adapt them to our setting.

2. Second, when the input graph is complete, we obtain more comprehensive results than the typical ones in the stochastic matching literature. Typical papers in this literature look for a graph with $O(1)$ degree to achieve a constant-ratio approximation, which translates to a linear loss $L(G) = \Omega(n)$ in our formulation. In comparison, we are interested in the whole density-loss spectrum. In particular, we also wish to find a graph with $\omega(1)$ degree to achieve a sublinear loss $L(G) = o(n)$, but with the optimal degree-loss dependence. In equivalent terms, we strive for $(1-\varepsilon)$ approximations of the complete graph with $\varepsilon = o(1)$ and explore the minimal degrees needed for different decay rates of ε.

3. Third, our methodology produces simple and easy-to-interpret graph designs (e.g., K-cluster, K-rings, or K-chains). In comparison, the focus of this literature is algorithmic, and the produced graphs are usually complicated (e.g., Behnezhad et al. 2019a, 2020; Blum et al. 2020). From a practical standpoint, we believe that the simplicity of the graph structure, besides sparsity, is an important feature in its own right.

4.2. Flexible Operations

On the application side, our paper contributes to the growing literature on flexible operations. Since the seminal work of Jordan and Graves (1995), there has been a considerable amount of research on designing flexible manufacturing processes (e.g., Chou et al. 2008, Simchi-Levi and Wei 2012, Deng and Shen 2013, Wang and Zhang 2015, Shi et al. 2019). The idea of flexibility operations has also been applied to many other contexts, including supply chains (e.g., Graves and Tomlin 2003), call centers (e.g., Iravani et al. 2007), dual-sourcing inventory systems (e.g., Allon and Van Mieghem 2010, Xin and Goldberg 2018), newsvendor networks (e.g., Bassamboo et al. 2010), queueing systems (e.g., Bassamboo et al. 2012, Tsitsiklis and Xu 2017, Afèche et al. 2022), e-commerce fulfillment (e.g., Lyu et al. 2019, Asadpour et al. 2020, Xu et al. 2020, DeValve et al. 2023), and vehicle routing

(Ledvina et al. 2022). Our paper reemphasizes the principle that a little flexibility goes a long way in the new context of e-commerce middle-mile transportation. At the same time, our paper also has several distinctive features:

1. Conceptually, most of the aforementioned papers study flexibility in the context of matching supply with demand so as to maximize the amount of fulfilled demand. By contrast, in our paper, there is no real constraint on the supply size (e.g., the number of available trucks), and all demand is eventually fulfilled. Instead, we strive to understand flexibility in a logistics network setting where the *operational decisions* of *how* to fulfill demand become central to minimize fulfillment costs.

2. In terms of modeling, the problem of matching supply with demand is commonly represented by a *bipartite* graph, in which edges encode flexibility. By contrast, our model is fundamentally different, as we deal only with demand nodes and therefore we use a *general* graph to represent different flexibility structures, where edges represent specific ways to fulfill demands (i.e., indirect shipping routes). Methodologically, the tools that are normally used to study flexibility in bipartite graphs (such as max-flow formulations) are ill suited in our setting with a general graph, and so our analysis relies on a different set of tools such as (random) graph theory.

3. Finally, there are also qualitative differences between supply–demand flexibility and our transportation flexibility. For example, our numerical studies reveal that highly sparse graphs are even more effective when node deletions are positively correlated. This stands in contrast to the typical finding in supply–demand flexibility that positive demand correlation reduces the benefit of flexibility (see, e.g., Jordan and Graves 1995).

Perhaps the recent paper by Ledvina et al. (2022) is most relevant to ours. It considers a vehicle routing problem (VRP) with stochastic demand and explores how to limit the amount of flexibility in the VRP routes without (significantly) affecting the system's performance compared with the unconstrained optimal solution. At a high level, both our paper and Ledvina et al. (2022) consider the problem of flexibility designs in the context of logistics and transportation operations. However, there are also significant differences between the two works. A major difference is that in Ledvina et al. (2022), a vehicle is allowed to visit arbitrarily many stations in one trip, whereas in our model, a vehicle can visit at most two stations in a trip (see Section 2.3 for detailed reasons on why we impose this constraint). Another difference is that node demand in Ledvina et al. (2022) can be split across vehicles, whereas, in the analytical portion of our paper, we consider a binary demand model with no demand splitting. Among other things, the features of at most two stops per vehicle and no demand splitting led us to make connections with graph matching and formulate a very different model from that in Ledvina et al. (2022). Consequently, our analytical approach and theoretical results are also substantially different from those in Ledvina et al. (2022).

4.3. E-Commerce Middle-Mile Transportation

The indirect shipping problem we consider is usually classified as middle-mile transportation because it concerns shipping from warehouses to last-mile delivery stations. Whereas there is a vast academic literature on last-mile delivery, the literature on middle-mile transportation is rather limited. One notable exception is Chen et al. (2020), which takes a data-driven approach to design a middle-mile transportation network.

5. Complete Input Graphs

In this section, we study the performance of four special classes of sparse graph designs, namely, K-clusters, K-rings, K-chains, and Erdős–Rényi graphs, for the case in which the input graph is complete, $G^{\text{in}} = K_n$. We focus on complete input graphs mainly for tractability reasons, but we note that in many applications, including our transportation problem discussed in Section 7, this is a reasonable assumption.

Before moving into the analysis of the four graphs designs, we first derive a general lower bound for the loss function for an arbitrary graph that will prove useful in assessing the suboptimality of K-clusters, K-rings, K-chains, and ER graphs.

5.1. Lower Bound

The following theorem provides a nonasymptotic lower bound on the loss function $L(G)$ for an arbitrary graph design G in terms of its average density $d(G)$ when the input graph G^{in} is complete.

Theorem 1. *Suppose $G^{\text{in}} = K_n$; then, for any graph G with density $d = d(G)$, the loss is bounded below by $L(G) \geq (npq^d - 1)^+ = (npe^{-\gamma_1 d} - 1)^+$, where $\gamma_1 = \log(1/q) > 0$.*

Proof of Theorem 1. We include the proof here, because it is instructive and contains several performance and design implications that we use in what follows. To prove the theorem, we first need the following lemma, whose proof can be found in the online appendix, Section EC.2.

Lemma 1. *We have $\mathbb{M}(K_n) = \frac{1}{2}\left(np - \frac{1}{2} + \frac{1}{2}(1-2p)^n\right)$. As a result, $(np-1)/2 \leq \mathbb{M}(K_n) \leq np/2$.*

Next, for any graph G with density d, we provide a lower bound on the expected number of isolated nodes in $\widetilde{G}$, the residual subgraph obtained from G

after random node deletions. Specifically, an isolated node in $\widetilde{G}$ is one that survives the node deletion, but whose neighbors in G are all deleted. Because node deletions are independent, for any given node i with node degree d_i, the probability that it becomes an isolated node in $\widetilde{G}$ is pq^{d_i}. Thus, the expected number of isolated nodes after random node deletion can be lower bounded as

$$\sum_{i=1}^{n} pq^{d_i} \geq npq^{\left(\sum_{i=1}^{n} d_i\right)/n} = npq^{d} = npe^{-\gamma_1 d}, \quad \text{where } \gamma_1 = \log(1/q). \tag{4}$$

The inequality in (4) follows from Jensen's inequality. Because every isolated node is unmatched in the residual subgraph $\widetilde{G}$, we have also obtained a lower bound on the expected number of unmatched nodes. It follows that

$$\begin{aligned} L(G) &= 2\mathbb{M}(K_n) - 2\mathbb{M}(G) \\ &\geq np - 1 - \mathbb{E}[2\mu(\widetilde{G})] && [2\mathbb{M}(K_n) \geq np - 1; \text{ by Lemma 1}] \\ &= \mathbb{E}[|\widetilde{G}| - 2\mu(\widetilde{G})] - 1 && [\mathbb{E}|\widetilde{G}| = np] \\ &\geq \mathbb{E}[\text{number of isolated nodes in } \widetilde{G}] - 1 \\ &\geq npq^{d} - 1 = npe^{-\gamma_1 d} - 1. && [\text{by (4)}]. \end{aligned}$$

The proof gets completed by noticing that the loss function is nonnegative.

Theorem 1 states that the loss $L(G)$ is at least an exponentially decaying function of the density d for any graph G. It also has several asymptotic implications. For example, if we require the loss to be *relatively negligible*, that is, $L(G) = o(n)$ (because $\mathbb{M}(K_n) = \Theta(n)$, $L(G) = o(\mathbb{M}(K_n))$ is equivalent to $L(G) = o(n)$), we need $e^{-\gamma_1 d} = o(1)$, which holds if and only if $d = \omega(1)$. Similarly, if we require the loss to be *negligible*, that is, $L(G) = O(1)$, we need, after a rearrangement of terms,

$$d \geq \frac{\log n + \log p - \log(O(1) + 1)}{\gamma_1} = \frac{1}{\gamma_1}\log n - O(1).$$

The proof of Theorem 1 also provides some insights into what constitutes a good graph design. Specifically, the inequality in (4) becomes tight when G is a regular graph; namely, given the density d of G, the expected number of isolated nodes is minimized when G is regular. This observation, in part, supports our choice of K-cluster, K-ring, K-chain, and Erdős–Rényi as viable graph designs. Let us also note that the proof of Theorem 1 uses a similar idea as the proof of theorem 6 in Chen et al. (2015).

Remark 1. Note that the lower bound in Theorem 1 is uninformative when $npe^{-\gamma_1 d} - 1 \leq 0$, that is, when $d \geq \frac{\log(np)}{\gamma_1}$. It is also worth noticing that the result in Theorem 1 extends to random graphs, by reinterpreting d as the graph's "expected density."

5.2. *K*-Clusters

The first family of graph designs that we consider, which we call K-clusters, is simple and motivated by industry practices. A K-cluster $G_{n,K}$ is defined as a disjoint union of n/K cliques, which we call *clusters*, each of which has K nodes. In other words, we can write a K-cluster as $G_{n,K} = ([n], E^C_{n,K})$, where $E^C_{n,K}$ is the set of edges, such that

$$\{i,j\} \in E^C_{n,K} \text{ if and only if } \lceil i/K \rceil = \lceil j/K \rceil, \tag{5}$$

where $\lceil \cdot \rceil$ is the ceiling truncation function. A K-cluster graph is a regular graph uniquely defined by each pair (n, K). In addition, there is a one-to-one correspondence between the parameter K and its density $d = K - 1$.[4]

Example 1. Consider a K-cluster $G_{n,K}$ with $n = 8$ and $K = 2$. The node set is $[n] = \{1,2,3,4,5,6,7,8\}$, and the edge set is $E^C_{n,K} = \{\{1,2\},\{3,4\},\{5,6\},\{7,8\}\}$. The density is $d = 1$. See Figure 2 for an illustration of this example.

Our interest in studying the performance of K-clusters is motivated by its simplicity and the fact that our industrial partner is currently implementing a version of this type of routing strategy. In the context of our middle-mile transportation problem, K-clusters correspond to the idea of breaking the entire transportation network into isolated subnetworks, which are operated completely independently of each other. In practice, these subnetwork typically represent different geographical zones with prespecified delivery routes.

The following theorem characterizes the matching loss performance of a K-cluster graph.

Theorem 2. *We have* $\frac{n}{d+1}\min\{p, 1-p\} - \frac{1}{2} \leq L(G_{n,K}) \leq \frac{n}{d+1}$.

We defer proofs to the online appendix unless specified otherwise.

Roughly speaking, Theorem 2 states that the loss $L(G_{n,K})$ is characterized by a reciprocally decaying function of the density d; namely, $L(G_{n,K}) = \Theta\left(\frac{n}{d}\right)$. As a result, using K-clusters, a density of $d = \omega(1)$ (no matter how slow the growth rate is) is sufficient to ensure that the loss is relatively negligible. In light of the lower bound result in Theorem 1, this is the smallest possible order. Therefore, in the regime where n grows large and we require only $L(G_{n,K}) = o(n)$, even a simple design such as a K-cluster leads to an effective sparse structure. In the context of our middle-mile transportation problem, this result partly justifies the emergence of K-cluster-type designs in the practice of our industrial partner.

However, K-clusters are less effective if we are more sensitive about matching losses. For example, if we require the loss to be negligible, the graph density that is needed must satisfy $d \geq \frac{\min\{p,1-p\}n}{1/2+L(G_{n,K})} - 1 = \frac{\min\{p,1-p\}n}{1/2+O(1)} - 1 = \Theta(n)$ for $p \in (0,1)$, that is, must exhibit linear growth with the input graph size. This observation motivates us to consider more effective graph designs.

5.3. *K*-Rings and *K*-Chains

To address some of the limitations of K-clusters, we consider the K-ring and K-chain designs, both of which have better matching loss performances. We chose to discuss K-rings and K-chains in the same subsection because these are closely related graph designs. In fact, as we will see in the sequel, performance bounds on K-chains can be obtained as a simple corollary of results on K-ring performances (see Corollary 1).

Let us begin by describing the structure of a K-ring design. Recall that a K-cluster is defined as a disjoint union of $\frac{n}{K}$ clusters of size K. A K-ring adds edges to a K-cluster by linking every pair of nodes in adjacent clusters. More specifically, we can write a K-ring as $R_{n,K} = ([n], E^R_{n,K})$, where $E^R_{n,K}$ is the set of edges, such that

$$\{i,j\} \in E^R_{n,K} \text{ if and only if either } (1)|\lceil j/K\rceil - \lceil i/K\rceil| \leq 1 \text{ or } (2)\ \lceil i/K\rceil = 1 \text{ and } \lceil j/K\rceil = \frac{n}{K}. \tag{6}$$

For each fixed n, there is a one-to-one correspondence between K and its density $d = 3K - 1$.[5]

Example 2. Consider a K-ring $R_{n,K}$ with $n = 8$ and $K = 2$. The node set is $[n] = \{1,2,3,4,5,6,7,8\}$ and the edge set is $E^R_{n,K} = \{\{1,2\},\{1,3\},\{1,4\},\{1,7\},\{1,8\},\{2,3\},\{2,4\},\{2,7\},\{2,8\},\{3,4\},\{3,5\},\{3,6\},\{4,5\},\{4,6\},\{5,6\},\{5,7\},\{5,8\},\{6,7\},\{6,8\},\{7,8\}\}$. The density is $d = 5$. See Figure 2 for an illustration of this example.

We characterize the performance of K-rings below.

Theorem 3. *We have* $L(R_{n,K}) \leq \frac{3n}{d+1} q^{(d+1)/3} = 3ne^{-\gamma_2(d+1)-\log(d+1)}$, *where* $\gamma_2 = \frac{1}{3}\log\left(\frac{1}{q}\right)$.

In comparison with Theorem 2, we find that K-rings significantly improve the density-loss frontier compared with K-clusters: the loss decays *exponentially* fast in density d under K-rings rather than *reciprocally* fast under K-clusters. Moreover, K-rings are in fact a family of *order-optimal* graph designs: Using a loose upper bound $L(R_{n,K}) \leq 3nq^{(d+1)/3}$ from Theorem 3, and the lower bound in Theorem 1, it can be computed that to achieve any given target loss l, it suffices to employ a K-ring with density d satisfying

$$d \leq 3\left(d_{\mathrm{OPT}} + \frac{\log(3/p)}{\log(1/q)}\right) = 3d_{\mathrm{OPT}} + O(1), \tag{7}$$

where d_{OPT} is the optimal required density. Note that Inequality (7) holds uniformly over all target loss l, and that the required density under K-rings is at most three times the optimal density, ignoring lower-order terms, implying the order optimality of K-rings. In particular, the K-ring design also has order-optimal performances in the two asymptotic regimes that we consider:

1. With a density of $d = \omega(1)$ (regardless how slow the growth rate is), a K-ring achieves a relatively negligible loss: $L(R_{n,K}) \leq \frac{3n}{d+1} \leq \frac{3n}{\omega(1)+1} = o(n)$.

2. With a density of $d = \frac{1}{\gamma_2}\log n$, a K-ring achieves a negligible loss: $L(R_{n,K}) \leq 3nq^{d+1} \leq 3nq^{\log n/\gamma_2} = O(1)$.

Let us now provide some remarks on the main ideas used in the proof of Theorem 3. The key observation is that the residual graph of a K-ring after random node deletion is the union of several connected components, where each component is a *chain of subclusters*, in the following sense: every subcluster in the component is a clique of size at most K, and all pairs of nodes in adjacent subclusters are linked (hence, a chain of subclusters). This special structure of the connected components implies that at most one node is left unmatched in a maximum matching for each component. Thus, to characterize the node loss of a K-ring, it essentially suffices to count the expected number of components in the residual graph, which leads to the upper bound in Theorem 3 (there is also a corner case that needs to be treated separately; see Section EC.4 of the online appendix for details).

We now turn our attention to the K-chain design. Recall that a K-chain is a graph in which a pair of nodes $i,j \in [n]$ are connected if $i - j \bmod n \leq K$. It follows that a K-chain is a d-regular graph with density $d = \min\{2K, n-1\}$. See Figure 2 for examples of K-chains.

The K-chain and K-ring are closely related graph designs. Indeed, as the following lemma states, roughly speaking, a K-ring is always "sandwiched" between a K-chain and a $(2K-1)$-chain.

Lemma 2. *For any positive integers n and K, where $n > K$ and is divisible by K, the K-chain $C_{n,K}$ is a subgraph of the K-ring $R_{n,K}$, which in turn is a subgraph of the $(2K-1)$-chain $C_{n,2K-1}$.*

Intuitively, Lemma 2 suggests that K-rings and K-chains should have similar performances, because they have similar structures and densities. Indeed, the following corollary is a simple consequence of Lemma 2 and Theorem 3 that provides an upper bound on the performance loss of a K-chain.[6]

Corollary 1. *For any positive integers n and K, where $n > K$ (and n is divisible by $\lfloor K/2 \rfloor$), let d denote the density of*

the K-chain $C_{n,K}$. Then,

$$L(C_{n,K}) \le L(R_{n,\lfloor K/2\rfloor}) \le \frac{4n}{d} q^{d/4} = 4ne^{-\gamma_3 d - \log d},$$

where $\gamma_3 = \frac{1}{4}\log\left(\frac{1}{q}\right)$.

We omit the proofs of Lemma 2 and Corollary 1 because they follow directly from the definitions of *K*-rings and *K*-chains and Theorem 3.

Similar to *K*-rings, *K*-chains are a family of *order-optimal* graph designs: to achieve any given target loss l, it suffices to employ a *K*-chain with density d satisfying

$$d \le 4d_{\mathrm{OPT}} + O(1), \tag{8}$$

where d_{OPT} is the optimal required density. Inequality (8) implies that to achieve any target loss l, the required density under *K*-chains is at most four times the optimal density, ignoring lower-order terms. Furthermore, similar to *K*-rings, the *K*-chain design achieves a relatively negligible loss with a density of $d = \omega(1)$, and a negligible loss with a density of $d = \frac{1}{\gamma_3}\log n$, where $\gamma_3 = \frac{1}{4}\log\left(\frac{1}{q}\right)$.

We conclude this subsection by noting that in the context of our middle-mile transportation problem, both the *K*-ring and *K*-chain transform into interpretable designs that utilize only "local" connections, in the sense that only nodes with indices that are nearby are connected under these designs. As a result, the *K*-ring and *K*-chain are designs better at respecting physical distances among stations, compared with, for example, the Erdős–Rényi design, which utilizes long-distance links with constant probability. These advantages make the *K*-ring and *K*-chain designs particularly useful in practice. We provide more details in our simulation and case study in Sections 6 and 7.

5.4. Erdős–Rényi Graphs

In the pursuit of higher-order optimality, we also consider and analyze ER graphs, a well-celebrated random graph family (see Erdős and Rényi 1966). An ER graph $\mathrm{ER}_{n,\alpha}$ is obtained by generating each edge independently with probability $\alpha \in (0,1)$, and the expected density of such a graph is $\mathbb{E}[d] = (n-1)\alpha$.

Our intuition for why ER graphs can be good candidates to study is twofold. First, the independence of the edge generation provides analytical tractability. Roughly speaking, it preserves many statistical properties of ER graphs after node deletion. Hence, we can largely reduce the problem into an equivalent setting without node deletion; see the discussion after Theorem 4 for more details. Second, ER graphs are well known to have good properties asymptotically. More specially, in the asymptotic regime where both n and α grow, the node degrees are independent and identically distributed (i.i.d.) with mean $n\alpha$. Therefore, one may expect an ER graph to be close to a symmetric and regular graph. In light of Fact 2, our intuition is that the loss for ER graphs is small in this regime. We summarize our main result for ER graphs below.

Theorem 4. *For every $q \in (0,1)$, $\varepsilon > 0$, and Erdős–Rényi graph $\mathrm{ER}_{n,\alpha}$ such that $\mathbb{E}[d] \ge \frac{1+\varepsilon}{\gamma_4}\log n$ for $\gamma_4 = 1-q$, we have $\mathbb{E}[L(\mathrm{ER}_{n,\alpha})] = o(1)$, where the expectation is taken over the random realization of $\mathrm{ER}_{n,\alpha}$.*

Theorem 4 describes the asymptotic optimality of ER graphs. Recall that Theorem 1 implies the density to achieve a negligible loss is at least $\frac{1}{\gamma_1}\log n - O(1)$. In comparison, the required density of ER graphs is (at most) $\frac{1+\varepsilon}{\gamma_4}\log n$ in expectation for any $\varepsilon > 0$. Because $\gamma_1 = \log(1/q) \sim 1 - q = \gamma_4$ as $q \to 1$, we conclude that the ER graph is asymptotically optimal as the most sparse design (with a constant-matching property) in a regime where $n \to \infty, q \to 1$, and the loss is negligible.

The significance of Theorem 4 is twofold. First, it demonstrates (theoretically) that *K*-rings can be further improved when the graph is large, the node-deletion probability is high, and we are more sensitive to *L*. Second, it suggests that the lower bound in Theorem 1, which is derived from counting the number of isolated nodes, is (somewhat surprisingly) tight, at least asymptotically in the aforementioned regime. Additionally, we offer a concise overview of the technical contributions related to proof techniques in the online appendix.

5.4.1. Discussion of ER Graphs When $\mathbb{E}[d] = o(\log n)$.

One may wonder how ER graphs would perform when they are substantially more sparse than in the critical case. Although we do not rigorously characterize the loss of ER graphs when $\mathbb{E}[d] = o(\log n)$, here we provide a plausible argument quantifying the loss for ER graphs when we fix the (expected) density $d_0 = \mathbb{E}[d]$, and as the number of nodes grows large. To do so, we recall the following result adapted from Frieze (1986).

Fact 1. As $m \to \infty$, the probability that an ER graph $\mathrm{ER}_{m,\alpha}$ with a fixed density $c = (m-1)\alpha$ contains a matching of size $\frac{1}{2}(1-(1+o(1))e^{-c})m$ goes to one, where $o(1) \to 0$ as $c \to \infty$.

In other words, Fact 1 states that given a sufficiently large density c, a large ER graph with m nodes contains a matching of size that is approximately equal to $\frac{1}{2}(1-e^{-c})m$, with high probability. Formally substituting $m = np$ and $c = pd_0$ into the preceding expression, where np is the expected number of nodes of $\widetilde{\mathrm{ER}}_{n,\alpha}$, $d_0 = (n-1)\alpha$ is the density of $\mathrm{ER}_{n,\alpha}$, and pd_0 is the density of $\widetilde{\mathrm{ER}}_{n,\alpha}$, and formally replacing the with-

high-probability statement with an in-expectation one, we expect that

$$\mathbb{M}(\mathrm{ER}_{n,\alpha}) \geq \frac{1}{2} np(1 - e^{-pd_0}).$$

Using the fact that $\mathbb{M}(K_n) \leq np/2$ from Lemma 1, our calculation suggests that

$$L(\mathrm{ER}_{n,\alpha}) \leq npe^{-pd_0}.$$

Comparing the preceding upper bound $npe^{-pd_0} = npe^{-\gamma_3 d_0}$ with the lower bound $npq^{d_0} = npe^{-\gamma_1 d_0}$, we see that the only difference is in the exponents γ_1 and γ_3. Thus, we expect the loss of ER graphs to have the same exponential decay rate in the regime of constant density and in the regime of negligible loss, and expect the performances of ER graphs in these regimes relative to the respective lower bounds to be similar as well.

6. Numerical Experiments

In this section, we investigate the robustness of our theoretical results in Section 5 for the case of a complete input graph using simulated data. We first explore different parameter settings under our base model with independent random node deletions, and then extend the base model in two directions. First, we consider correlations in node deletions. Second, we generalize the notion of node deletion and take the random residual graph in a way that mimics a nonbinary demand distribution. We find that our insights from the theoretical analysis are qualitatively robust under these settings. We also obtain some additional insights for the sparse graph design problem through comparative statics across these different settings.

6.1. Main Simulation Studies

During the simulation, we examine four distinct parameter configurations by modifying the following:

1. Number of nodes (n). We consider two scenarios: $n = 150$ (a relatively large system) and $n = 30$ (a relatively small one).

2. Node deletion probability (q). Again, we consider two cases: $q = 0.7$ (a relatively large node deletion probability) and $q = 0.3$ (a relatively small one).

In terms of graph designs, we consider the following families:

1. K-clusters ($G_{n,K}$), where K ranges from 1 to n. If K does not divide n, we extend the definition of a K-cluster according to (5).

2. K-rings ($R_{n,K}$), where K ranges from 0 to $\lfloor n/3 \rfloor + 1$. If K does not divide n, we extend the definition of a K-ring according to (6).[7]

3. K-chain ($C_{n,K}$), where K ranges from 0 to $\lfloor n/2 \rfloor$.

4. Erdős–Rényi graphs ($\mathrm{ER}_{n,\alpha}$), where α ranges from zero to one. Because ER is a randomized graph design, we draw multiple samples for every fixed α and evaluate the performance for each sample.

5. Random d-regular graphs ($G_{n,d}$), where d ranges from 1 to $n-1$. A *d-random regular* graph is a random graph sampled uniformly randomly from the set of all d-regular graphs. Besides the regularity of $G_{n,d}$ (see the discussion at the end of Section 5.1), another reason for us to consider them is that with high probability, even a highly sparse random regular graph contains a perfect matching.

Fact 2 (Adapted from Frieze and Karoński 2016, Corollary 11.10). Let $G_{n,d}$ be a random d-regular graph, where $d \geq 3$ and n is even. Then, $\lim_{n\to\infty} \Pr(G_{n,d}$ has a perfect matching$) = 1$.

Although $\widetilde{G}_{n,d}$, the random residual subgraph of $G_{n,d}$, is unlikely to contain a perfect matching, Fact 2 suggests that $G_{n,d}$ may have low node loss; hence, it is a potentially good design to consider.

Because a random regular graph is a randomized graph design, we draw multiple samples for every fixed d and evaluate the performance for each sample.

We use several metrics to evaluate the performances of different graph designs. Besides the (expected) density $d(G)$ and the node loss $L(G)$ of a (realization of a) graph design G from one of the families above, we also use the metric $\mathbb{M}(G)/\mathbb{M}(G^{\mathrm{in}}) = \mathbb{M}(G)/\mathbb{M}(K_n)$ to evaluate the *relative* performance of G when compared with the input graph K_n. More specifically, $\mathbb{M}(G)/\mathbb{M}(K_n)$ captures the fraction captured by $\widetilde{G}$ of the expected size of a maximum matching of the input residual subgraph $\widetilde{K_n}$ after random node deletion. Noting that $\mathbb{M}(G_{n,1}) = 0$, where $G_{n,1}$ is the 1-cluster, or, equivalently, the empty graph, we also have

$$\frac{L(G_{n,1}) - L(G)}{L(G_{n,1}) - L(K_n)} = \frac{\mathbb{M}(K_n) - [\mathbb{M}(K_n) - \mathbb{M}(G)]}{\mathbb{M}(K_n) - 0} = \frac{\mathbb{M}(G)}{\mathbb{M}(K_n)}.$$

Thus, if we use the empty graph $G_{n,1}$ as a benchmark, then $L(G_{n,1}) - L(G)$ is the *loss reduction* of using graph G, and the metric $\mathbb{M}(G)/\mathbb{M}(K_n)$ can be equivalently interpreted as the percentage of loss reduction achieved by a graph design G when compared with the input graph K_n.

Among all the metrics described above, $L(G)$ is the most computationally intensive. We use Monte Carlo simulation to (approximately) compute $L(G)$. To reduce variance and the required number of generated samples, we couple the sample paths associated with random node deletions in the following way, for different graph designs: Suppose a collection of graphs designs $\{G_0, G_1, \ldots, G_k\}$ is given, where $G_0 = K_n$ is the complete graph. For every $t = 1, \ldots, T' := 400{,}000$, we delete each node independently with probability q. Then, we compute the maximum

matching of the residual graphs of $\{G_0, G_1, \ldots, G_k\}$ on the *same* realizations of node deletions. Then, for graph design G_i, we compute the difference of number of matched nodes between G_i and G_0. Finally, We take the average of those differences over t. It is an unbiased estimate of $L(G_i)$ and has less variance than generating random demand samples for $G_0, G_1, \ldots, G_k$ separately.

6.1.1. Summary of Findings. We find that our qualitative insights from the theoretical analysis remain robust to those parameter variations. We summarize our findings below:

1. *All graph designs we consider find relatively large matching while being fairly sparse.* For example, even when $n = 30$ and $q = 0.7$, a K-ring with density $d = 6$ can achieve a loss of around $L(R_{n,2}) = 2$. To put things into perspective, note that a 1-cluster $G_{n,1}$ (i.e., null graph) has loss $L(G_{n,1}) \approx 9$, which is also the loss reduction of the complete graph. Our graphs achieve $[L(G_{n,1}) - L(R_{n,2})]/[L(G_{n,1})] \approx 80\%$ of the loss reduction of the complete graph using $d/(n-1) \approx 20\%$ of its edges. The benefit of sparse matching is more significant when n is large and q is small. For example, when $n = 150$ and $q = 0.3$, a graph with density $d = 2$ can achieve a loss of around $L(G) = 20$, which, roughly speaking, translates to 80% of the loss reduction of the complete graph, using only 1.3% of the edges; see Figure 4 for more details.

This finding is consistent with our theoretical results that $L(G_n) = o(n)$ for all $c = \omega(1)$ when G_n belongs to the K-cluster, K-ring, K-chain, and ER graph designs, respectively. An additional insight from the simulation is that the benefit of those sparse graph designs takes effect even when n is relatively small ($n = 30$).

2. *The performances of the graph designs can be clearly divided into two groups (when $d > 1$).* The first group contains K-rings, ER graphs, K-chains, and random regular graphs. They all achieve a similar density-loss spectrum (though the random regular may perform slightly better), and all of them significantly outperform the K-cluster. This finding is best seen in the log scale, and we refer the readers to Figure 5 for more details.

Figure 4. (Color online) Matching Loss ($L(G)$) as a Function of the Graph Density (d) for the Various Graph Designs (Linear Scale)

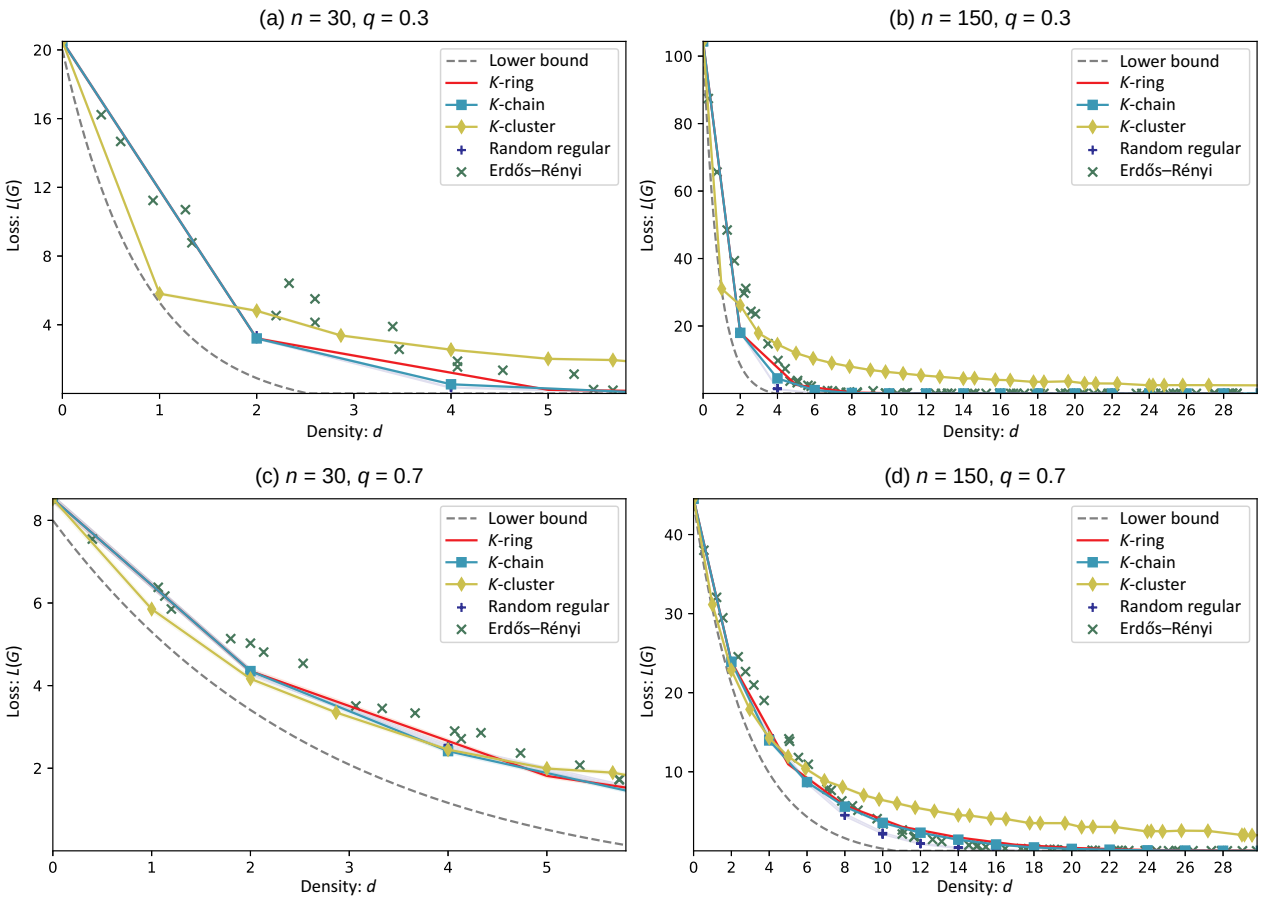
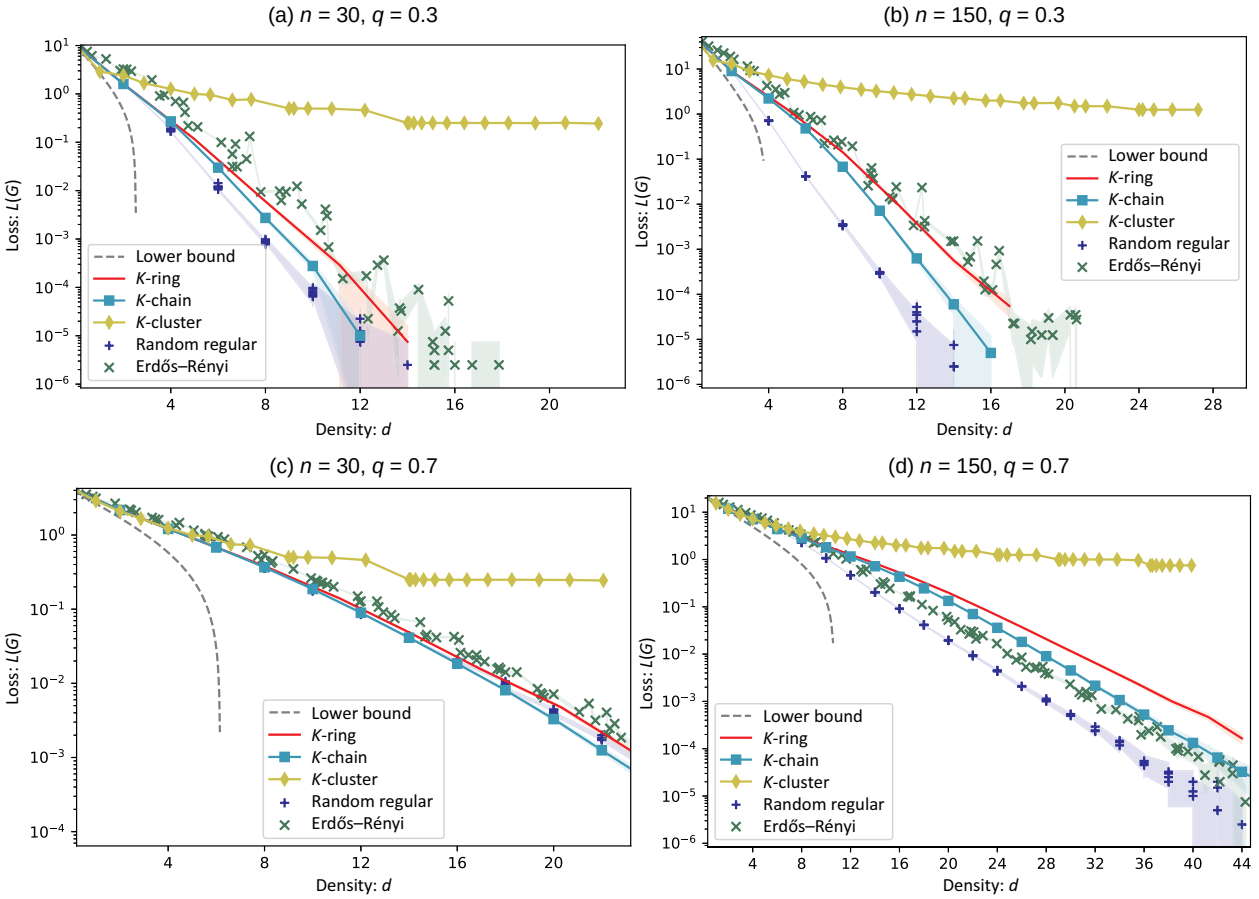


Figure 5. (Color online) Matching Loss ($L(G)$) as a Function of the Graph Density (d) for the Various Graph Designs (Log Scale)



Note. The shaded band represents the estimated 95% confidence interval.

This finding is consistent with our theoretical results that the decay rate of $L(G_n)$ is *reciprocal* for K-clusters and *exponential* for K-rings and K-chains, and we only need $d(n) = (1+\epsilon)\log n/\gamma_4$ to achieve $L(G_n) = o(1)$ for ER graphs. Although our theory does not explicitly characterize the performance of random regular graphs, their effectiveness is expected from Fact 2. What is somewhat surprising is that random regular graphs are even better than K-rings, K-chains, and ER graphs (though the difference is only noticeable when the loss $L \ll 1$).

In addition, through comparative statics across those settings, we also obtain some further insights regarding how to pick the right design:

1. *When n is larger and q is smaller, the potential benefit of a graph design is larger.* To be more precise, the potential benefit of graph design is defined by the loss reduction of the complete graph compared with the null graph, which equals $L(G_{n,1}) - L(K_n) \approx n(1-q)$. This finding can be easily seen from the closed-form expression and is verified in Figure 4.

2. *When q is smaller, the potential benefit of a graph design is better captured by sparse ones.* For example, if we compare panels (b) and (d) in Figure 4, we can see that a K-ring, K-chain, or random regular graph with density $d = 2$ can achieve a loss of $L(G) \approx 20$ both when $q = 0.3$ and when $q = 0.7$. However, that translates to roughly $(150 \times 0.7 - 20)/(150 \times 0.7) \approx 80\%$ of the potential benefit in the former case and only roughly $(150 \times 0.3 - 20)/(150 \times 0.3) \approx 55\%$ in the latter case. A similar finding can be verified by comparing panels (a) and (c) in Figure 4, or looking at the plot in log scale in Figure 5.

This finding is consistent with our results because the decay rates of the loss-density spectrum under K-clusters, K-rings, K-chains, and ER graphs are captured by $\gamma_1 = \log(1/q)$, $\gamma_2 = \log(1/q)/3$, $\gamma_3 = \log(1/q)/4$, and $\gamma_4 = 1-q$, respectively, which all decrease in q.

A takeaway from our previous comparative statics analysis is that the benefits of sparse design are most prominent when the number of nodes n is large and the probability q is small: the total benefit is too large to ignore (for, otherwise, a null graph would be a conveniently trivial option), and a sparse design is sufficient to capture most of it (for, otherwise, one has to look for a near-complete graph).

6.2. Extensions and Robustness Checks

6.2.1. Extension 1: Correlated Node Deletion. We first "stress test" our base model by considering correlation among node deletions. This is motivated by our transportation problem collaboration as demands across delivery stations are affected by common factors and are thus correlated.

Although there are many potential ways to model correlation, we choose arguably one of the simplest: the popular one-factor Gaussian copula model (e.g., Hull 2018). Under this model, demand realizations depend on two components: a common factor F that systematically affects the demands across all stations and idiosyncratic demand shocks at each station. One could interpret this common factor as seasonality, weather (e.g., temperature), promotions, etc.

In the simulation, we rerun our simulation but with a coefficient of correlation of $\rho = 0.9$. See Figure 6 for the results of our simulation. Two key findings emerge from this numerical experiment with (positively) correlated node deletion:

1. *Even sparse graphs can capture most of the benefits of graph design.* For example, let us compare panel (b) in Figure 4 (where $\rho = 0$) and panel (b) in Figure 6 (where $\rho = 0.9$): a 2-cluster (with $d = 1$) captures around 70% of the potential benefit in the former but around 90% in the latter; a 1-ring (with $d = 2$) only captures around 80% in the former but more than 90% in the latter. Furthermore, we find from Figure 6 that the density-loss spectrum tends to be L-shaped (i.e., convex) rather than linear in the log scale. In other words, correlated node deletion bends the loss-density spectrum in a way that a (sufficiently) sparse graph could perform even better and a (sufficiently) dense graph even worse. As a consequence, correlated node deletion further encourages a (highly) sparse graph.

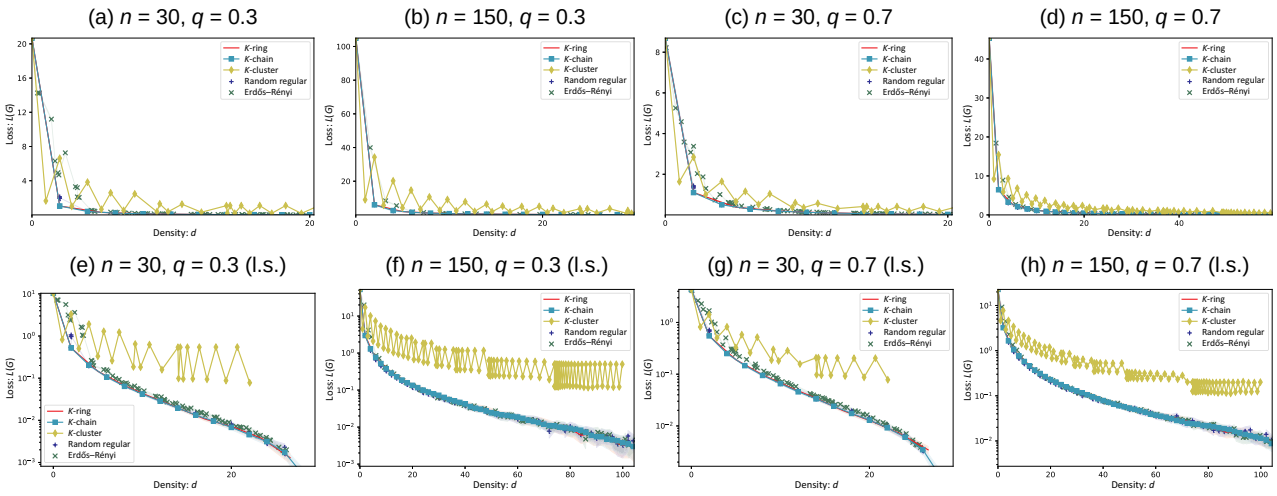
2. *The performances of K-ring, ER graphs, K-chains, and random regular graphs are nearly indistinguishable when graph density is sufficiently large.* This finding is also most clearly seen from the log-scale plot in Figure 6.

With our numerical results from the extension, we have a richer understanding of the graph design problem. Roughly speaking, our numerical findings suggest that, ceteris paribus, a simple and (highly) sparse graph such as 1-ring could perform even better when node deletions are correlated. In this sense, our original model of independent node deletion could be viewed as a "worst-case" instance for sparse graphs.

To build intuition why a sufficiently sparse graph could perform better when node deletions are correlated, let us consider an extreme case. Suppose node deletions are perfectly correlated: either all nodes are deleted or no nodes are deleted. We claim that even a 2-cluster (with $d = 1$) achieves zero loss: when all nodes are deleted, all graphs have the same performance, and thus there are no lost nodes; when no nodes are deleted, a 2-cluster supports a perfect matching, and thus there are no lost nodes either. This argument plus the numerical results motivates us to conjecture that for a sufficiently sparse (respectively, dense) graph, its loss monotonically decreases (respectively, increases) with respect to the correlation coefficient. We leave it as an open question for future research.[8]

Finally, we note that our qualitative findings send a different message from that in the traditional process flexibility literature. It is common wisdom that positively correlated demand reduces the benefit of

Figure 6. (Color online) Matching Loss ($L(G)$) as a Function of the Graph Density (d) Under Correlated Node Deletion for a Coefficient of Correlation $\rho = 0.9$



Note. Panels in the top row depict the performance in a linear scale, whereas those on the bottom row show the performance in a log scale (l.s.).

limited flexibility (e.g., see Jordan and Graves 1995). In this regard, our numerical results reveal that the sparse graph design problem is fundamentally different from the traditional ones studied in the process flexibility literature.

6.2.2. Extension 2: Generalized Stochastic Matching Based on Edge Capacity. The stochastic matching problem based on random node deletions is motivated by a stylized formulation of the middle-mile transportation management when the demand at each node (delivery station) is binary (either high or low with respect to the capacity of a truck). In what follows, we investigate numerically the effects of modifying this assumption by considering a nonbinary demand distribution at each node. In this generalized setup, each node $i \in [n]$ is assigned a nonnegative random demand D_i, where $\{D_i\}_{i\in[n]}$ are independently drawn from a common distribution F. Given a truck capacity C, the residual demand at node i is defined as follows: $\breve{D}_i := D_i - C\lfloor D_i/C \rfloor$. This represents the excess demand at node i after removing all the demand that can be fulfilled using dedicated full trucks with capacity C (see Section 7.2 for further details; note that there is indeed a nontrivial fraction of data points where the true demand exceeds the truck capacity). Thus, a truck can serve two stations if the sum of their residual demands is less than or equal to its capacity.

Figure 7 depicts the distribution of the residual demand when the original demand at a node follows a lognormal distribution with mean μ and standard deviation σ. The figure shows 24 different instances that we obtained by varying μ and the coefficient of variation $\mathrm{CV} = \sigma/\mu$. Each of the eight panels corresponds to a fixed value of $\mu \in \{0.5kC : k \in [8]\}$, and within each panel, we vary $\mathrm{CV} \in \{0.2, 0.5, 0.8\}$.

It is worth noticing that for small values of the mean demand relative to the capacity of a truck (i.e., panels in the top row of Figure 7), the distribution of the residual demand is rather sensitive to μ. On the other hand, when the mean demand is large relative to the capacity of a truck (i.e., panels in the bottom row), the distribution of the residual demand approaches a uniform distribution. We formalize this observation in the following proposition.

Proposition 1. *Let D be the random demand at a node, and let F denote its probability distribution. For a given truck capacity $C > 0$, let*

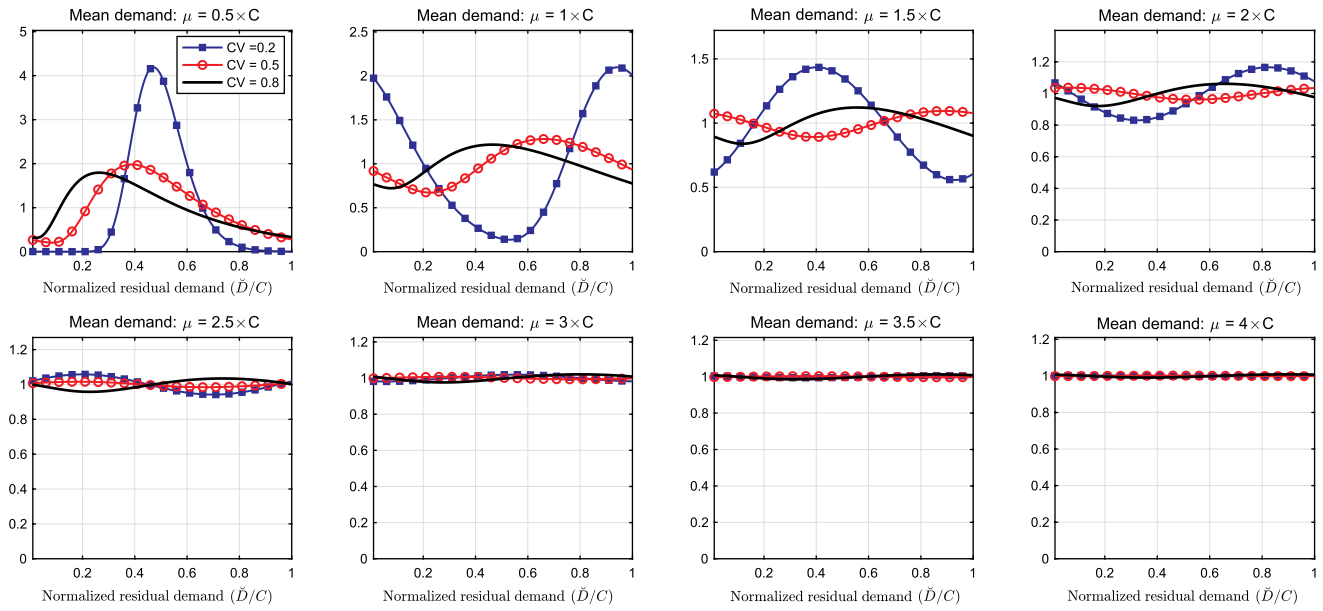
$$\breve{D} = D - C\left\lfloor \frac{D}{C} \right\rfloor \quad \text{and} \quad \overline{D} = \frac{\breve{D}}{C} = \frac{D}{C} - \left\lfloor \frac{D}{C} \right\rfloor$$

be the node's residual demand and normalized residual demand, respectively, and let $\breve{F}$ and $\overline{F}$ denote their corresponding probability distributions. Suppose F has a continuously differentiable density f that satisfies $f'(x) \le Ae^{-\alpha x}$ for constants $A > 0$ and $\alpha > 0$. Then, $\overline{F}$ converges to a Uniform distribution in $[0,1]$ as $C \downarrow 0$.

Remark 2. An alternative version of Proposition 1 can be stated in which the truck capacity C is kept fixed and demand D is let to grow large.

Let us turn now to the matching problem with general node demand. Given a graph design $G = ([n], E)$, truck capacity parameter C, and a vector of residual demands $\{\breve{D}_i\}_{i\in[n]}$, the residual graph $\widetilde{G} = ([n], \widetilde{E})$ is

Figure 7. (Color online) Numerically Computed Density of the Normalized Residual Demand $\breve{D}/C$



obtained by setting $\widetilde{E} = \{(i,j) \in E : \breve{D}_i + \breve{D}_j \le C\}$. That is, the edge $(i,j) \in E$ remains in $\widetilde{E}$ as along as the sum of the residual demands at nodes i and j can be fulfilled using a single truck. A maximum matching problem is then solved under the residual subgraph $\widetilde{G}$. The corresponding loss function $L(G)$ is defined as the expected difference between the number of matched nodes in G and the number of matched nodes in the complete graph. In Figure 8 we compute the value of $L(G)$ as a function of the density d of G when $n = 100$ for the same graph designs presented above. Also, motivated by our previous discussion and the result in Proposition 1, we consider three different distributions for the normalized residual demand $\breve{D}/C$ at a node:

1. Uniform distribution [0,1]: This case would capture the general situation in which μ is a lot higher than C or there is high uncertainty in D.

2. Lognormal distribution with $\mu = 1$ and CV = 0.2: This is a case in which the residual demand is bimodal with two peaks in the boundary around zero and one, mimicking the "binary distribution" in our base case.

3. Lognormal distribution where $\mu = 1.5$ and CV = 0.2: This is a case in which the normalized residual demand has a peak around 0.5.

By comparing the results in the base model with a binary demand in Figures 4 and 5 to those in Figure 8, we see that the performances of the various graph designs are qualitatively similar. This robustness check provides additional support.

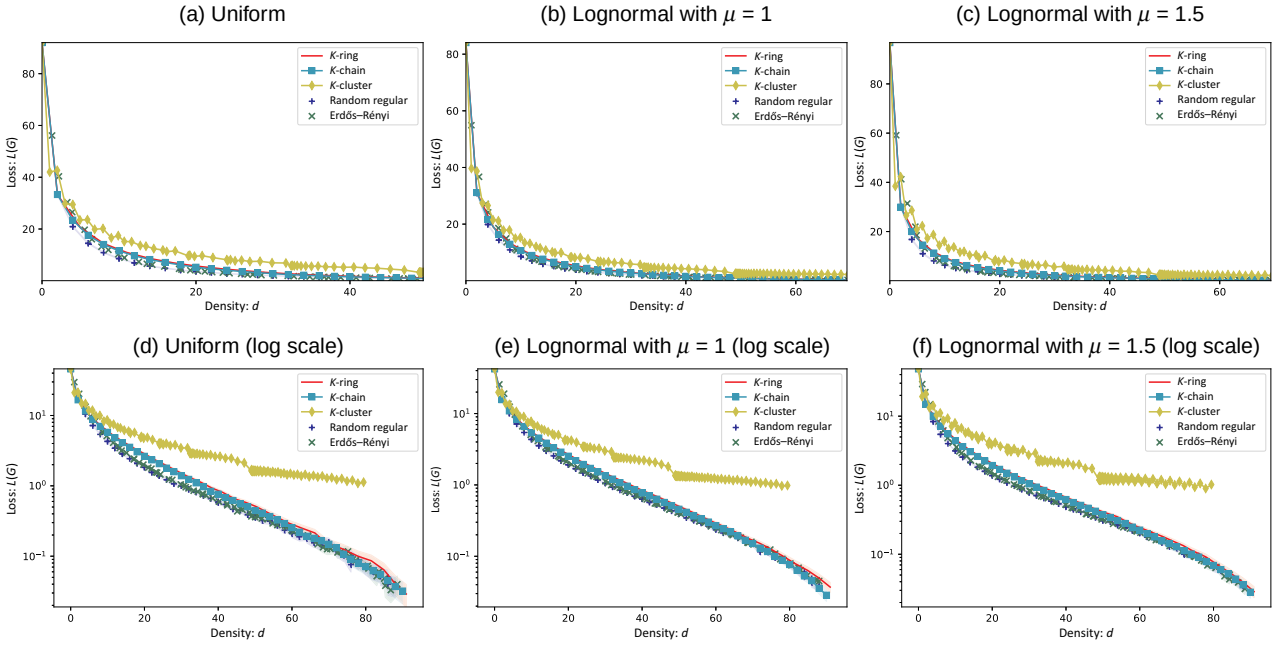
7. Case Study: An Indirect Shipping Problem for an E-Commerce Retailer

In this case study, we showcase our solution concepts by considering a transportation management problem. This problem is motivated by an industry collaboration with a leading e-commerce retailer in China (particularly, the arm in charge of its logistics and online demand fulfillment).

7.1. Background

In the past decade, China's annual express parcel volume has skyrocketed from 3.67 billion to 83 billion, a nearly 23-fold increase (Qin et al. 2022). The rapid growth of e-commerce has created a number of new operational challenges, especially in logistics. China's ratio of social logistics cost to gross domestic product is 14.2%, significantly higher than that of the U.S. logistics sector. For these reasons, China's logistics industry has been exploring various ways to improve efficiency and cut costs. One important problem is how to manage middle-mile transportation. In this problem, the e-commerce retailer manages a logistic network of transporting parcels from a depot (e.g., a distribution center) to a collection of last-mile delivery stations. At the beginning of each day, the retailer observes the volume of parcels needed to be transported to each delivery station and then orders trailer trucks to fulfill the transportation requests. Each truck has limited capacity. We consider the following two

Figure 8. (Color online) Comparison of Graph Designs for Generalized Stochastic Matching Based on Edge Capacity



Note. The labeling of curves is the same as in Figure 4.

ways that a truck can fulfill the transportation requests:

- Direct shipping ("*zhifa*"): The truck goes directly from the depot to the delivery station.
- Indirect shipping ("*chuandian*"): The truck visits multiple stations in one single trip, thus satisfying the needs of multiple stations simultaneously.

The e-commerce retailer thus specifies how many trucks to order and how each truck fulfills the transportation requests. The retailer faces several operational challenges and constraints. First, it is hard to accurately predict the demand at each delivery station in advance. Second, all the parcels must be delivered in a timely manner, and the retailer cannot deliberately delay the delivery. In the face of those operational challenges and constraints, the problem is how to fulfill the transportation needs in a *cost-effective* and *easy-to-manage* manner.

The key variable in the trade-off between cost-effectiveness (i.e., minimizing transportation cost) and ease of management (i.e., minimizing management complexity) is how much flexibility to allow for indirect shipping. On the one hand, indirect shipping helps better utilize the trucks and hence is beneficial in saving on transportation costs. To see why, note that when the demand for a delivery station is less than the truck capacity, direct shipping wastes truck capacity. Instead, if the same truck could use indirect shipping to cover a second station, its utilization goes up, potentially reducing the total number of trucks needed. On the other hand, when the delivery station demands are volatile, indirect shipping causes frequent updates of routes and is harder to manage. Routes need to be updated frequently because the actual feasibility of indirect shipping depends on the daily demand realizations of stations. Frequent route updates increase the management complexity because the controller needs to coordinate the truck routes for the whole network. Such complexity can also lead to pressure on the service level, which could in turn lead to the implicit cost increase in dollar amounts. For example, if the truck drivers are asked to operate on unfamiliar routes, more unwanted outcomes such as delivery delays could be likely, which further leads to customer dissatisfaction.

In this case study, we investigate how to leverage the methodology we developed in this paper to resolve the trade-off between transportation cost and management complexity in the transportation management problem faced by our industry collaborator.

7.2. The Indirect Shipping Problem and Stochastic Matching

Let us first consider the following indirect shipping problem. Suppose there are n delivery stations and T time periods. The demand for station i at period t is denoted by D_i^t. We make the following assumptions about these demands.

Assumption 1. *The demands satisfy the following*:

1. *Demand i.i.d. across stations and time: The demands* $\{D_i^t\}$ *are i.i.d. as a random variable* D.
2. *Binary demand:*

$$D = \begin{cases} D_L & \text{with probability } p; \\ D_H & \text{with probability } q := 1 - p. \end{cases}$$

3. *Relationship of parameters: In comparison with the truck capacity* $C > 0$, *we have* $0 < D_L < D_H \le C, 2D_L \le C$ *and* $D_L + D_H > C$.

In Assumption 1, the first condition states that the station demands are homogeneous and independent across stations and time periods. The second condition states that the demands follow a binary distribution. Finally, the last condition states that a truck can handle the direct shipping of each station (regardless of high or low demand) but can handle an indirect shipping trip only if the demands for both stations are low.

The e-commerce retailer makes two-stage decisions. First, the retailer designs a collection of *feasible* routes from the depot to one or to multiple delivery stations. (We implicitly assume that direct shipping routes are always feasible). This stage corresponds to the tactical-level decision of how much flexibility to allow for indirect shipping (to reduce management complexity). This decision is prior to all the demand realizations. Second, given the demand realizations at each time period, the retailer specifies for each truck which route to take and which delivery station(s) to serve. The route that the retailer uses needs to belong to the feasible ones. Also, the total fulfilled demands on the same route cannot exceed the capacity of the truck. This stage corresponds to the operational level of decisions of how to make the best use of indirect shipping routes (to reduce the transportation cost).

There is a one-to-one correspondence between the transportation management problem as described and our graph theoretical model. In the first stage, we can use a flexibility graph $G = (V, E)$ to represent the collection of feasible routes for indirect shipping. In this graph, every node corresponds to a distinct delivery station. Every edge corresponds to a pair of delivery stations that can be served by a single truck during a trip. The e-commerce retailer wishes to identify a *sparse* graph G, which corresponds to reducing the management complexity. In the second stage, node i remains in the graph if and only if $D_i = D_L$ (i.e., a survival node).[9] The route choice problem is thus equivalent to finding a matching in the residual graph $\widetilde{G}$: an edge $\{i, j\}$ is in the matching if and only if a truck is used to serve both node i and j. If a node is not covered by the matching, it will be served by a truck

using direct shipping. Because the number of trucks equals $2n - \mu(\widetilde{G})$, the retailer wishes to find a maximum matching in $\widetilde{G}$ to minimize the transportation cost.

In studying this transportation problem, we make a number of simplifying assumptions. First, every indirect shipping route contains no more than two delivery stations. This is for analytical tractability but is also typically the case in our data. The reason is that unloading packaging from the trucks usually takes time, and there are timeliness requirements for last-mile deliveries. In other words, it is practically uncommon to visit more than three stations in a single trip. Second, we use the *cardinality* of feasible routes as a surrogate for the management complexity for reasons we stated before. Third, the transportation cost is proportional to the number of trucks ordered. This is a good approximation when there is a homogeneous fleet of trucks and the "fixed" cost of ordering a truck is significantly high compared with the "variable" cost of visiting an additional station or traveling a longer distance.

7.3. Data Description and Analysis

In our case study, we build on the trip-level transportation data of our industrial partner for a single category, $n = 77$ stations, and $T = 196$ days from December 2018 to July 2019. More specifically, our data set contains the following information:

- the locations of the logistic facilities, including the depot (i.e., warehouse) and last-mile delivery stations[10] (we also use the route (i.e., ordered sequence of facilities) of each trip episode);
- the cost of every trip, as well as the cost-relevant factors such as truck type (e.g., 4.2 meters versus 7.6 meters in length), origin–destination distance, and route, among others; and
- the realized daily demand of every station, which can thus be inferred from the transportation data and which we denote by $\{\hat{D}_i^t : i \in [n]; t = 1, \ldots, T\}$.[11]

In our case study, we focus on a demand fulfillment network in the urban area of one of the largest cities in China. We focus on this city because it is of sufficient business importance and also possesses a couple of salient features that are well approximated by our model. First, all of the delivery stations in this region are served by a single warehouse. Second, the stations are relatively close to each other, making indirect shipping physically feasible. Third, the current operational data suggest that a trip containing more than three stations is uncommon; see Table 3. This supports our model premises to focus on indirect shipping routes containing up to two stations. We also provide an illustration of the locations of the nodes with selected graph designs; see Figure 9.

Table 3. Summary of Trip Types

# intermediate steps	Count	Percentage (%)
0 (direct shipping)	6,615	43.6
1	7,876	51.9
≥2	687	4.5
Total	15,178	100

Using the cost information, we are able to see that the real cost structure is well approximated by the one in our model. More specifically, the transportation cost of trip i can be written as

$$c_{total}(i) = c_{base}(i) + c_{unit} \times k(i). \tag{9}$$

In the expression above, the quantity $c_{base}(i)$ represents the "base" cost of trip i. The quantity $k(i)$ represents the number of intermediate delivery stations in trip i (a direct shipping route has $k = 0$). Finally, c_{unit} represents the unit cost of visiting each additional station, which is fixed to be $c_{unit} = 30$ in our data.

Although the base cost c_{base} varies from trip to trip, we find that it is approximately a (large) constant after controlling for the truck type. More specifically, we find that the following approximation works well in the data:

$$c_{base}(i) \approx \begin{cases} 420 & \text{if truck size} = 7.6 \text{ m}, \\ 320 & \text{if truck size} = 4.2 \text{ m}. \end{cases}$$

In fact, the cost residual unexplained by the constants above, which depends on other cost factors such as the origin–destination distance, is surprisingly small; it has a mean of -0.67 with a standard deviation of 27.68. In our case study, we adopt a homogeneous fleet of trucks with an estimated cost of $\hat{c}_{base} = 360$. Because it is much larger than the unit cost $c_{unit} = 30$, we conclude that it is reasonable to approximate the transportation cost by the number of trucks dispatched.

7.4. Model Calibration and Performance Evaluation

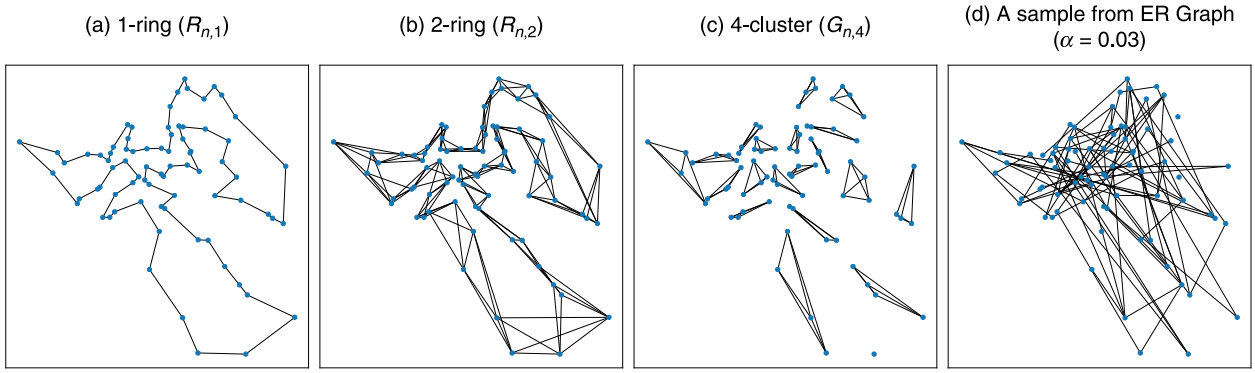
7.4.1. Model Calibration. Some of our problem parameters can be directly observed or easily inferred from the data, namely,

- the number of nodes $n = 77$;
- the capacity $C = 1,600$, which is inferred from the load and utilization rate data;
- the fixed cost $\hat{c}_{base} = 360$, which is inferred from the cost and trip data.

To calibrate the demand model, we take two different approaches, as we explain below.

The Binary Model–Based Approach. We calibrate the best-fitting parameter values (p, q, D_H, D_L) for the demand model under Assumption 1. More specifically, we use the Gaussian mixture model to cluster the pooled

Figure 9. (Color online) Selected Graph Designs



Notes. Each node represents a delivery station. The location of the every node represents (the masked version of) the physical location of the corresponding station. Each edge represents a feasible indirect shipping route in the graph design. The indexing of the nodes is determined by solving a traveling salesman problem to visit all nodes from the warehouse.

demand observations into two segments. The parametric values we obtain are $p = 0.6$, $q = 0.4$, $D_H = 1,190$, and $D_L = 456$. We can verify that the parameter values satisfy the conditions in Assumption 1.

The Fully Empirical Approach. We fully relax Assumption 1 and use a retrospective (backward) analysis of the graph performances. That is, we do not calibrate any demand model. Instead, for any fixed graph, we look at the historical demand realizations and calculate how many trucks are needed in hindsight. We refer the readers to the online appendix, Section EC.1.3, for more technical details.

7.4.2. Performance Evaluation and Results. We consider the same family of graph designs as we did in Section 6; namely, we consider K-clusters, K-rings, K-chains, ER graphs, and random regular graphs. As in Section 6, we consider two dimensions of the performance metric of a given graph G. The first dimension is *management complexity related*: we use the fraction of edges of G compared with the complete graph. Mathematically, this is equal to $d/(n-1)$. The second dimension is *transportation cost related*: we use the additional number of trucks compared with the complete graph. Mathematically, this is equal to $L(G)/2$.

We take both the binary model–based and fully empirical approaches to comparing the graph designs. The results are summarized in Figure 10. We find that under the binary model–based approach, the simulated results are consistent with our theoretical predictions: On the one hand, all of the considered graph designs incur a small number of additional trucks when the graph designs are relatively sparse. On the other hand, the K-rings, K-chains, random regular graphs, and ER graphs are all systematically more effective than the K-cluster graphs. Within those four families, the random regular graphs are the most effective (although not by a large margin).

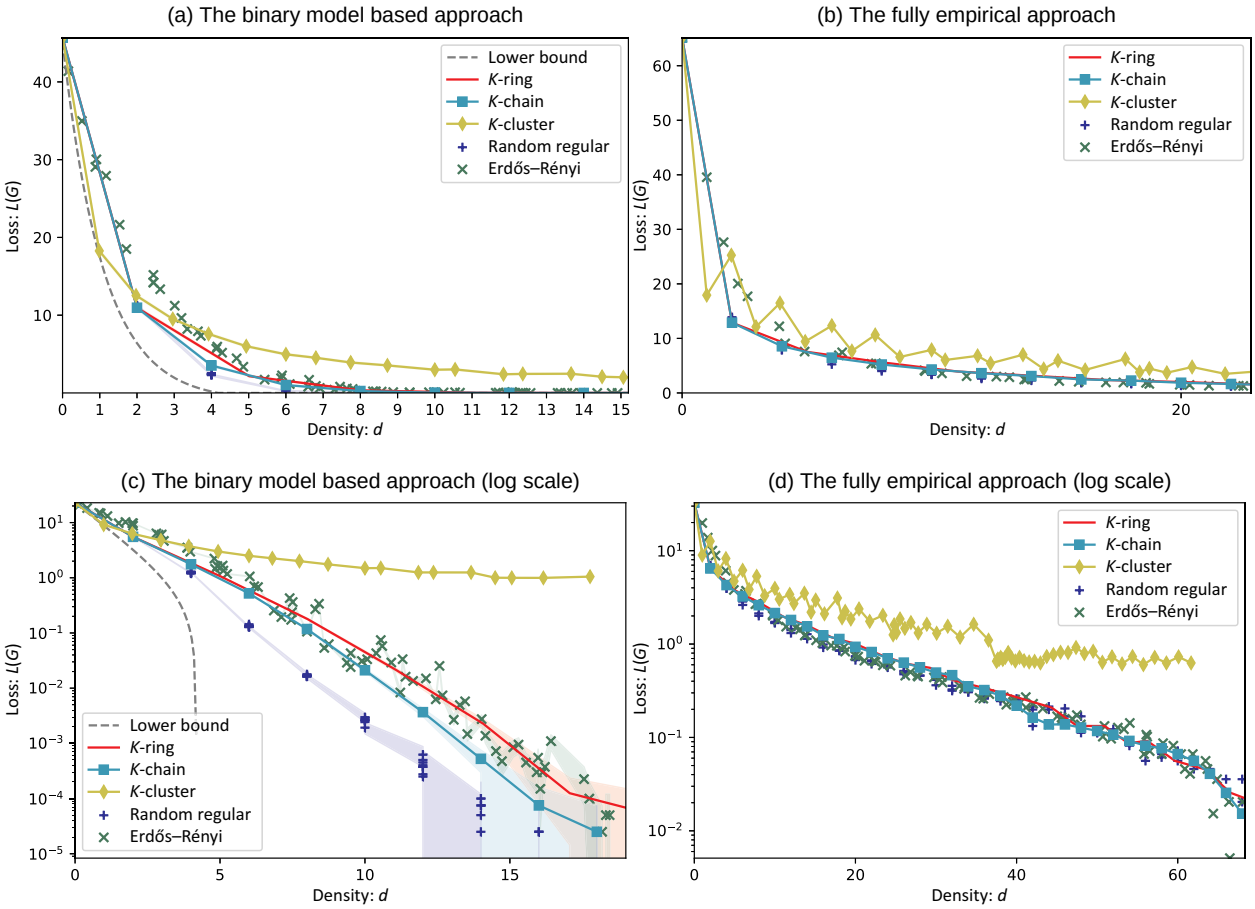
Under the fully empirical approach, the results remain qualitatively similar. We also find the quantitative differences interesting: the fully empirical approach leads to results somewhat similar to those of our earlier simulation where node deletions are correlated. More specifically, comparing the same graph(s) under the empirical approach versus the binary model–based one,

- if the graph is highly sparse (e.g., 2-cluster and 1-ring), it captures more benefits of the complete graph;
- if the graph is sufficiently dense (e.g., the fraction of edges succeeds 20%), it captures less benefit of the complete graph;
- the performances of K-rings, K-chains, ER graphs, and random regular graphs seem closer.

Inspired by our earlier simulation studies, we conjecture that the quantitative differences are at least partially driven by demand correlation across stations.

7.4.3. Counterfactual Analysis. To better understand the benefit of our methodology, we also perform a counterfactual analysis. More precisely, we single out a few graph designs and compare them to the status quo operations. These designs are $R_{n,1}$, the 1-ring; $R_{n,2}$, the 2-ring; and K_n, the complete graph. It is also possible to consider other designs, such as $C_{n,1}$ or $C_{n,2}$, the 1-chain and 2-chain, respectively. However, because $C_{n,1}$ completely coincides with $R_{n,1}$ and $C_{n,2}$ is essentially "sandwiched" between $R_{n,1}$ and $R_{n,2}$ (cf. Lemma 2), the performances of $C_{n,1}$ and $C_{n,2}$ can be bounded by those of $R_{n,1}$ and $R_{n,2}$. As a result, we only provide discussion of $R_{n,1}$, $R_{n,2}$, and K_n here, with the understanding that any qualitative conclusions that we draw for $R_{n,1}$ and $R_{n,2}$ hold for $C_{n,1}$ and $C_{n,2}$ as well.

Figure 10. (Color online) Comparison of Graph Designs for the Case Study



Notes. The number of nodes is $n = 77$. We take both the binary model–based and fully empirical approaches to making the comparison. In the former approach, the model parameters are $p = 0.6, q = 0.4, C = 1,600, D_H = 1,190$, and $D_L = 456$, which are all fitted from data. The labeling of curves is the same as in Figure 4.

For thoroughness, we consider a collection of auxiliary metrics that are more concrete and closely related to the day-to-day operations. Let us list them below:

- The *transportation cost–related* metrics include the following:
  1. The average number of trucks per day to serve the region's demand, denoted by V.
  2. The average utilization rate of trucks, denoted by ρ. It is defined by the ratio between the total number of parcels delivered and the total truck transportation capacity.
  3. The projected monetary daily transportation cost, denoted by $\hat{c}$. The projected cost for trip i is calculated by $\hat{c}(i) = \hat{c}_{base} + k(i) \times c_{unit} = 360 + 30k(i)$; see Equation (9) and the discussion below for more details. Here, we further take the variable cost of indirect shipping into account.
- The *management complexity–related* metrics include the following:
  1. The route variety, denoted by RV. It is defined as the maximum number of routes taking a station as the first stop.
  2. The total number of (directed) routes, denoted by R_d. That is, this metric treats the routes $A \to B$ and $B \to A$ as different ones.
  3. The total number of (undirected) routes, denoted by R_u. That is, this metric treats the routes $A \to B$ and $B \to A$ as the same one. We include this metric because in the status quo, we can observe trucks following the same (undirected) routes but in different directions. This metric eliminates the directional effect in the status quo data.

Note that those metrics depend not only on the graph structure but also on the demand realizations and dispatching policy. That is, even with the same transportation network but with different demand realizations and dispatching policies, we can get different values of those metrics. To further illustrate

Table 4. Performance Comparison: Status Quo vs. Selected Graph Designs

	Transportation cost related			Management cost related		
Graph design	V	ρ	$\hat{c}$	RV	R_d	R_u
Status quo	77.4	0.47	30.6K	8	328	250
$R_{n,1}$ (1-ring)	54.0	0.67	20.2K	2	154	154
$R_{n,2}$ (2-ring)	51.3	0.71	19.3K	4	269	269
K_n (complete graph)	47.6	0.76	18.1K	49	1,798	1,798

how the metrics above are calculated from the trip record, we walk through a simple example; see Example EC.1 in the online appendix, Section EC.1.4.

In our counterfactual analysis, we use the fully empirical approach to compare with the status quo. That is, we first calculate the values of the performance metrics based on the status quo trip records. Then, we repeat the process for the *same* demand realizations but under *different* transportation networks and the dispatching policy to leverage the maximum matching.

We find that $R_{n,1}$, the simplest K-ring design, offers a particularly good performance. It reduces by more than 30% the truck usage and total transportation cost while being simpler to manage (using metrics such as RV, R_d, and R_u). We can enjoy better cost reduction by adopting the $R_{n,2}$ design by maintaining roughly the same level of management complexity. The marginal cost reduction decreases when we consider more complicated graph designs. We have also included more details of our findings in Table 4.

7.4.4. Final Recommendation and Discussion. Note that, on the one hand, $R_{n,1}$ can already save much cost. On the other hand, the benefits of more dense graphs beyond $R_{n,2}$ are quite limited. Therefore, our preliminary graph design recommendation from this case study is to implement either a sparse K-ring design, $R_{n,1}$ or $R_{n,2}$ (or a sparse K-chain design, $C_{n,1}$ or $C_{n,2}$; see earlier remarks on our choices of graph designs).

From a managerial point of view, our finding in the case study echoes the literature on process flexibility. In the context of the indirect shipping problem with uncertain demands, we find that limiting to a small fraction of indirect shipping routes imposes almost no increment in the transportation cost despite its apparent benefit in reducing the management complexity. In this sense, we find that "a little flexibility is all you need."

What makes this case study somewhat surprising is *how little* flexibility is needed; that is, we find that even highly sparse and simple graphs such as $R_{n,1}$ or $R_{n,2}$ can be highly effective. This phenomenon can be much explained by our theoretical results: the loss decays *exponentially* fast with the density, and the implied node deletion probability in this case study is $q = 0.4$, a relatively low value. Moreover, we find that the performance results under the fully empirical approach are somewhat similar to those of our earlier simulation studies for correlated node deletion. This observation motivates us to conjecture that demand correlation also contributes to the effectiveness of highly sparse graphs such as $R_{n,1}$ and $R_{n,2}$.

Of course, looking beyond the case study, we will not anticipate $R_{n,1}$ or $R_{n,2}$ to be the *universal* solution for all environments. For example, we anticipate that in the future, the e-commerce retailer's business will grow, and so will the number of delivery stations in the same region. Our theory provides guidance for such cases: A denser graph with more indirect shipping routes will be needed. Therefore, it may be preferable to move down along the density-cost efficient frontier and pick a graph such as $R_{n,k}(k \geq 3)$.

Last but not least, we want to point out that in practice, K-rings (as well as K-chains and K-clusters) are far more attractive than ER graphs, because a deterministic graph design is more desirable than a randomized one. In addition, distance between nodes is an important factor to consider and K-rings (as well as K-chains and K-clusters) implicitly take this factor into account, whereas ER graphs may include edges linking two nodes far away from each other.

8. Concluding Remarks

In this paper, we study the problem of designing a sparse graph that can support a large matching when nodes are randomly removed. This class of problems is motivated by the need to balance the trade-off between costs and routing complexity in a middle-mile logistics network. To tackle this random matching problem, we study four graph families, namely, K-clusters, K-rings, K-chains, and Erdős–Rényi graphs, and provide a theoretical analysis of their performance and show that their matching loss can be close to that of a complete graph. To complement our theoretical study, we evaluate the empirical performance of these graph designs using real data from our industry partner. Our results show that adding flexibility to the transportation network can significantly reduce operating costs. Overall, our study highlights the importance of carefully designing sparse graphs that can support large matchings under node deletion.

Our findings have practical implications for middle-mile transportation and can be used to inform the design of efficient and cost-effective transportation networks.

There are several potential directions for further research. One of them relates to the binary demand assumption that we have imposed. Relaxing this assumption and allowing for general (possibly non-identical) demand distributions at the delivery stations would bring our random matching formulation closer to the logistics network application. In Section 6.2, we take a first step in this direction by numerically studying the performance of our proposed graph designs when the demand distribution is nonbinary. Our computational experiments reveal that the performance of K-clusters, K-rings, K-chains, and Erdős–Rényi graphs under a nonbinary demand model remains qualitatively similar to that observed under a binary demand model.

Another interesting direction for extending our work is related to the number of stations a single truck can visit. In our paper, we assumed that a truck can visit at most two delivery stations, an assumption that aligns with the current operations of our industry partner (less than 5% of trips involve three or more stations). However, in the highly competitive logistics transportation industry, even a small percentage reduction in operating costs can significantly impact a company's bottom line. We anticipate that extending our model to accommodate multiple truck stops would result in a substantially more challenging problem. For instance, if trucks could serve any number of delivery stations, determining the minimum number of trucks needed to serve all delivery stations would be equivalent to a bin packing problem, which is strongly NP-hard. In contrast, our constrained model, which allows a maximum of two stops per truck, enables the identification of the minimum number of trucks using a polynomial-time algorithm.

A third direction along which our work can be extended involves allowing for a general input graph rather than a complete one, as we have assumed in our paper. For example, a somewhat trivial extension is to consider input graphs that are complete minus $O(1)$ number of edges; in this case, all our theoretical results in Section 5 essentially carry over verbatim. A more important and challenging extension is to provide a precise characterization of the density-loss trade-off for a broader, practically relevant class of input graphs. We anticipate that a key challenge in handling a general input graph would be the derivation of tight upper and lower bounds on the number of matchings in the residual subgraph.

A final direction for extending our work is to adopt a more abstract approach to the problem of graph design, one that is closer in spirit to a theoretical computer science approach. For example, one could exploit the so-called edge-degree-constrained subgraph construction (see Assadi and Bernstein 2019) to produces a subgraph that contains a large matching of the original graph. The following result illustrates this idea.

Theorem 5. *There exists a deterministic polynomial-time algorithm, which, given a graph $G^{\text{in}} = (V, E^{\text{in}})$ and parameter $\varepsilon < 1/4$, computes a subgraph G of G^{in} with maximum degree $O(\log \frac{1}{\varepsilon p}/(\varepsilon^2 p))$, such that*

$$\mathbb{M}(G^{\text{in}}) \le \left(\frac{3}{2} + \varepsilon\right)\mathbb{M}(G). \tag{10}$$

Acknowledgments

The authors express their appreciation for the support from colleagues affiliated with Alibaba Cainiao, including Jianya Ding, Fan Dong, Lei Shen, Yinghui Xu, and Lijun Zhu.

Endnotes

[1] Here, "mod" represents the modulo operation for (possibly negative) integers. For example, $1 \bmod 10 = 1$, $11 \bmod 10 = 1$, and $(-1) \bmod 10 = 9$. We also added a nominal "0" chain, which represents a null graph.

[2] To put things into perspective, in our motivating middle-mile delivery example, it is not uncommon that a distribution center can cover hundreds of last-mile delivery stations (corresponding to the setting with $n \approx 100$).

[3] This notation means that $d - \log n \to +\infty$ as n grows; see Section 2.

[4] Although a K-cluster is can be well defined for a generic K, we always assume that n is a multiple of K in our analysis for simplicity of handling the rounding issues. This is an immaterial convention when n is large, which will be confirmed by our numerical studies. Under this convention, the feasible densities for K-clusters are $\{K-1 : K|n\}$.

[5] Similar to K-clusters, we always assume that n is a multiple of K in our analysis. Under this convention, the feasible densities for K-rings is $\{3K-1 : K|n\}$. We will relax this assumption in our numerical studies to confirm that it is an immaterial convention.

[6] Whereas the definition of a K-chain allows n and K to be arbitrary, Corollary 1 requires n to be divisible by $\lfloor K/2 \rfloor$. This is an immaterial restriction when n is large, as is the case with K-clusters and K-rings.

[7] We also added a nominal "0" ring, which represents a null graph.

[8] It is also worth noting that the performances of K-clusters become nonmonotone in K when node deletions are highly correlated. This is because, in this case, the parity of the cluster size becomes overwhelmingly important. To illustrate, suppose again node deletions are perfectly correlated. Now think of the case where no nodes are deleted. Whereas a 2-cluster supports a perfect matching, a 3-cluster leaves one-third of the nodes unmatched because one node is unmatched for each cluster.

[9] Note that we suppress the time superscript in D_i^t here. By doing so, we are implicitly reducing the T-period demand process into a one-period one. It is without loss of generality because demands are i.i.d. across time, which further implies that the performance of G with respect to T periods is proportional to that with respect to one period only.

[10] To protect the privacy of our partner, the geographic coordinate locations of the facilities are masked with approximations and rigid transformation.

[11] Here both the capacity and demands are measured in the number of parcels. We find it a reasonable measure for the chosen category because the parcel shapes and sizes are relatively standardized and uniform for the category we consider.

References

Afèche P, Caldentey R, Gupta V (2022) On the optimal design of a bipartite matching queueing system. *Oper. Res.* 70(1):363–401.

Allon G, Van Mieghem JA (2010) Global dual sourcing: Tailored base-surge allocation to near-and offshore production. *Management Sci.* 56(1):110–124.

Angluin D, Valiant L (1979) Fast probabilistic algorithms for Hamiltonian paths and matchings. *J. Comput. System Sci.* 18(2):155–193.

Aronson J, Frieze A, Pittel BG (1998) Maximum matchings in sparse random graphs: Karp–Sipser revisited. *Random Structures Algorithms* 12(2):111–177.

Asadpour A, Wang X, Zhang J (2020) Online resource allocation with limited flexibility. *Management Sci.* 66(2):642–666.

Ashlagi I, Gamarnik D, Rees MA, Roth AE (2012) The need for (long) chains in kidney exchange. Working Paper No. 18202, National Bureau of Economic Research, Cambridge, MA.

Assadi S, Bernstein A (2019) Toward a unified theory of sparsification for matching problems. *Proc. Second Sympos. Simplicity Algorithms* (Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Wadern, Germany), 11:1–11:20.

Assadi S, Khanna S, Li Y (2016) The stochastic matching problem with (very) few queries. *Proc. 2016 ACM Conf. Econom. Comput.* (Association for Computing Machinery, New York), 43–60.

Bassamboo A, Randhawa RS, Van Mieghem JA (2010) Optimal flexibility configurations in newsvendor networks: Going beyond chaining and pairing. *Management Sci.* 56(8):1285–1303.

Bassamboo A, Randhawa RS, Van Mieghem JA (2012) A little flexibility is all you need: On the asymptotic value of flexible capacity in parallel queuing systems. *Oper. Res.* 60(6):1423–1435.

Behnezhad S, Derakhshan M, Hajiaghayi MT (2020) Stochastic matching with few queries: (1-ϵ) approximation. *Proc. 52nd Annual ACM SIGACT Sympos. Theory Comput.* (Association for Computing Machinery, New York), 1111–1124.

Behnezhad S, Farhadi A, Hajiaghayi MT, Reyhani N (2019a) Stochastic matching with few queries: New algorithms and tools. *Proc. 30th Annual ACM-SIAM Sympos. Discrete Algorithms* (Society for Industrial and Applied Mathematics, Philadelphia), 2855–2874.

Behnezhad S, Derakhshan M, Farhadi A, Hajiaghayi M, Reyhani N (2019b) Stochastic matching on uniformly sparse graphs. Fotakis D, Markakis E, eds. *Algorithmic Game Theory* (Springer, Cham, Switzerland), 357–373.

Berge C (1958) Sur le couplage maximum d'un graphe. *Comptes Rendus de l'Académie des Sciences* 247(3):258–259.

Blum N (1990) A new approach to maximum matching in general graphs. Paterson MS, ed. *Automata, Languages and Programming*. Lecture Notes in Computer Science, vol. 443 (Springer, Berlin), 586–597.

Blum A, Dickerson JP, Haghtalab N, Procaccia AD, Sandholm T, Sharma A (2020) Ignorance is almost bliss: Near-optimal stochastic matching with few queries. *Oper. Res.* 68(1):16–34.

Bollobás B, Béla B (2001) *Random Graphs*, 2nd ed. (Cambridge University Press, Cambridge, UK).

Chebolu P, Frieze A, Melsted P (2010) Finding a maximum matching in a sparse random graph in o(n) expected time. *J. ACM* 57(4):1–27.

Chen S, Song JS, Wei Y (2020) Data-driven scalable e-commerce transportation network design with unknown flow response. Preprint, submitted May 29, https://dx.doi.org/10.2139/ssrn.3590865.

Chen X, Zhang J, Zhou Y (2015) Optimal sparse designs for process flexibility via probabilistic expanders. *Oper. Res.* 63(5):1159–1176.

Chou MC, Teo CP, Zheng H (2008) Process flexibility: Design, evaluation, and applications. *Flexible Services Manufacturing J.* 20(1–2):59–94.

Deng T, Shen ZJM (2013) Process flexibility design in unbalanced networks. *Manufacturing Service Oper. Management* 15(1):24–32.

DeValve L, Wei Y, Wu D, Yuan R (2023) Understanding the value of fulfillment flexibility in an online retailing environment. *Manufacturing Service Oper. Management* 25(2):391–408.

Dickerson JP, Procaccia AD, Sandholm T (2012) Optimizing kidney exchange with transplant chains: Theory and reality. *Proc. 11th Internat. Conf. Autonomous Agents Multiagent Systems*, vol. 2 (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC), 711–718.

Ding Y, Ge D, He S, Ryan CT (2018) A nonasymptotic approach to analyzing kidney exchange graphs. *Oper. Res.* 66(4):918–935.

Edmonds J (1965) Paths, trees and flowers. *Canadian J. Math.* 17:449–467.

Erdős P, Rényi A (1966) On the existence of a factor of degree one of a connected random graph. *Acta Math. Academiae Scientiarum Hungarica* 17(3–4):359–368.

Even S, Kariv O (1975) An O($n^{2.5}$) algorithm for maximum matching in general graphs. *Proc. IEEE 16th Annual Sympos. Foundations Comput. Sci.* (Institute of Electrical and Electronics Engineers, Piscataway, NJ), 100–112.

Frieze AM (1986) On large matchings and cycles in sparse random graphs. *Discrete Math.* 59(3):243–256.

Frieze A, Karoński M (2016) *Introduction to Random Graphs* (Cambridge University Press, Cambridge, UK).

Fujii M, Kasami T, Ninomiya K (1969) Optimal sequencing of two equivalent processors. *SIAM J. Appl. Math.* 17(4):784–789.

Galil Z (1986) Efficient algorithms for finding maximum matching in graphs. *ACM Comput. Surveys* 18(1):23–38.

Graves SC, Tomlin BT (2003) Process flexibility in supply chains. *Management Sci.* 49(7):907–919.

Hall P (1935) On representatives of subsets. *J. London Math. Soc.* s1-10(1):26–30.

Hopcroft J, Karp R (1973) An O($n^{2.5}$) algorithm for maximum matching in bipartite graphs. *SIAM J. Comput.* 2(4):225–231.

Hull JC (2018) *Risk Management and Financial Institutions*, 5th ed. (John Wiley & Sons, Hoboken, NJ).

Iravani SMR, Kolfal B, Van Oyen MP (2007) Call-center labor cross-training: Its a small world after all. *Management Sci.* 53(7):1102–1112.

Jordan WC, Graves SC (1995) Principles on the benefits of manufacturing process flexibility. *Management Sci.* 41(4):577–594.

Kameda T, Munro I (1974) An O($|V||E|$) algorithm for maximum matching of graphs. *Computing* 12:91–98.

Karp RM, Sipser M (1981) Maximum matching in sparse random graphs. *Proc. IEEE 22nd Annual Sympos. Foundations Comput. Sci.* (Institute of Electrical and Electronics Engineers, Piscataway, NJ), 364–375.

Ledvina K, Qin H, Simchi-Levi D, Wei Y (2022) A new approach for vehicle routing with stochastic demand: Combining route assignment with process flexibility. *Oper. Res.* 70(5):2655–2673.

Lyu G, Cheung WC, Chou MC, Teo CP, Zheng Z, Zhong Y (2019) Capacity allocation in flexible production networks: Theory and applications. *Management Sci.* 65(11):5091–5109.

May JW (2015) Cheminformatics for genome-scale metabolic reconstructions. Unpublished PhD thesis, University of Cambridge, Cambridge, UK.

Micali S, Vazirani V (1980) An O($\sqrt{V}E$) algorithm for finding maximum matching in general graphs. *Proc. 21st IEEE Annual Sympos. Foundations Comput. Sci.* (IEEE Computer Society Press, Los Alamitos, CA), 17–27.

Naughton K, Boyle M (2019) Walmart targets automated "middle-mile" delivery to cut shipping costs. *Transport Topics* (June 19), https://www.ttnews.com/articles/walmart-targets-automated-middle-mile-delivery-cut-shipping-costs.

Qin H, Xiao J, Ge D, Xin L, Gao J, He S, Hu H, Carlsson JG (2022) JD.com: Operations research algorithms drive intelligent warehouse robots to work. *INFORMS J. Appl. Anal.* 52(1):42–55.

Roth AE, Sönmez T, Ünver MU (2005) Pairwise kidney exchange. *J. Econom. Theory* 125(2):151–188.

Shi C, Wei Y, Zhong Y (2019) Process flexibility for multi-period production systems. *Oper. Res.* 67(5):1300–1320.

Simchi-Levi D, Wei Y (2012) Understanding the performance of the long chain and sparse designs in process flexibility. *Oper. Res.* 60(5):1125–1141.

Tsitsiklis JN, Xu K (2017) Flexible queueing architectures. *Oper. Res.* 65(5):1398–1413.

Tutte WT (1947) The factorization of linear graphs. *J. London Math. Soc.* 1(2):107–111.

Wang X, Zhang J (2015) Process flexibility: A distribution-free bound on the performance of *k*-chain. *Oper. Res.* 63(3):555–571.

Wolsey LA (1998) *Integer Programming* (John Wiley & Sons, Hoboken, NJ).

Xin L, Goldberg DA (2018) Asymptotic optimality of tailored base-surge policies in dual-sourcing inventory systems. *Management Sci.* 64(1):437–452.

Xu Z, Zhang H, Zhang J, Zhang RQ (2020) Online demand fulfillment under limited flexibility. *Management Sci.* 66(10): 4667–4685.