Masking Primal and Dual Models for
Data Privacy in Network Revenue Management

Utku Karaca
Erasmus University Rotterdam, 3000 DR, Rotterdam P.O. Box 1738, The Netherlands

Ş. İlker Birbil
University of Amsterdam, 11018 TV, Amsterdam P.O. Box 15953, The Netherlands

Nurşen Aydın
University of Warwick, Coventry CV4 7AL, United Kingdom

Gizem Mullaoğlu *
Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands

Abstract: We study a collaborative revenue management problem where multiple decentralized parties agree to share some of their capacities. This collaboration is performed by constructing a large mathematical programming model available to all parties. The parties then use the solution of this model in their own capacity control systems. In this setting, however, the major concern for the parties is the privacy of their input data along with their individual optimal solutions. We first reformulate a general linear programming model that can be used for a wide-range of network revenue management problems. Then, we address the data-privacy concern of the reformulated model and propose an approach based on solving an equivalent data-private model constructed with input masking via random transformations. Our main result shows that after solving the data-private model, each party can safely access only its own optimal capacity control decisions. We also discuss the security of the transformed problem in the considered multi-party setting. We conduct simulation experiments to support our results and evaluate the computational efficiency of the proposed data-private model. Our work provides an analytical approach and insights on how to manage shared resources in a network problem while ensuring data privacy. Constructing and solving the collaborative network problem requires information exchange between parties which may not be possible in practice. Including data-privacy in decentralized collaborative network revenue management problems with capacity sharing is new to the literature and relevant to practice.

Keywords: Data privacy; network revenue management; collaboration; resource sharing

1. Introduction. Forming alliances is an important business strategy for a firm to streamline its costs to remain competitive. Alliances can be considered as the collaboration among several parties to conduct various activities such as allocating resources, sharing information, and providing complementary services. These partnerships can also be observed among competitors, like several firms joining their professional assets to manage a supply chain network (Granot and Sošić, 2005; He and Yin, 2015). In power networks, independent power suppliers operate on a distributed system where they work together to balance the power demand and supply (Ghaderyan et al., 2021). Recently, logistics companies have started to collaborate by sharing empty vehicle capacities to overcome the problem of excess capacity in freight transportation (Speranza, 2018). One of the important advantages of collaboration is the increase in economies of scale. In airline revenue management, the carriers sign an alliance contract, called codeshare agreements, to share their flight capacities, and provide joint services. The companies can offer more products through joint services, leading to greater revenue opportunities in the long term (Topaloglu, 2012). A greater customer value can be achieved with increased flexibility in the provided services. In addition to economic benefits, the companies can also improve their reputation by improving the sustainability aspects of their operations (Gansterer and Hartl, 2016).

Coordination of partnerships or collaborations involves a series of challenges. Generally, in an alliance, parties pool some of their resources and share the necessary information for the collaborative decision making process. Besides coordinating shared resources, they also manage their individual local resources. Therefore, these parties, though often working towards a common goal, can be competitors and may be unwilling or unable to disclose complete information about their operations to protect their own interests. In addition, legal frameworks like the antitrust law force companies to take extra measures to protect their data (Gerlach et al., 2013; Wright, 2014; Albrecht and Stadtler, 2015). Depending on the industry, this sensitive information may involve demand forecasts, selling prices, operational costs and available resources. For instance, in air-cargo transportation, airlines and freight forwarders collaborate to sell the flight capacity. In this partnership, the freight forwarders keep their demand information, operating costs, and reservation prices private to protect their interests (Amaruchkul et al., 2011). Gerlach et al. (2013) report that sharing information obtained through dual variables in an airline alliance requires antitrust immunities, and in practice, airlines do not prefer to exchange such information to protect their own interests. Due to the restrictions in information exchange, decomposition and decentralization approaches have been studied to minimize information sharing between parties in collaborative networks (Poundarikapuram and Veeramani, 2004; Albrecht and Stadtler, 2015; Singh and O'Keefe, 2016; Ding and Kaminsky, 2020). Research to date on decentralized collaborative decision-making problems has primarily adopted iterative negotiation-based approaches to decompose the centralized model, which requires complete information sharing. Although information exchange is reduced with the decentralized coordination and negotiation-based approaches, the parties still have to share some information on their individual operations, which may reveal confidential information about their activities. Several studies have pointed out that revealing primal or dual optimal solution can provide strategic advantage to other parties in the cooperation (Albrecht and Stadtler, 2015; Singh and O'Keefe, 2016; Lai et al., 2019; Rius-Sorolla et al., 2020). There is no mechanism available in the literature to coordinate independent cooperating parties while ensuring that the information shared by the parties remains private.

This paper considers a general setting for capacity collaboration in network revenue management problems, where applications can be found in airline alliances, air-cargo transportation (Wright, 2014; Houghtalen et al., 2011), collaborative logistics (Adenso-Díaz et al., 2014; Gansterer and Hartl, 2016; Jin et al., 2019), decentralized supply chains (Albrecht and Stadtler, 2015; Singh and O'Keefe, 2016). Considering the benefits of collaboration, we assume that multiple parties agree to collaborate by sharing some of their capacities. Each party also controls its own private capacities in addition to the shared resources. This collaboration is performed by constructing a large mathematical programming model available to all parties. The fundamental aim of the parties is to identify the optimal allocated capacities for the shared resources and to evaluate the opportunity costs of the capacities available to them, *i.e.*, dual variables. These opportunity costs are used in various capacity control policies. For instance, dual variables are used in well-known *bid-price control* policy to manage customer requests for quantity-based network revenue management problems (Phillips, 2005). In our setting, the parties jointly compute the optimal capacity allocations and bid-prices of their collaborative model without disclosing any private information, such as selling prices and local capacity restrictions. However, without the necessary and mostly private information about the collaborative network problem, the correct values of bid-prices and the capacity allocations for the shared resources cannot be computed. This lack of proper information about the network problem raises an important question: How can one compute the correct bid-prices and the capacity allocations of the shared resources that maximize the overall revenue under privacy concerns? This question constitutes the main motivation behind our current study. Thus, by answering this question, we can provide a mechanism for the parties to collaborate without revealing their private data.

Contributions. We present a new transformation-based approach that considers data privacy in collaborative network revenue management problem, where multiple parties agree to share some of the network capacities. The proposed approach allows partners to use their individual private data while solving the collaborative capacity control problem to identify the optimal capacity sharing setting for the alliance. In our setting, each party keeps its data private through random data masking. Unlike the previous decomposition methods proposed for decentralized collaborative resource sharing problems, this method does not require any unmasked information exchange among parties while solving the collaborative model. To the best of our knowledge, our approach is the first attempt in the literature to deal with data privacy, considering both primal and dual variables. Our analysis makes use of several previous privacy studies based on random transformations of the problem data. However, our focus on bid-prices allows us to extend these studies with new results about the privacy of dual solutions. We show that the original primal and dual optimal solutions can be derived from the proposed transformed data-private model. Furthermore, in a separate section, we discuss the security of our mathematical model, where we apply a special set of random matrices (M-matrices) for transforming the simple inequalities. This set of matrices is much larger than the set of permutation matrices used in other studies which enhances the security of the proposed method. We also contribute to that literature with a new result showing that even if a private dataset of a firm is guessed, a brute-force approach to obtain random matrices in order to reveal primal and dual optimal solutions is computationally infeasible. We support our results with a simulation study on a set of revenue management problems, where the network structure is taken from an actual firm and adapted to an alliance network. Finally, we remark that the steps that we follow in this study can be extended to other resource sharing applications, where linear programming is one of the fundamental tools, and data privacy is a major concern.

2. Review of Related Literature.

Collaboration via forming alliances is common in many industries. Therefore, the efficient capacity allocation among the involved parties has long constituted an intriguing research topic with applications in airlines (Wright et al., 2010; Topaloglu, 2012; Chun et al., 2017), logistics and maritime shipping (Agarwal and Ergun, 2010; Zheng et al., 2015; Gansterer and Hartl, 2018; Ding and Kaminsky, 2020), and retail industries (Guo and Wu, 2018). Previous literature on collaborative decision-making has primarily concentrated on the development of centralized models, which require information sharing among partners or with a central planner (Boyd, 1998; Topaloglu, 2012; Zheng et al., 2015). Recent studies have pointed out the impracticality of central planning due to the restrictions in information sharing. Belobaba and Jain (2013) and Wright (2014) have discussed the limitations in information sharing among alliance partners in airlines. Similarly, Albrecht and Stadtler (2015) have studied the collaboration in supply chains and pointed out that the current advanced supply chain planning systems assume complete information sharing between firms, and there is no mechanism to coordinate a system where the partners only share limited information. They have proposed a negotiation based scheme to coordinate collaborative parties in a decentralized supply chain environment. Due to limitations in information sharing, decentralized models have been studied for collaborative decision-making problems (Rius-Sorolla et al., 2020).

Poundarikapuram and Veeramani (2004) have proposed a decentralized decision-making framework based on the L-shaped method for a collaborative planning problem in a supply chain. This framework separates the centralized problem into a master problem that includes common variables for all parties and sub-problems with private local objectives and variables. The authors have presented an iterative procedure, where the parties can solve their local problems privately and disclose limited information to solve the master problem at each iteration. Topaloglu (2012) has studied a centralized alliance problem in airline revenue management. By relaxing the shared constraints with dual variables, he has proposed a decomposition approach to find booking limits for each alliance partner as well as bid-prices. Amaruchkul et al. (2011) have addressed data privacy in air-cargo transportation. The carrier allocates bulk cargo capacity to the forwarder that sells this capacity to individual customers. The authors have studied the capacity contract between these two partners when the forwarder has private information on demand distribution, operating costs, and reservation profits. Chun et al. (2017) have addressed the capacity exchange problem in maritime transportation and proposed a two-stage framework to obtain the optimal resource allocation policy between alliance partners. In the first stage, optimal capacity exchange amount is determined so that the total alliance profit is maximized. Given the allocated capacities, each party decides on the reservation price to maximize its own local objective in the second stage. Recently, Lai et al. (2019) have studied the capacity allocation problem for a freight alliance by considering the data privacy in revenues and profit margins. They assume that the alliance partners jointly book freight capacity from the market according to the forecasted shipping demand, and then share this capacity during the planning period. The authors have first studied the centralized capacity allocation problem assuming revenue information is public. Due to the difficulty in solving the centralized problem, they have proposed an iterative auction mechanism based on the primal-dual method to find the capacity allocation policy. In the designed auction mechanism, the alliance partners do not need to share their private data except the forecasted shipping demand.

Although decomposition or decentralization approaches reduce the information exchange among parties, they do not completely ensure data privacy. We review two main approaches in privacy preserving methods for optimization and data analysis: cryptographic and non-cryptographic approaches (Weeraddana et al., 2013). Cryptography-based techniques such as secure multiparty computation allow several parties to do computation jointly over their inputs while ensuring that they remain private to each party. There is an extensive literature on cryptographic methods, yet computational issues are always of concern. In fact, forming a completely secure protocol for an optimization problem requires very high computational power (Hong et al., 2018). Li and Atallah (2006) have presented a secure simplex algorithm for a setting, where the objective function and the constraints are arbitrarily partitioned. Toft (2009) has followed the same scheme with Li and Atallah and presented a protocol for solving linear programs using a secure simplex algorithm. However, Dreier and Kerschbaum (2011) have implemented the algorithm proposed by Toft (2009) and pointed out that it is computationally inefficient even for small-scale problems.

Transformation is a non-cryptographic technique that involves converting a given linear optimization problem into a new problem via algebraic transformations, such that the solution of the new problem is the same as that of the original problem (Mangasarian, 2012; Wang et al., 2011). This enables parties to disguise private data effectively while preserving the quality of the solution. Du (2001) and Vaidya (2009a) have used transformation method in linear programming models. However, Bednarz (2012) has shown that this method is open to information acquisition attacks; that is, the private coefficients in the model can be learned by others. Mangasarian (2011, 2012) has proposed transformation techniques for vertically and horizontally partitioned linear programming models. Li et al. (2013) have extended this approach by incorporating inequality constraints to horizontally partitioned linear programming models. Hong and Vaidya (2014) have showed that the transformation method proposed by Li et al. (2013) is also open to attacks. There are few other transformation approaches for privately solving collaborative linear programming problems (Hong et al., 2018; Weeraddana et al., 2013). To the best of our knowledge, all of these transformation approaches focus on privacy in input data (private data) and primal optimal solution. How the dual solution is affected by the transformation applied to the primal model is unexplored.

3. Data-Private Capacity Control. Consider a collaborative network revenue management problem where multiple parties cooperate to share several capacities of the network for their own resource allocation systems. In addition to the shared capacities, each party controls its own private capacities. The aim of the parties is to decide on the optimal allocated capacities for the shared resources and to identify the bid-prices of the capacities available to them. In this setting, parties jointly build and solve the capacity sharing problem without disclosing any private information about their operations. Parties set a partnership agreement at the beginning of the planning horizon. Therefore, a partner does not have any access to the private data of any other partner in the cooperation. Depending on the industry, this private data may be revenues, available resources, operation routes, and demand information.

Like other studies in the literature, we also assume that the parties in an alliance cooperate truthfully (Krajewska and Kopfer, 2006; Topaloglu, 2012; Hyndman et al., 2013). This assumption is known as semi-honest behaviour in computer science literature (Vaidya, 2009b). It implies that the involved parties do not alter their data to get a better position in the collaboration. This assumption is naturally satisfied for several important reasons: If one of the parties strategically manipulate the collaboration and secure most of the shared capacities, then there can be legal consequences or simply loss-of-goodwill for their business. Firms must consider the long-term prospects of future collaborations that can be jeopardized with an opportunistic behaviour. Moreover, as the optimal primal and dual solutions do not result from the correct network information, they are of no use for controlling the shared capacities. Such a harmful move from one of the parties can also cause other parties to opt out of the partnership in the next round. Even worse, the other parties may start altering their data, which also eventually leads to total collapse of the collaboration. As a final example, consider a holding company that owns all the parties in the collaboration. Due to legal regulations, it may be impossible to share data among the involved parties without transformation. In such a setting, there is no incentive for any participant to manipulate the collaboration at the expense of the others.

3.1 Capacity Sharing Model. In this section, we first describe the general type of collaborative network revenue management problems that can be solved via our approach. We start with an illustration of the proposed capacity sharing setting. Figure 1 shows a simple network structure for two parties. Each link between a pair of nodes corresponds to a shared or private capacity on the network. The first party operates the capacities (1-3) and (3-4), whereas the second party operates the capacities (2-3) and (2-4). The second party also uses the capacity (3-4) shared by the first party. A sequence of links constitutes a path (*e.g.* route or itinerary). The set of paths listed in Figure 1 shows a number of origin-destination combinations. Moreover, on each path there could be multiple products with different revenues. An example could be the set of different transshipment prices on a route that involves several stops with different truck capacities on a transportation network. Birbil et al. (2014) have also explored this network structure and proposed a framework based on path decomposition. They treat each path as a single resource problem for a fixed capacity and solve an optimization problem over all possible allocations of the capacities. We next reconsider the path-based model of Birbil et al. (2014) and write it in a form that we can analyze to propose a data-private capacity control approach.

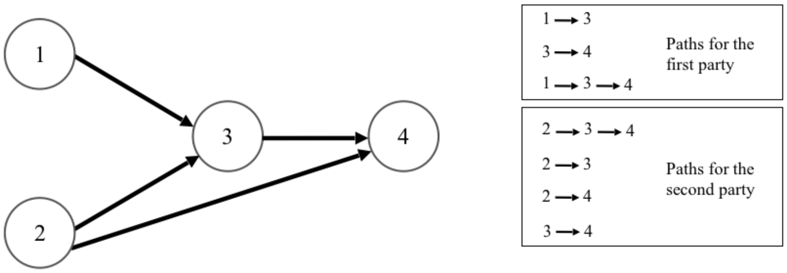


Figure 1: An illustrative network structure for two parties. The capacity on path 1 → 3 is private for the first party, whereas the capacities on paths 2 → 3 and 2 → 4 are private for the second party. Two parties share the capacity on path 3 → 4.

Before we present our main capacity sharing model, let us give our notation. We denote the set of parties by $\mathcal{K}$ and the set of paths controlled by party $k \in \mathcal{K}$ is denoted by $\mathcal{S}_k$. Let $\mathcal{J}$ be the set of m capacities shared by at least two parties. In addition, each party $k \in \mathcal{K}$ has its own m_k private capacities given by the set $\mathcal{J}_k$. If the collection of paths is $\mathcal{S}$ and x_s is the allocated capacity to path $s \in \mathcal{S}$, then the generic model becomes

$$\begin{aligned} \text{maximize} \quad & \sum_{s \in \mathcal{S}} \phi_s(x_s), & \\ \text{subject to} \quad & \sum_{s \in \mathcal{S}} a_{js} x_s \leq c_j, & j \in \mathcal{J}, \\ & \sum_{s \in \mathcal{S}_k} a_{js} x_s \leq c_j, & j \in \mathcal{J}_k, k \in \mathcal{K}, \\ & x_s \geq 0, & s \in \mathcal{S}, \end{aligned} \tag{1}$$

where $a_{js} = 1$, if path s uses one unit from capacity j; 0, otherwise. The shared capacities are denoted by c_j, $j \in \mathcal{J}$, and c_j, $j \in \mathcal{J}_k$ stand for the private capacities of party $k \in \mathcal{K}$. The first set of constraints ensures that the capacity allocation decision for paths do not violate the shared capacities. The second set of constraints guarantees that the capacity allocation decisions to paths for party $k \in \mathcal{K}$ do not exceed the private capacity limits for that party. The function $\phi_s(x_s)$ for a given x_s is itself evaluated by solving an optimization problem that yields the allocation decision of x_s capacities to different classes with the objective of maximizing revenue. For instance, $\phi_s(x_s)$ may correspond to a stochastic dynamic programming or a deterministic programming model constructed for capacity allocation problem for each path s. We assume that $x_s \mapsto \phi_s(x_s)$, $s \in \mathcal{S}$ are discrete concave functions. Birbil et al. (2014) have discussed that many well-known single dimension capacity control models proposed in the revenue management literature satisfy this assumption. Any of these models can be used to construct the objective function. We refer to Birbil et al. (2014) for an elaborate discussion on different dynamic and static network revenue management problems that can be considered within this generic structure.

As the objective function is concave and separable, we can replace it by a piece-wise linear concave function and reformulate the problem as a linear program. Dantzig (1956) has proposed an approach which represents the concave objective function as an indefinite integral and approximates it by a sum over fixed intervals. Following this approach, we can reformulate the model (1) as follows:

$$
\begin{aligned}
Z = \text{maximize} \quad & \sum_{k \in \mathcal{K}} \mathbf{r}_k^{\intercal} \mathbf{x}_k, && && (2)\\
\text{subject to} \quad & \sum_{k \in \mathcal{K}} \mathbf{A}_k \mathbf{x}_k \le \mathbf{c}, && && (\boldsymbol{\alpha}) \qquad (3)\\
& \mathbf{B}_k \mathbf{x}_k \le \mathbf{c}_k, && k \in \mathcal{K}, && (\boldsymbol{\alpha}_k) \qquad (4)\\
& \mathbf{0} \le \mathbf{x}_k \le \mathbf{1}, && k \in \mathcal{K}, && (5)
\end{aligned}
$$

where $\mathbf{1}$ and $\mathbf{0}$ stand for the vector of ones and the vector of zeros, respectively. The details of this model construction is given in Appendix A. In this model, the columns designated by subscript $k \in \mathcal{K}$ show all products owned by party k and the vector $\mathbf{r}_k \in \mathbb{R}^{n_k}$ denotes the corresponding expected revenues for the same party. The $m \times n_k$ incidence matrix $\mathbf{A}_k$ shows whether a product of party k uses the shared capacities. Likewise, the $m_k \times n_k$ matrix $\mathbf{B}_k$ consists of columns incident to the private capacities. The vectors $\boldsymbol{\alpha} \in \mathbb{R}^m$ and $\boldsymbol{\alpha}_k \in \mathbb{R}^{m_k}$, $k \in \mathcal{K}$ given in parentheses are the dual variables (bid-prices) associated with the common and the individual capacity constraints, respectively. After solving this problem, each party obtains its own optimal allocations and bid-prices. They also receive the optimal common bid-price vector corresponding to the shared capacities. These bid-prices can be used by the parties to implement their decision-making policies. Therefore, the decision vector $\mathbf{x}_k$ and individual dual variables $\boldsymbol{\alpha}_k$ are private to party $k \in \mathcal{K}$. In our subsequent discussion, the optimal values of capacity allocations and bid-prices for each party and the optimal bid-price vector for shared capacities are denoted by $\mathbf{x}_k^*$, $\boldsymbol{\alpha}_k^*$ and $\boldsymbol{\alpha}^*$. Before discussing the data-private mathematical model, let us define formally what constitutes as the private dataset for each party.

Definition 3.1 *In multi-party capacity sharing problem* (2)-(5)*, the private dataset for party $k \in \mathcal{K}$ consists of the matrices $\mathbf{A}_k$, $\mathbf{B}_k$, and the vectors $\mathbf{c}_k$, $\mathbf{r}_k$.*

One may question why multiple parties would prefer solving the network problem collectively? Instead, the shared capacity, $\mathbf{c}$ may be partitioned among the parties and each party can solve its problem independently. That is, each party $k \in \mathcal{K}$ receives its share of the capacity denoted by $\mathbf{s}_k$ such that $\mathbf{c} = \sum_{k \in \mathcal{K}} \mathbf{s}_k$. Then, the same party k can solve the following problem without sharing any private data:

$$
Z_k = \text{maximize} \quad \mathbf{r}_k^{\intercal} \mathbf{x}_k, \qquad (6)
$$

$$\begin{aligned} \text{subject to} \quad & \mathbf{A}_k \mathbf{x}_k \leq \mathbf{s}_k, && (7)\\ & \mathbf{B}_k \mathbf{x}_k \leq \mathbf{c}_k, && (8)\\ & \mathbf{0} \leq \mathbf{x}_k \leq \mathbf{1}. && (9) \end{aligned}$$

However, this approach has three drawbacks: First, it is not clear how to determine the optimal partitioning of the shared capacity among the parties without having the complete information of the network problem. As a result of this suboptimal partitioning, the capacities allocated to a party might be left unused, even though those capacities could have been filled up by the other parties, if they had been shared. These adverse effects of the defragmentation of network capacities have also been observed in the literature (Curry, 1990; Kunnumkal and Topaloglu, 2010). This suboptimal partitioning leads us to the second drawback. The pre-allocation of the common capacities yields less total expected revenue than that of the collective model (2)-(5). In other words, irrespective of the way in which the common capacity is shared, we have $\sum_{k\in\mathcal{K}} Z_k \leq Z$. This result simply follows from the fact that the collection of feasible solutions to each (6)-(9) is also a feasible solution to (2)–(5). Third, the primal and dual variables obtained from the individual models depend on the partitioning of the common capacity and lack information about the entire network. Overall, it is more beneficial for all parties to collaborate and solve the collective model (2)–(5).

3.2 Data-Private Mathematical Model. Even though they may have agreed to collaborate, the major concern for the parties is the privacy of the input data (see Definition 3.1) and their sensitive decisions when solving the joint problem (2)-(5). In our problem setup, the parties keep their revenue and capacity vectors as well as their product matrices private. Although the parties sign up for sharing some capacities on the network, they do not share any information about their individual capacities. Thus, only the shared resource capacities are not private for the involved parties. In the subsequent part of this section, we present the steps for the parties to randomly transform their private input and output data in order to ensure data-privacy while collectively solving the network problem. Then, with this transformed data, the overall private model is constructed and made available to all parties. We conclude this section with our key result, which shows that parties can still recover their optimal allocations and bid-prices after the proposed private model is solved by each party. In our following discussion, we use the term *masked problem* to refer to the resulting problem after applying random transformations to the input data.

First, we start with concealing the private output; that is, the individual optimal capacity allotments ($\mathbf{x}_k^*$) and the individual bid-prices ($\boldsymbol{\alpha}_k^*$). To this end, we first ask each party $k \in \mathcal{K}$ to generate its own private pair of random vectors, $\boldsymbol{\eta}_k \in \mathbb{R}^{n_k}$ and $\boldsymbol{\xi}_k \in \mathbb{R}^{m_k}$ to transform the primal and dual solutions. Then, we use the auxiliary variables $\mathbf{z}_k$ and $\mathbf{v}_k$ for $k \in \mathcal{K}$ to construct the following mathematical model:

$$\begin{aligned} \text{maximize} \quad & \sum_{k\in\mathcal{K}} (\mathbf{r}_k + \mathbf{B}_k^\intercal \boldsymbol{\xi}_k)^\intercal \mathbf{z}_k + \sum_{k\in\mathcal{K}} \boldsymbol{\xi}_k^\intercal \mathbf{v}_k && && (10)\\ \text{subject to} \quad & \sum_{k\in\mathcal{K}} \mathbf{A}_k \mathbf{z}_k \leq \mathbf{c} + \sum_{k\in\mathcal{K}} \mathbf{A}_k \boldsymbol{\eta}_k, && & (\boldsymbol{\beta}) & (11)\\ & \mathbf{B}_k \mathbf{z}_k + \mathbf{v}_k = \mathbf{c}_k + \mathbf{B}_k \boldsymbol{\eta}_k, && k \in \mathcal{K}, & (\boldsymbol{\beta}_k) & (12)\\ & \mathbf{z}_k \leq \mathbf{1} + \boldsymbol{\eta}_k, && k \in \mathcal{K}, & & (13)\\ & \mathbf{z}_k \geq \boldsymbol{\eta}_k, && k \in \mathcal{K}, & & (14)\\ & \mathbf{v}_k \geq \mathbf{0}, && k \in \mathcal{K}, & & (15) \end{aligned}$$

where the vectors in parentheses are again the dual vectors associated with the corresponding constraints. The constraints (11) and (12) correspond to the capacity constraints (3) and (4) in path-based formulation (2)–(5), respectively. Note that, we use the transformations $\mathbf{z}_k = \mathbf{x}_k + \boldsymbol{\eta}_k$ and $\mathbf{v}_k = \mathbf{c}_k - \mathbf{B}_k \mathbf{x}_k$ for $k \in \mathcal{K}$ to obtain the new model. Since $\mathbf{x}_k$ is increased by $\boldsymbol{\eta}_k$, the right hand side of constraints (3) and (4) are increased by $\sum_{k\in\mathcal{K}} \mathbf{A}_k\boldsymbol{\eta}_k$ and $\mathbf{B}_k\boldsymbol{\eta}_k$ in constraints (11) and (12), respectively. Given that the auxiliary variable $\mathbf{v}_k$ corresponds to the slack of constraints (3), we can rewrite the constraint (12) as an equality. The new auxiliary variable $\mathbf{v}_k$ and its cost coefficient $\boldsymbol{\xi}_k$ are introduced to make sure that the dual optimal solution is shifted with a random vector. The following lemma formally shows that the optimal allocations and the dual variables for each party are indeed perturbed with private random noise vectors after solving this problem. As long as each party $k \in \mathcal{K}$ does not share its random vectors $\boldsymbol{\eta}_k$ and $\boldsymbol{\xi}_k$ with the other parties, the individual optimal capacity allotments and the individual bid-prices remains private. The proof of the lemma is given in the Appendix B.

Lemma 3.1 *If we denote the primal optimal solution of* (10)-(15) *by* $(\mathbf{z}_k^*, \mathbf{v}_k^*)_{k\in\mathcal{K}}$ *and the dual optimal variables associated with the capacity constrains by* $(\boldsymbol{\beta}^*, \boldsymbol{\beta}_k^*)_{k\in\mathcal{K}}$*, then we have*

$$\begin{aligned}
\mathbf{z}_k^* &= \mathbf{x}_k^* + \boldsymbol{\eta}_k, \quad k \in \mathcal{K},\\
\boldsymbol{\beta}^* &= \boldsymbol{\alpha}^*,\\
\boldsymbol{\beta}_k^* &= \boldsymbol{\alpha}_k^* + \boldsymbol{\xi}_k, \quad k \in \mathcal{K}.
\end{aligned}$$

We note that, in order to solve problem (10)-(15), each party $k \in \mathcal{K}$ still needs to reveal its pair of random vectors, $\boldsymbol{\eta}_k$ and $\boldsymbol{\xi}_k$ so that the objective function and the bound constraints can be constructed. Consequently, the optimal allocations and the dual variables of each party are no longer private. Thus, our next step is to conceal the random vectors by using a linear transformation. That is, we set $\mathbf{v}_k = \mathbf{E}_k^\intercal \mathbf{w}_k$ for $k \in \mathcal{K}$, where $\mathbf{E}_k$ is a $t_k \times m_k$ random matrix with $t_k \geq m_k$. Likewise, we can also set $\mathbf{z}_k = \mathbf{D}_k^\intercal \mathbf{u}_k$ for $k \in \mathcal{K}$, where $\mathbf{D}_k$ is a $s_k \times n_k$ random matrix with $s_k \geq n_k$. This is, in fact, the transformation proposed by Mangasarian (2011). We note that we can simply form matrices $\mathbf{D}_k$ and $\mathbf{E}_k$ with real or rational random values. Then, the resulting matrices are almost-surely full rank (Feng and Zhang, 2007). We then obtain

$$\begin{aligned}
\text{maximize} \quad & \sum_{k\in\mathcal{K}} (\mathbf{r}_k + \mathbf{B}_k^\intercal \boldsymbol{\xi}_k)^\intercal \mathbf{D}_k^\intercal \mathbf{u}_k + \sum_{k\in\mathcal{K}} \boldsymbol{\xi}_k^\intercal \mathbf{E}_k^\intercal \mathbf{w}_k && && (16)\\
\text{subject to} \quad & \sum_{k\in\mathcal{K}} \mathbf{A}_k \mathbf{D}_k^\intercal \mathbf{u}_k \leq \mathbf{c} + \sum_{k\in\mathcal{K}} \mathbf{A}_k \boldsymbol{\eta}_k, && && (17)\\
& \mathbf{B}_k \mathbf{D}_k^\intercal \mathbf{u}_k + \mathbf{E}_k^\intercal \mathbf{w}_k = \mathbf{c}_k + \mathbf{B}_k \boldsymbol{\eta}_k, && k \in \mathcal{K}, && (18)\\
& \mathbf{D}_k^\intercal \mathbf{u}_k \leq \mathbf{1} + \boldsymbol{\eta}_k, && k \in \mathcal{K}, && (19)\\
& \mathbf{D}_k^\intercal \mathbf{u}_k \geq \boldsymbol{\eta}_k, && k \in \mathcal{K}, && (20)\\
& \mathbf{E}_k^\intercal \mathbf{w}_k \geq \mathbf{0}, && k \in \mathcal{K}. && (21)
\end{aligned}$$

Nonetheless, this transformation is still not enough to conceal the data or the random vectors because the parties have to explicitly share the random matrices $\mathbf{E}_k$ due to constraints (18) and (21). Likewise, the random matrices $\mathbf{D}_k$ as well as the random vectors $\boldsymbol{\eta}_k$ need to be revealed because of the bound constraints (19)-(20). In fact, Mangasarian (2011) deals only with linear programming models *without* bound constraints and mentions that there is “a difficulty associated with possibly including non-negativity constraints.” Li et al. (2013) include inequality constraints, and resolve this privacy issue by allowing each party to generate a positive diagonal random matrix for their slack variables. Their approach has been shown to be open to attacks (Hong and Vaidya, 2014). We, on the other hand, propose to sample from the set of M-matrices for which the positive diagonal matrices constitute a subset. This choice is valid because if $\mathbf{S}$ is an M-matrix, then $\mathbf{Sx} \geq \mathbf{0}$ implies $\mathbf{x} \geq \mathbf{0}$ (Horn and Johnson, 1991). This leads to the following model:

$$\text{maximize} \quad \sum_{k\in\mathcal{K}} (\mathbf{r}_k + \mathbf{B}_k^\intercal \boldsymbol{\xi}_k)^\intercal \mathbf{D}_k^\intercal \mathbf{u}_k + \sum_{k\in\mathcal{K}} \boldsymbol{\xi}_k^\intercal \mathbf{E}_k^\intercal \mathbf{w}_k \qquad (22)$$

$$
\begin{aligned}
\text{subject to}\quad & \sum_{k\in\mathcal{K}} \mathbf{A}_k\mathbf{D}_k^\intercal\mathbf{u}_k \leq \mathbf{c} + \sum_{k\in\mathcal{K}} \mathbf{A}_k\boldsymbol{\eta}_k, && && (23)\\
& \mathbf{F}_k\mathbf{B}_k\mathbf{D}_k^\intercal\mathbf{u}_k + \mathbf{F}_k\mathbf{E}_k^\intercal\mathbf{w}_k = \mathbf{F}_k(\mathbf{c}_k + \mathbf{B}_k\boldsymbol{\eta}_k), && k\in\mathcal{K}, && (24)\\
& \mathbf{G}_k\mathbf{D}_k^\intercal\mathbf{u}_k \leq \mathbf{G}_k(\mathbf{1}+\boldsymbol{\eta}_k), && k\in\mathcal{K}, && (25)\\
& \mathbf{H}_k\mathbf{D}_k^\intercal\mathbf{u}_k \geq \mathbf{H}_k\boldsymbol{\eta}_k, && k\in\mathcal{K}, && (26)\\
& \mathbf{L}_k\mathbf{E}_k^\intercal\mathbf{w}_k \geq \mathbf{0}, && k\in\mathcal{K}, && (27)
\end{aligned}
$$

where the $m_k \times m_k$ matrices $\mathbf{F}_k$ and $\mathbf{L}_k$ as well as the $n_k \times n_k$ matrices $\mathbf{G}_k$ and $\mathbf{H}_k$ are all M-matrices. In order to conceal the random vector η_k and the random matrices $\mathbf{D}_k$ and $\mathbf{E}_k$ for $k \in \mathcal{K}$, we multiply constraints (18)-(21) by M-matrices $\mathbf{F}_k$, $\mathbf{G}_k$, $\mathbf{H}_k$ and $\mathbf{L}_k$, respectively and obtain constraints (24) - (27). Model (22) - (27) ensures privacy of individual input and output data of each party while parties solve their joint capacity control problem. Next, we give the formal definition of M-matrix that is used to conceal data in our data-private model (22) - (27).

Definition 3.2 (M-matrix (Poole and Boullion, 1974)) *An $\ell \times \ell$ matrix $\mathbf{M}$ that can be expressed in the form $\mathbf{M} = s\mathbf{I} - \mathbf{N}$, where $\mathbf{N} = (n_{ij})$ with $n_{ij} \geq 0,\ i, j \in 1, \ldots, \ell$, and $s > \rho(\mathbf{N})$ is called an M-matrix where $\rho(\mathbf{N}) = \max\{|\lambda| : \det(\lambda\mathbf{I} - \mathbf{N}) = 0\}$.*

This definition also gives a procedure to obtain a random M-matrix: First sample a random nonnegative $\mathbf{N}$ matrix and select a random $s > \rho(\mathbf{N})$. Then, $s\mathbf{I} - \mathbf{N}$ becomes an M-matrix. This simple procedure clearly shows that it is possible to produce infinitely many M-matrices. We will make use of this observation, when we discuss the security of our transformed problem in the next section.

To simplify our notation, we further define for $k \in \mathcal{K}$ the following

$$
\begin{array}{cccc}
\bar{\mathbf{r}}_k = \mathbf{D}_k(\mathbf{r}_k + \mathbf{B}_k^\intercal\boldsymbol{\xi}_k), & \bar{\boldsymbol{\xi}}_k = \mathbf{E}_k\boldsymbol{\xi}_k, & \bar{\mathbf{A}}_k = \mathbf{A}_k\mathbf{D}_k^\intercal, & \bar{\mathbf{c}} = \mathbf{c} + \sum_{k\in\mathcal{K}} \mathbf{A}_k\boldsymbol{\eta}_k = \mathbf{c} + \sum_{k\in\mathcal{K}} \tilde{\boldsymbol{\eta}}_k\\
\bar{\mathbf{B}}_k = \mathbf{F}_k\mathbf{B}_k\mathbf{D}_k^\intercal, & \bar{\mathbf{F}}_k = \mathbf{F}_k\mathbf{E}_k^\intercal & \bar{\mathbf{c}}_k = \mathbf{F}_k(\mathbf{c}_k + \mathbf{B}_k\boldsymbol{\eta}_k), & \bar{\mathbf{G}}_k = \mathbf{G}_k\mathbf{D}_k^\intercal\\
\bar{\mathbf{1}}_k = \mathbf{G}_k(\mathbf{1}+\boldsymbol{\eta}_k) & \bar{\mathbf{H}}_k = \mathbf{H}_k\mathbf{D}_k^\intercal, & \bar{\boldsymbol{\eta}}_k = \mathbf{H}_k\boldsymbol{\eta}_k, & \bar{\mathbf{L}}_k = \mathbf{L}_k\mathbf{E}_k^\intercal,
\end{array}
\tag{28}
$$

and rewrite model (22)-(27) as

$$
\begin{aligned}
\bar{Z} = \text{maximize}\quad & \sum_{k\in\mathcal{K}} \bar{\mathbf{r}}_k^\intercal\mathbf{u}_k + \sum_{k\in\mathcal{K}} \bar{\boldsymbol{\xi}}_k^\intercal\mathbf{w}_k && && && (29)\\
\text{subject to}\quad & \sum_{k\in\mathcal{K}} \bar{\mathbf{A}}_k\mathbf{u}_k \leq \bar{\mathbf{c}}, && && (\boldsymbol{\gamma}) && (30)\\
& \bar{\mathbf{B}}_k\mathbf{u}_k + \bar{\mathbf{F}}_k\mathbf{w}_k = \bar{\mathbf{c}}_k, && k\in\mathcal{K}, && (\boldsymbol{\gamma}_k) && (31)\\
& \bar{\mathbf{G}}_k\mathbf{u}_k \leq \bar{\mathbf{1}}_k, && k\in\mathcal{K}, && && (32)\\
& \bar{\mathbf{H}}_k\mathbf{u}_k \geq \bar{\boldsymbol{\eta}}_k, && k\in\mathcal{K}, && && (33)\\
& \bar{\mathbf{L}}_k\mathbf{w}_k \geq \mathbf{0}, && k\in\mathcal{K}, && && (34)
\end{aligned}
$$

where $(\boldsymbol{\gamma}, \boldsymbol{\gamma}_k)_{k\in\mathcal{K}}$ are the dual variables. The following theorem shows that after the random transformations the *exact* primal and dual solutions of the original problem can easily be recovered. The proof of this theorem is given in the appendix.

Theorem 3.1 *Let $(\mathbf{u}_k^*, \mathbf{w}_k^*)_{k\in\mathcal{K}}$ and $(\boldsymbol{\gamma}^*, \boldsymbol{\gamma}_k^*)_{k\in\mathcal{K}}$ be the primal and dual optimal solutions of (29)-(34). Using again the primal and dual optimal solutions, $(\mathbf{x}_k^*)_{k\in\mathcal{K}}$ and $(\boldsymbol{\alpha}^*, \boldsymbol{\alpha}_k^*)_{k\in\mathcal{K}}$ of the original problem (2)-(5), we* *obtain*

$$\begin{aligned} Z &= \bar{Z} - \textstyle\sum_{k\in\mathcal{K}} \mathbf{r}_k^\intercal \boldsymbol{\eta}_k - \sum_{k\in\mathcal{K}} (\mathbf{c}_k + \mathbf{B}_k\boldsymbol{\eta}_k)^\intercal \boldsymbol{\xi}_k, & \\ \mathbf{x}_k^* &= \mathbf{D}_k^\intercal \mathbf{u}_k^* - \boldsymbol{\eta}_k, & k \in \mathcal{K}, \\ \boldsymbol{\alpha}^* &= \boldsymbol{\gamma}^*, & \\ \boldsymbol{\alpha}_k^* &= \mathbf{F}_k^\intercal \boldsymbol{\gamma}_k^* - \boldsymbol{\xi}_k, & k \in \mathcal{K}. \end{aligned}$$

With this main theorem, we conclude that the parties can safely obtain their own eaxact solutions, since the set of random matrices designated with subscript k is known only to the individual party $k \in \mathcal{K}$. It is important to note that the parties can generate their primal solutions by using the random matrices, yet the dual solutions of the original problem are exactly the same as the transformed problem ($\boldsymbol{\gamma}^* = \boldsymbol{\alpha}^*$).

Algorithm 1 presents our transformation-based protocol and shows how one party ($\hat{k}$ in the algorithm) can apply the data-private capacity control. In Step 1, the party prepares the input data. This data is transformed in Step 2, and shared with the other parties. Now, the input for the overall private model is available to everyone (Step 3). Each party then solves the private problem and obtains the optimal solutions in Step 4. Then, party $\hat{k}$ recovers its optimal dual variable vectors in Step 5. As a result, using Lemma 3.1 and Theorem 3.1, we show that the optimal primal and dual solutions of the original model can be obtained safely from the optimal primal and dual solutions of the transformed model. Hence, we conclude that the correctness of the original problem is preserved.

Algorithm 1 Data-Private Capacity Control for Party $\hat{k} \in \mathcal{K}$

```
1: Compile private individual input
       $\boldsymbol{\xi}_{\hat{k}}, \boldsymbol{\eta}_{\hat{k}}, \mathbf{D}_{\hat{k}}, \mathbf{E}_{\hat{k}}, \mathbf{F}_{\hat{k}}, \mathbf{G}_{\hat{k}}, \mathbf{H}_{\hat{k}}, \mathbf{L}_{\hat{k}}.$
2: Transform individual input using (28) and share
       $\bar{\mathbf{r}}_{\hat{k}}, \bar{\boldsymbol{\xi}}_{\hat{k}}, \bar{\mathbf{A}}_{\hat{k}}, \bar{\mathbf{B}}_{\hat{k}}, \bar{\mathbf{F}}_{\hat{k}}, \bar{\mathbf{c}}_{\hat{k}}, \bar{\mathbf{G}}_{\hat{k}}, \bar{\mathbf{1}}_{\hat{k}}, \bar{\mathbf{H}}_{\hat{k}}, \bar{\boldsymbol{\eta}}_{\hat{k}}, \bar{\mathbf{L}}_{\hat{k}}, \tilde{\boldsymbol{\eta}}_{\hat{k}}.$
3: Store all transformed data
       $(\bar{\mathbf{r}}_k, \bar{\boldsymbol{\xi}}_k, \bar{\mathbf{A}}_k, \bar{\mathbf{B}}_k, \bar{\mathbf{F}}_k, \bar{\mathbf{c}}_k, \bar{\mathbf{G}}_k, \bar{\mathbf{1}}_k, \bar{\mathbf{H}}_k, \bar{\boldsymbol{\eta}}_k, \bar{\mathbf{L}}_k, \tilde{\boldsymbol{\eta}}_k)_{k\in\mathcal{K}}.$
4: Solve (29)-(34) with $\bar{\mathbf{c}} = \mathbf{c} + \sum_{k\in\mathcal{K}} \mathbf{A}_k\boldsymbol{\eta}_k$ and the stored data. Obtain transformed optimal solution
       $(\mathbf{u}_k^*, \mathbf{w}_k^*, \boldsymbol{\gamma}^*, \boldsymbol{\gamma}_k^*)_{k\in\mathcal{K}}.$
5: Recover private individual output using the transformed solution and the private input in Step 1:
       $\mathbf{x}_{\hat{k}}^* = \mathbf{D}_{\hat{k}}^\intercal \mathbf{u}_{\hat{k}}^* - \boldsymbol{\eta}_{\hat{k}},\ \ \boldsymbol{\alpha}^* = \gamma^*,\ \ \boldsymbol{\alpha}_{\hat{k}}^* = \mathbf{F}_{\hat{k}}^\intercal \gamma_{\hat{k}}^* - \boldsymbol{\xi}_{\hat{k}}.$
```

3.3 Security. We next discuss the security of our data private model (29)-(34) in the presence of attacks. With transformation-based approaches, there is an overarching trade-off between efficiency and security (Goldreich, 2009). Actually, Laud and Pankova (2013) show that it is impossible to achieve an information theoretical security with the transformation techniques of multiplication, scaling, permutation and shifting; a point that is also noted by Dreier and Kerschbaum (2011) in their work on linear programming. In transformation-based methods, the concern is how much information may leak to other parties during the transformation. Indeed, this concern is also raised by Du and Zhan (2002) as to quantify the security achieved in each transformation-based protocol, so that these protocols can be compared in terms of the level of security they can achieve. To be able to attain this, the notion of information leakage, also known as the vulnerability of the system, is introduced by Braun et al. (2009) in which it is defined as "the amount of information learnt by the adversary by observing the output of the protocol.” In a more recent study, Dreier and Kerschbaum (2011) use information leakage concept to measure how much information about the private information is revealed to an adversary. They quantify this leakage when the parameters of a linear program are masked with random matrices. In their analysis, the components of all data and random matrices are assumed to be nonnegative integers, and the leakage results depend on the largest components of the random matrices. Even though our input matrices are binary, we do not impose upper bounds or integrality requirements on the components of the random vectors or matrices (not necessarily square) used in our transformations. It is crucial to point out that the authors only use *positive monomial matrices* (permutation matrices with nonnegative pivot elements) to deal with the inequalities, and state that these matrices cause vulnerability in security. Unlike these simple permutation matrices, we sample from the much larger set of M-matrices. In fact, as Definition 3.2 and the subsequent paragraph show, this set is uncountable.

Recall that the matrices $\mathbf{A}_k$ and $\mathbf{B}_k$ constitute the products for each firm $k \in \mathcal{K}$. In certain applications, one of the parties may be able to guess the products and the capacities of the other parties. For instance, in airline revenue management, these products correspond to different fare class itineraries which may be collected by web-scraping. As Theorem 3.1 and (28) show, the security of our transformations relies mainly on private random matrices $\mathbf{D}_k$, $\mathbf{F}_k$, $\boldsymbol{\eta}_k$ and $\boldsymbol{\xi}_k$. We show in the next lemma that even all private data (see Definition 3.1) of a firm are perfectly guessed, a brute-force approach to obtain the private random matrices is computationally infeasible. The proof of this lemma is given in the appendix.

Lemma 3.2 *Suppose for $k \in \mathcal{K}$ that $1 \leq m < n_k \leq s_k$, $1 < m_k \leq t_k$, and both $\mathbf{A}_k$ and $\mathbf{B}_k$ have full rank. Even if all private data of party $k \in \mathcal{K}$ are known (see Definition 3.1), then finding any one of $\mathbf{D}_k$, $\mathbf{F}_k$, $\boldsymbol{\eta}_k$ or $\boldsymbol{\xi}_k$ requires obtaining a particular solution to a system of linear equations with infinitely many solutions.*

The condition in Lemma 3.2 implies that each party should participate in this collaboration with multiple individual capacities and multiple products. This is a reasonable assumption, since it is easier for the other parties to guess the actual values for very small datasets. We note that even if the dataset of a party is small, the same party can still enlarge its dataset by adding redundant constraints or dummy products.

4. Simulation Study.

We devote this section to our simulation study for discussing different aspects of our proposed data-private model. In particular, we investigate the impact of capacity sharing and evaluate the computational performance of the data-private model. We next explain our simulation setup in detail and then present our numerical results.

4.1 Setup.

We design our experiments by using an airline network structure obtained from an actual firm. These data include flight legs with corresponding capacities, flight itineraries and origin-destination (OD) paths. Since the network data belongs to a single airline, it does not include any alliance information. In order to construct an alliance network, we randomly allocate OD-paths in each network to obtain artificially generated airline partners. The partners set a block space partnership agreement to share capacities on some of the flights at the beginning of the planning horizon. Although the real-time flight information such as marketed flight itineraries and associated prices can be partially available through online travel agencies during the sale season, the complete flight information including forecasted demand, prices and flight capacities are not available when the codeshare agreements are set at the beginning of the sale season.

We simulate the arrival of reservation requests over a planning horizon of length T. We assume that the booking requests for OD path $s \in \mathcal{S}$ arrive according to a homogeneous Poisson process with rate λ_s. Given that a booking request arrives for OD-path s at time period t, it is for product i with probability $p_{is}(t)$. The way we generate these arrival probabilities is quite similar to the one given by Birbil et al. (2014). The simulation process is defined as follows. We first generate the arrival times of booking requests for all

OD-paths over the planning horizon T. By using the arrival probabilities for products in each OD path, we find the product of the requests and apply the corresponding booking policies. To change the tightness of the flight capacities, we use a load factor parameter (ρ). The average arrival rate, μ_j for flight $j \in \mathcal{J}$ depends on the value of the load factor. This relation can be expressed as $\mu_j = \rho \frac{c_j}{TN_j}$, where N_j is the number of OD-paths using flight leg j. Then, the arrival rate λ_s for OD-path s is generated as follows $\lambda_s = \frac{\sum_{j \in J_s} \mu_j}{J_s}$, where J_s is the number of flight legs used by OD-path s.

Our experimental design is based on various factors. These are the number of alliance partners (K), the number of OD-paths (N) and the load factor (ρ). In simulation experiments, we design three alliance partnerships with different numbers of partners $K \in \{2, 4, 6\}$. We assume that all alliance partners have a similar market share in terms of number of OD-paths in the alliance network. We test three networks with sizes $N \in \{100, 200, 400\}$. We extract these networks from the overall network data, which include 119, 215 and 368 flight legs, and 869, 1,762 and 3,567 products, respectively. Since we randomly divide OD paths among the fictitious alliance partners, the number of shared flights can change depending on the allocated flights. Therefore, Table 1 presents the average number of shared flights in each subnetwork. The last parameter set comes from the load factor $\rho \in \{1.2, 1.6\}$ corresponding to medium and high loads, respectively. The computational results are reported over 100 simulation runs. We take the reservation period length as $T = 1,000$.

Table 1: The average number of shared flights in each network

Number of Parties	**Network Size (N)**		
(K)	**100**	**200**	**400**
2	9	32	81
4	16	42	109
6	18	43	124

4.2 Results. In this section, we conduct simulation experiments to evaluate the effects of collaborative capacity sharing and provide a sensitivity analysis with respect to various parameters. In particular, we investigate centralized coordination with complete information sharing, coordination with data privacy and individual control strategies. These strategies are formally introduced as follows:

Collaborative Capacity Planning (CP). This strategy assumes that alliance partners act collaboratively and the booking decisions for shared capacities are controlled through an integrated planning system which requires complete information sharing. Topaloglu (2012) describes this system as “centralized planning” where the booking decisions are made by considering the overall alliance benefit. CP solves the model (2)-(5) to compute the optimal values of the dual variables ($\boldsymbol{\alpha} \in \mathbb{R}^m$ and $\boldsymbol{\alpha}_k \in \mathbb{R}^{m_k}$, $k \in \mathcal{K}$) associated with the capacity constraints. When a request for a product arrives, the summation of the optimal dual variables corresponding to the used flight legs becomes the bid-price for accepting or rejecting the request. That is, assuming a product request using path s arrives for party k, we accept this request if the fare of the product is greater than or equal to $\sum_{j \in \mathcal{J}} a_{js}\alpha_j + \sum_{j \in \mathcal{J}_k} b_{js}\alpha_{jk}$. To consider the effects of reoptimization, we divide the planning horizon into five equal segments and resolve the model (2)-(5) at the beginning of each segment with the updated capacities. Since CP coordinates the booking decisions for the whole alliance, this strategy requires access to all flight information of the partners (the capacities on all flight legs and the expected demands for all products) which is quite unlikely to occur in practice (Topaloglu, 2012).

Coordinated Capacity Sharing (CCS). In this strategy, the parties come together and solve the data-private model (29)-(34) by transforming their private information to obtain the optimal values of the transformed dual variables and the capacity allocations (see Algorithm 1). After partners receive the transformed solution, each of them converts the transformed values to the original ones as shown in Theorem 3.1. During the booking horizon, each partner makes its own booking control decisions by using the optimal dual variables and the allocated leg capacities. Letting, $(\boldsymbol{\alpha}^*, \boldsymbol{\alpha}_k^*)_{k \in \mathcal{K}}$ be the recovered dual variables obtained by the dual optimal solution of model (29)-(34), we accept the arriving path s request if the fare of the product is greater than or equal to $\sum_{j \in \mathcal{J}} a_{js}\alpha_j^* + \sum_{j \in \mathcal{J}_k} b_{js}\alpha_{jk}^*$ and there is enough allocated leg capacity for the flights covered by path s. Similar to CP, we divide the booking horizon into five segments and resolve problem (29)-(34) at the beginning of each segment. Unlike CP, CCS does not require alliance partners to share any private information regarding flights.

Individual Control (IC). This strategy solves problem (6)-(9) for each partner. For shared flight-legs, partner-based capacity allocations are calculated with respect to the expected demands. In particular, letting d_{jk} be the demand for partner k in shared flight leg j, the allocated capacity for partner k is calculated as $\frac{d_{jk}}{\sum_{k \in K_j} d_{jk}} c_j$, where K_j is the set of partners using leg j. In this strategy, each partner makes its own booking control decisions by using the optimal bid-prices associated with capacity constraints in problem (6)-(9). Similar to previous strategies, we divide the planning horizon into five segments and revise the bid-prices at the beginning of each segment. IC strategy requires alliance partners to share their demand information in order to allocate the capacities of the shared flights.

Recall that the objective function value in path-based formulation can also be obtained by solving different single-capacity, static and dynamic programming models. Our approach here is applicable in all those cases. In our numerical experiments, we assume that the three strategies listed above use a deterministic linear programming (DLP) model to compute the booking control policies. The reason behind this choice is two-fold: First, DLP models are frequently used in the literature (Poundarikapuram and Veeramani, 2004; Albrecht and Stadtler, 2015). Second, bid-price control with a DLP model is a competitive strategy when compared against other static and dynamic network models (Talluri and van Ryzin, 2004).

Figure 2 shows our simulation results in terms of revenues (objective functions) for three alliance networks with two, four and six partners, respectively. In these figures, we present the relative differences with respect to the CP strategy, since it always performs better than the other two strategies. CP oversees the whole alliance network and makes accept-reject decisions for arriving reservation requests for all partners. On the other hand, airlines individually make their booking control decisions in strategies CCS and IC without sharing any information over the planning horizon. In Figure 2, the dashed line passing through 100 corresponds to CP, and bar charts are used to show the relative difference for strategies CCS and IC.

When we compare the respective performances, we observe that the average revenues obtained by CCS are very close to those obtained by CP, especially for the networks with 100 and 200 OD-paths. The average performance gaps between CP and CCS are only 0.20%, 0.35% and 0.85% for the problems with 100, 200 and 400 OD-paths, respectively. As Figure 2 illustrates, the performance gaps between CP and CCS slightly decrease when the load factor is high. We conjecture that CP and CCS make same booking decisions most of the time since both of these strategies solve the same centralized model to obtain their optimal booking policies The only difference is that CCS allocates shared flight capacities to partners; hence, each airline is restricted by that limit while making booking control decisions. CP strategy pools the capacities of the shared legs and does not consider the individual booking limits. When the arrival intensity is high, CCS can compensate the revenue loss due to these restrictive booking limits. We have validated that the relative differences between the total expected revenues obtained by CCS and CP are statistically significant at 95% level in 18 test scenarios. For the relative differences between the total expected revenues achieved by CCS and CP, we have failed to reject the null hypothesis in only one of the test instances. Thus, CCS can deliver similar results with the ideal case where each party shares its information.

Comparing IC with CP and CCS in Figure 2, we observe that IC obtains lower expected revenues in all cases. We notice that, as the number of OD-pairs (network size) and the alliance partners increases, it performance deteriorates. The expected revenues obtained with IC can lag on average 7.35% behind those obtained with CCS. This striking performance gap between strategies CCS and IC is due to the management of the shared capacities. While CCS solves the data-private model (29)-(34) to obtain booking control variables by considering the whole alliance network, IC allocates the capacities of the shared flight legs by only considering the expected demand information of each partner. This demonstrates the importance of considering overall network information while making capacity allocation decisions.

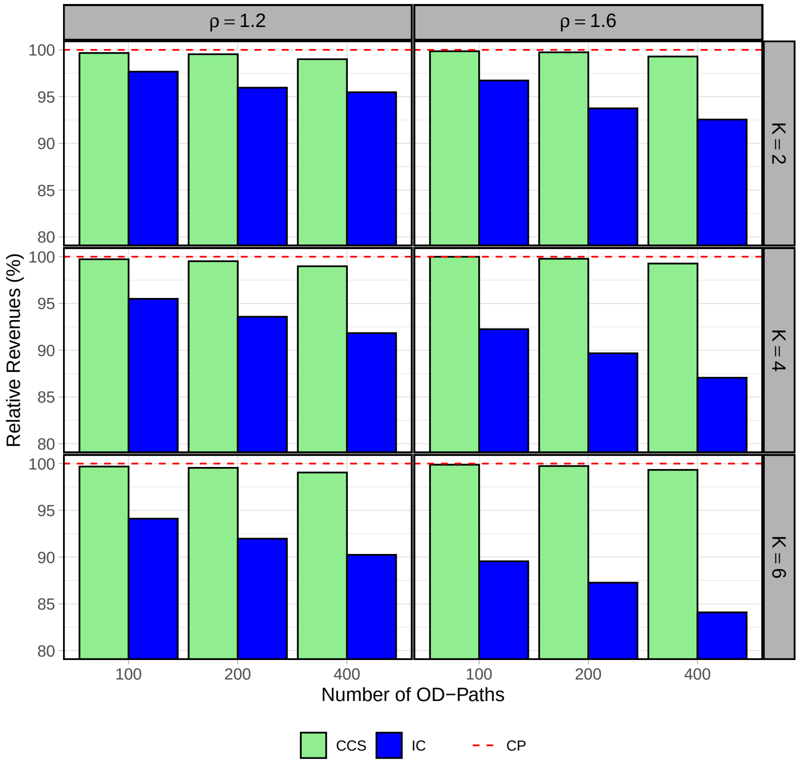
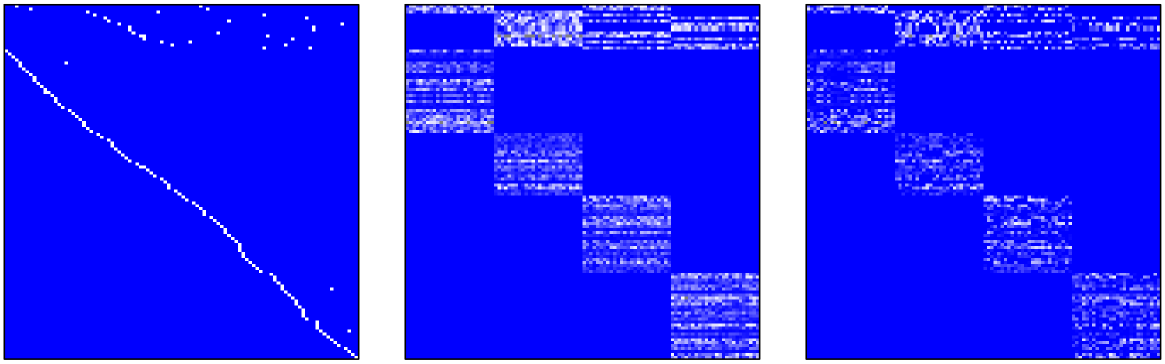


Figure 2: Relative average revenues with respect to CP.

4.3 Computational Efficiency. In the last step, we evaluate the computational efficiency of the data-private model (29)-(34). When we consider the structure of the proposed data-private model, we observe that the matrices in the original model (2)-(5) lose their sparse structure after the reformulation. Take for instance the matrix $\mathbf{A}_k$ and its transformed counterpart $\bar{\mathbf{A}}_k$. An incidence matrix $\mathbf{A}_k$ is sparse whereas $\bar{\mathbf{A}}_k$ is quite dense. This loss of sparsity structure in the overall problem should be expected to cause an increase in the computation time. Indeed, we have observed that whenever the matrices $\bar{\mathbf{A}}_k$ and $\bar{\mathbf{B}}_k$ are obtained by straightforward randomization, then the solution time of the data-private model is considerably longer than the time to solve the original problem (see Section 4 for our actual computation times). Figures 3(a) and 3(b) show the sparsity structure before and after direct random transformation, respectively.

In order to circumvent this loss of sparsity, we try to randomize the matrices in a structured manner so that we can obtain transformed matrices that are as sparse as possible. To this end, we aim at filling in the nonzero entries of the random matrix $\mathbf{D}_k$ in such a way that the multiplication of its components with the components of $\mathbf{A}_k$ and $\mathbf{B}_k$ yields as many zeros as possible. This observation leads to the following mathematical programming model:

$$\text{minimize} \quad \mathbf{1}_m^\intercal(\mathbf{A}_k\mathbf{U})\mathbf{1}_{s_k} + \mathbf{1}_{m_k}^\intercal(\mathbf{B}_k\mathbf{U})\mathbf{1}_{s_k} + \mathbf{1}_{n_k}^\intercal\mathbf{U}\mathbf{1}_{s_k} \tag{35}$$



(a) Original $\mathbf{A}_k$ and $\mathbf{B}_k$ matrix (b) Dense $\bar{\mathbf{A}}_k$ and $\bar{\mathbf{B}}_k$ matrix (c) Sparse $\bar{\mathbf{A}}_k$ and $\bar{\mathbf{B}}_k$ matrix

Figure 3: The sparsity structure of matrices for an example problem from our computational study. The values in each cell of the matrix is between zero (dark blue) and one (bright white). The darker the cell, the closer the value to zero.

$$\begin{aligned} \text{subject to} \quad & \mathbf{U}\mathbf{1}_{s_k} \geq s_k \mathbf{1}_{n_k}, && (36)\\ & \mathbf{1}_{n_k}\mathbf{U} \geq s_k \mathbf{1}_{s_k}, && (37)\\ & \mathbf{U} \text{ is a binary matrix}, && (38) \end{aligned}$$

where the subscript $\bullet$ in $\mathbf{1}_\bullet$ shows the dimension of the vector of ones. The first two terms in (35) are added to obtain as many zeros as possible after multiplying $\mathbf{A}_k$ and $\mathbf{B}_k$ with the binary matrix $\mathbf{U}$. The last term of (35) makes sure that the solution is filled with zeros instead of ones as long as the first two terms are not affected. The constraints (36)-(37) guarantee that we have s_k many ones in each column and row of $\mathbf{U}$. This is a network flow problem satisfying the total unimodularity property. Therefore, it can be solved very efficiently by a standard network simplex algorithm. Moreover, the resulting optimal spanning tree solution $\mathbf{U}^*$ has full rank (Wright, 2000). Then, the last step is to randomize this binary matrix to obtain the desired matrix. Formally, $\mathbf{D}_k^\intercal = \mathbf{U}^* \odot \mathbf{R}$, where $\odot$ stands for the Hadamard product and $\mathbf{R}$ is an $n_k \times s_k$ random matrix. When contrasted against Figure 3(b), Figure 3(c) shows how obtaining the matrix $\mathbf{D}_k$ by solving (35)-(38) changes the sparsity structure of the data-private model.

To understand the effect of transformation, we report the computation times for the original model (2)-(5), the data-private model (29)-(34) and the sparsity induced model. The data-private model results are first given with straightforward randomization, which ends with full matrices. Then, we solve (35)-(38) to obtain sparse matrices. We evaluate the computation times for all network sizes with four parties. Figure 4 presents the average computation times on a semi-logarithmic plot; that is, the values on the vertical axis are scaled by taking their logarithms. The legend shows the original model (CP), the masked model with straightforward randomization (CCS - Dense) and the masked model with sparsity inducing transformations (CCS - Sparse). Our numerical results confirm that this loss causes a significant increase in the computation time. The data-private model with dense matrices takes by far the largest computation time compared to other models. As the network size increases, the solution time of the data-private model also increases. Taking sparsity into consideration for the data-private model does indeed pay off, as the computation time with the sparsity inducing transformations reduces the computation times considerably.

At this point, we should emphasize that the random matrices obtained after solving the mathematical programming model (35)-(38) may not be secure in the sense of Section 3.3. Thus, the gain from maintaining sparsity may come at the cost of a security breach. This happens because we do not have a control on the optimal solution of the model, and hence, it is not easy to quantify the potential leakage (Hong et al., 2018).

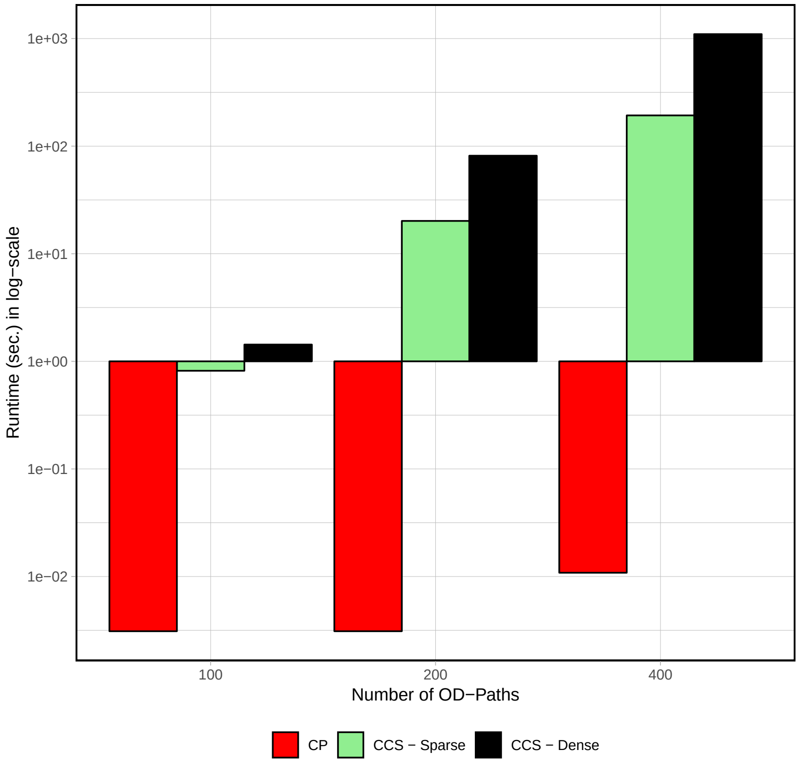


Figure 4: Average computation time ($\rho = 1.2, K = 4$).

We leave this discussion for a future work.

5. Conclusion. We have presented a mathematical model which considers data privacy in collaborative resource management problem when multiple parties share some of the capacities of the network. The proposed approach is based on applying matrix transformations to collaborative network problem. We have shown that the original primal and dual optimal solutions can be derived from the proposed data-private mathematical model. We have also discussed the security of the input data after solving the transformed problem.

We have conducted a simulation study on a network structure of an airline. Our results have illustrated the benefits of the proposed data-private capacity control. We have considered the setting with and without a collaboration, and shown that, with collaboration the revenues for the parties are significantly higher. Therefore, these results offer an economic motivation for the parties to form an alliance. Nevertheless, the privacy comes at a cost. Unlike the sparse structure of the original problem, the data-private model has a dense structure. This loss of sparsity causes a considerable increase in computation times. To overcome this problem, we have provided an approach based on solving a network flow model. We have demonstrated how this approach positively affects the computational effort. We have also cautioned that our approach for maintaining the sparse structure may come at a security leakage cost. An important point that we have left for future study.

Even though we have conducted a simulation study on an airline network, our approach is not limited to airline problems. The proposed approach can be used in different network problems from collaborative supply chains to power networks. For instance, in the logistics sector, sharing network capacities among several companies may increase the utilization of the resources. This high utilization, in turn, increases the competitive advantage of a participating company in terms of higher profit and less environmental impact. Furthermore, our approach can be used for decision-making problems outsourced in a cloud environment. Cloud computing provides computing resources and it is widely used by companies to efficiently solve their large-scale decision making problems. However, one of the main issues is the security and the privacy of the stored data. Our approach can be used to mask input data and solve the problem while ensuring data privacy.

There are other interesting research questions about data privacy in network management. We have presented a transformation-based approach for capacity control. Although we guarantee to obtain the exact optimal solutions for each party, our approach may become vulnerable for potential security breaches especially for small-scale problems. Investigating the effect of small-scale problems on data-privacy is one of our future research directions. Another approach to tackle data privacy in collaborative decision making problems could be using the concept of differential-privacy, where the main idea is to perturb the output with a carefully adjusted noise. Such an approach does lead to approximate solutions but the privacy levels can be quantified and controlled. This approach is also on our agenda for future research.

References

Adenso-Díaz, B., Lozano, S., Garcia-Carbajal, S., and Smith-Miles, K. (2014). Assessing partnership savings in horizontal cooperation by planning linked deliveries. *Transportation Research Part A: Policy and Practice*, 66:268–279.

Agarwal, R. and Ergun, O. (2010). Network design and allocation mechanisms for carrier alliances in liner shipping. *Operations Research*, 58(6):1726–1742.

Albrecht, M. and Stadtler, H. (2015). Coordinating decentralized linear programs by exchange of primal information. *European Journal of Operational Research*, 247(3):788–796.

Amaruchkul, K., Cooper, W. L., and Gupta, D. (2011). A note on air-cargo capacity contracts. *Production and Operations Management*, 20(1):152–162.

Bednarz, A. (2012). *Methods for Two-Party Privacy-Preserving Linear Programming*. PhD thesis, The University of Adelaide, Adelaide, Australia.

Belobaba, P. and Jain, H. (2013). Alliance revenue management in practice: Impacts of bid price sharing and dynamic valuation. *Journal of Revenue and Pricing Management*, 12(6):475–488.

Birbil, S. I., Frenk, J. B. G., Gromicho, J. A. S., and Zhang, S. (2014). A network airline revenue management framework based on decomposition by origins and destinations. *Transportation Science*, 48(3):313–333.

Boyd, E. A. (1998). Airline alliance revenue management. Technical report, Pros Strategic Solutions, Houston, TX.

Braun, C., Chatzikokolakis, K., and Palamidessi, C. (2009). Quantitative notions of leakage for one-try attacks. *Electronic Notes in Theoretical Computer Science*, 249:75–91.

Chun, S. Y., Kleywegt, A. J., and Shapiro, A. (2017). When friends become competitors: The design of resource exchange alliances. *Management Science*, 63(7):2127–2145.

Curry, R. E. (1990). Optimal airline seat allocation with fare classes nested by origins and destinations. *Transportation Science*, 24(3):193–204.

Dantzig, G. B. (1956). Recent advances in linear programming. *Management Science*, 2(2):131–144.

Ding, S. and Kaminsky, P. M. (2020). Centralized and decentralized warehouse logistics collaboration. *Manufacturing & Service Operations Management*, 22(4):812–831.

Dreier, J. and Kerschbaum, F. (2011). Practical privacy-preserving multiparty linear programming based on problem transformation. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 916–924. IEEE.

Du, W. (2001). *A Study of Several Specific Secure Two-Party Computation Problems*. PhD thesis, Purdue University, West Lafayette, Indiana.

Du, W. and Zhan, Z. (2002). A practical approach to solve secure multi-party computation problems. In *Proceedings of the 2002 Workshop on New Security Paradigms*, pages 127–135.

Feng, X. and Zhang, Z. (2007). The rank of a random matrix. *Applied Mathematics and Computation*, 185(1):689–694.

Gansterer, M. and Hartl, R. F. (2016). Request evaluation strategies for carriers in auction-based collaborations. *OR Spectrum*, 38:3–23.

Gansterer, M. and Hartl, R. F. (2018). Collaborative vehicle routing: A survey. *European Journal of Operational Research*, 268(1):1–12.

Gerlach, M., Cleophas, C., and Kliewer, N. (2013). Airline codeshare alliances: Marketing boon and revenue management information systems challenge. *Business & Information Systems Engineering*, 5(3):153–163.

Ghaderyan, D., Pereira, F. L., and Aguiar, A. P. (2021). A fully distributed method for distributed multiagent system in a microgrid. *Energy Reports*, 7:2294–2301.

Goldreich, O. (2009). *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press.

Granot, D. and Sošić, G. (2005). Formation of alliances in internet-based supply exchanges. *Management Science*, 51(1):92–105.

Guo, L. and Wu, X. (2018). Capacity sharing between competitors. *Management Science*, 64(8):3554–3573.

He, Y. and Yin, S. (2015). Joint selling of complementary components under brand and retail competition. *Manufacturing & Service Operations Management*, 17(4):470–479.

Hong, Y. and Vaidya, J. (2014). An inference–proof approach to privacy-preserving horizontally partitioned linear programs. *Optimization Letters*, 8(1):267–277.

Hong, Y., Vaidya, J., Rizzo, N., and Liu, Q. (2018). Privacy-preserving linear programming. In *World Scientific Reference on Innovation Volume 4: Innovation in Information Security*, World Scientific Book Chapters, chapter 4, pages 71–93. World Scientific Publishing Co. Pte. Ltd.

Horn, R. A. and Johnson, C. R. (1991). *Topics in Matrix Analysis*. Cambridge University Press, Cambridge.

Houghtalen, L., Ergun, O., and Sokol, J. (2011). Designing mechanisms for the management of carrier alliances. *Transportation Science*, 45(4):465–482.

Hyndman, K., Kraiselburd, S., and Watson, N. (2013). Aligning capacity decisions in supply chains when demand forecasts are private information: Theory and experiment. *Manufacturing & Service Operations Management*, 15(1):102–117.

Jin, X., Park, K. T., and Kim, K. H. (2019). Storage space sharing among container handling companies. *Transportation Research Part E: Logistics and Transportation Review*, 127:111–131.

Krajewska, M. A. and Kopfer, H. (2006). Collaborating freight forwarding enterprises. *OR Spectrum*, 28(3):301–311.

Kunnumkal, S. and Topaloglu, H. (2010). A new dynamic programming decomposition method for the network revenue management problem with customer choice behavior. *Production and Operations Management*, 19:575–590.

Lai, M., Xue, W., and Hu, Q. (2019). An ascending auction for freight forwarder collaboration in capacity sharing. *Transportation Science*, 53(4):1175–1195.

Laud, P. and Pankova, A. (2013). On the (im) possibility of privately outsourcing linear programming. In *Proceedings of the 2013 ACM Workshop on Cloud Computing Security*, pages 55–64.

Li, J. and Atallah, M. J. (2006). Secure and private collaborative linear programming. In *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 1–8. IEEE.

Li, W., Li, H., and Deng, C. (2013). Privacy-preserving horizontally partitioned linear programs with inequality constraints. *Optimization Letters*, 7(1):137–144.

Mangasarian, O. L. (2011). Privacy-preserving linear programming. *Optimization Letters*, 5(1):165–172.

Mangasarian, O. L. (2012). Privacy-preserving horizontally partitioned linear programs. *Optimization Letters*, 6(3):431–436.

Phillips, R. L. (2005). *Pricing and Revenue Management*. Stanford University Press, Stanford, CA, USA.

Poole, G. and Boullion, T. (1974). A survey on M-matrices. *SIAM review*, 16(4):419–427.

Poundarikapuram, S. and Veeramani, D. (2004). Distributed decision-making in supply chains and private e-marketplaces. *Production and Operations Management*, 13(1):111–121.

Rius-Sorolla, G., Maheut, J., and Estellés-Miguel, S. (2020). Coordination mechanisms with mathematical programming models for decentralized decision-making: a literature review. *Central European Journal of Operations Research*, 28:61–104.

Singh, G. and O'Keefe, C. M. (2016). Decentralised scheduling with confidentiality protection. *Operations Research Letters*, 44(4):514–519.

Speranza, M. G. (2018). Trends in transportation and logistics. *European Journal of Operational Research*, 264(3):830–836.

Talluri, K. T. and van Ryzin, G. J. (2004). *The Theory and Practice of Revenue Management*. Springer, New York, NY.

Toft, T. (2009). Solving linear programs using multiparty computation. In *International Conference on Financial Cryptography and Data Security*, pages 90–107. Springer.

Topaloglu, H. (2012). A duality based approach for network revenue management in airline alliances. *Journal of Revenue Pricing and Management*, 11(5):500–517.

Vaidya, J. (2009a). Privacy-preserving linear programming. In *Proceedings of the 2009 ACM Symposium on Applied Computing*, pages 2002–2007. ACM.

Vaidya, J. (2009b). A secure revised simplex algorithm for privacy-preserving linear programming. In *2009 International Conference on Advanced Information Networking and Applications*, pages 347–354. IEEE.

Wang, C., Ren, K., and Wang, J. (2011). Secure and practical outsourcing of linear programming in cloud computing. In *INFOCOM, 2011 Proceedings IEEE*, pages 820–828. IEEE.

Weeraddana, P. C., Athanasiou, G., Fischione, C., and Baras, J. S. (2013). Per-se privacy preserving solution methods based on optimization. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 206–211.

Wright, C. P. (2014). Decomposing airline alliances: A bid-price approach to revenue management with incomplete information sharing. *Journal of Revenue and Pricing Management*, 13(3):164–182.

Wright, C. P., Groenevelt, H., and Shumsky, R. A. (2010). Dynamic revenue management in airline alliances. *Transportation Science*, 44(1):15–37.

Wright, P. (2000). On minimum spanning trees and determinants. *Mathematics Magazine*, 73(1):21–28.

Zheng, J., Gao, Z., Yang, D., and Sun, Z. (2015). Network design and capacity exchange for liner alliances with fixed and variable container demands. *Transportation Science*, 49(4):886–899.

Appendix A. Suppose for each path or route $s \in \mathcal{S}$ that the objective function $\phi_s(x_s)$ consists of B_s breakpoints (intervals with a length of one) that subdivides the range of x_s. These breakpoints collectively form the set $\mathcal{B}_s$ for $s \in \mathcal{S}$. We introduce the auxiliary variables x_{bs} for $b \in \mathcal{B}_s$, and set

$$x_s = \sum_{b \in \mathcal{B}_s} x_{bs}.$$

Since the length of each interval is one, we have $x_{bs} \leq 1, s \in \mathcal{S},\ b \in \mathcal{B}_s$. If we denote the partial revenues by r_{bs}, then the new model becomes

$$\begin{aligned}
\text{maximize} \quad & \sum_{s \in \mathcal{S}} \sum_{b \in \mathcal{B}_s} r_{bs} x_{bs}, & \\
\text{subject to} \quad & \sum_{s \in \mathcal{S}} \sum_{b \in \mathcal{B}_s} a_{js} x_{bs} \leq c_j, & j \in \mathcal{J}, \\
& \sum_{s \in \mathcal{S}_k} \sum_{b \in \mathcal{B}_s} a_{js} x_{bs} \leq c_j, & j \in \mathcal{J}_k, k \in \mathcal{K}, \\
& 0 \leq x_{bs} \leq 1, & s \in \mathcal{S},\ b \in \mathcal{B}_s.
\end{aligned}$$

Due to the concavity of the objective function, we have $r_{1s} \geq r_{2s} \geq r_{B_s s}$ for $s \in \mathcal{S}$. This structure allows us to partition for $k \in \mathcal{K}$, the decision variables and the objective function parameters as

$$\mathbf{x}_k = [x_{bs} : s \in \mathcal{S}_k, b \in \mathcal{B}_s]^\intercal \text{ and } \mathbf{r}_k = [r_{bs} : s \in \mathcal{S}_k, b \in \mathcal{B}_s]^\intercal,$$

respectively. Again for $k \in \mathcal{K}$, we next define the $m \times n_k$ matrix $\mathbf{A}_k$ with $n_k = \sum_{s \in \mathcal{S}_k} B_s$ and the $m_k \times n_k$ matrix $\mathbf{B}_k$ as

$$\mathbf{A}_k = \begin{bmatrix} \underbrace{a_{js}\ a_{js}\ \cdots\ a_{js}}_{B_s \text{ times}} \end{bmatrix}_{j \in \mathcal{J}, s \in \mathcal{S}_k} \quad \text{and} \quad \mathbf{B}_k = \begin{bmatrix} \underbrace{b_{js}\ b_{js}\ \cdots\ b_{js}}_{B_s \text{ times}} \end{bmatrix}_{j \in \mathcal{J}_k, s \in \mathcal{S}_k}, \tag{39}$$

respectively. Here $b_{js} = 1$, if path s uses one unit from capacity j; otherwise, $b_{js} = 0$. The last step is to introduce the shared and the private capacity vectors as

$$\mathbf{c} = [c_j : j \in \mathcal{J}]^\intercal \text{ and } \mathbf{c}_k = [c_j : j \in \mathcal{J}_k]^\intercal \text{ for all } k \in \mathcal{K},$$

respectively. We are now ready to give our main capacity sharing model with the path-based formulation:

$$\begin{aligned}
Z = \text{maximize} \quad & \sum_{k \in \mathcal{K}} \mathbf{r}_k^\intercal \mathbf{x}_k, & & \\
\text{subject to} \quad & \sum_{k \in \mathcal{K}} \mathbf{A}_k \mathbf{x}_k \leq \mathbf{c}, & & (\boldsymbol{\alpha}) \\
& \mathbf{B}_k \mathbf{x}_k \leq \mathbf{c}_k, & k \in \mathcal{K}, & (\boldsymbol{\alpha}_k) \\
& \mathbf{0} \leq \mathbf{x}_k \leq \mathbf{1}, & k \in \mathcal{K}, &
\end{aligned}$$

where $\mathbf{1}$ and $\mathbf{0}$ stand for the vector of ones and the vector of zeros, respectively.

Appendix B. We have reserved this section for the proofs of our theoretical results, which we have repeated here for clarity of presentation.

Lemma 3.1 *If we denote the primal optimal solution of* (10)-(15) *by* $(\mathbf{z}_k^*, \mathbf{v}_k^*)_{k\in\mathcal{K}}$ *and the dual optimal variables associated with the capacity constrains by* $(\boldsymbol{\beta}^*, \boldsymbol{\beta}_k^*)_{k\in\mathcal{K}}$*, then we have*

$$\begin{aligned} \mathbf{z}_k^* &= \mathbf{x}_k^* + \boldsymbol{\eta}_k, \quad k \in \mathcal{K}, \\ \boldsymbol{\beta}^* &= \boldsymbol{\alpha}^*, \\ \boldsymbol{\beta}_k^* &= \boldsymbol{\alpha}_k^* + \boldsymbol{\xi}_k, \quad k \in \mathcal{K}. \end{aligned}$$

Proof. We first define $(\boldsymbol{\lambda}_k)_{k\in\mathcal{K}}$ as the dual vector corresponding to the upper bound constraints (5). Then, the dual of (2)–(5) becomes

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^\intercal\boldsymbol{\alpha} + \sum_{k\in\mathcal{K}} \mathbf{c}_k^\intercal\boldsymbol{\alpha}_k + \sum_{k\in\mathcal{K}} \mathbf{1}^\intercal\boldsymbol{\lambda}_k & & (40)\\ \text{subject to} \quad & \mathbf{A}_k^\intercal\boldsymbol{\alpha} + \mathbf{B}_k^\intercal\boldsymbol{\alpha}_k + \boldsymbol{\lambda}_k \geq \mathbf{r}_k, & k \in \mathcal{K}, \quad & (41)\\ & \boldsymbol{\alpha}, \boldsymbol{\alpha}_k, \boldsymbol{\lambda}_k \geq \mathbf{0}, & k \in \mathcal{K}. \quad & (42) \end{aligned}$$

Likewise, we also define $(\boldsymbol{\nu}_k)_{k\in\mathcal{K}}$ and $(\boldsymbol{\theta}_k)_{k\in\mathcal{K}}$ as the dual vectors corresponding to the constraints (13) and (14), respectively. Then, the dual of (10)-(15) is obtained as

$$\begin{aligned} \text{minimize} \quad & (\mathbf{c} + \sum_{k\in\mathcal{K}} \mathbf{A}_k\boldsymbol{\eta}_k)^\intercal\boldsymbol{\beta} + \sum_{k\in\mathcal{K}} (\mathbf{c}_k + \mathbf{B}_k\boldsymbol{\eta}_k)^\intercal\boldsymbol{\beta}_k + \sum_{k\in\mathcal{K}} (\mathbf{1} + \boldsymbol{\eta}_k)^\intercal\boldsymbol{\nu}_k - \sum_{k\in\mathcal{K}} \boldsymbol{\eta}_k^\intercal\boldsymbol{\theta}_k & & (43)\\ \text{subject to} \quad & \mathbf{A}_k^\intercal\boldsymbol{\beta} + \mathbf{B}_k^\intercal\boldsymbol{\beta}_k + \boldsymbol{\nu}_k - \boldsymbol{\theta}_k = \mathbf{r}_k + \mathbf{B}_k^\intercal\boldsymbol{\xi}_k, & k \in \mathcal{K}, \quad & (44)\\ & \boldsymbol{\beta}_k \geq \boldsymbol{\xi}_k, & k \in \mathcal{K}, \quad & (45)\\ & \boldsymbol{\beta}, \boldsymbol{\nu}_k, \boldsymbol{\theta}_k \geq \mathbf{0}, & k \in \mathcal{K}. \quad & (46) \end{aligned}$$

Suppose that $(\mathbf{z}_k^*, \mathbf{v}_k^*)_{k\in\mathcal{K}}$ and $(\boldsymbol{\beta}^*, \boldsymbol{\beta}_k^*, \boldsymbol{\nu}_k^*, \boldsymbol{\theta}_k^*)_{k\in\mathcal{K}}$ are the primal and the dual optimal solutions for (10)–(15), respectively. Let

$$\mathbf{z}_k^* = \mathbf{x}_k^* + \boldsymbol{\eta}_k, \; k \in \mathcal{K}. \tag{47}$$

We plug this particular vector into (10)-(15) and observe that $\mathbf{v}_k \geq \mathbf{0}$, $k \in \mathcal{K}$. Thus, $(\mathbf{x}_k^*)_{k\in\mathcal{K}}$ is a feasible solution for (2)–(5). Next we plug

$$\begin{aligned} \boldsymbol{\beta}^* &= \boldsymbol{\alpha}^*, \\ \boldsymbol{\beta}_k^* &= \boldsymbol{\alpha}_k^* + \boldsymbol{\xi}_k, \quad k \in \mathcal{K}, \\ \boldsymbol{\nu}_k^* &= \boldsymbol{\lambda}_k^*, \qquad\quad k \in \mathcal{K}, \end{aligned} \tag{48}$$

into (43)–(46) and note that $\boldsymbol{\theta}_k \geq \mathbf{0}$, $k \in \mathcal{K}$. This shows that $(\boldsymbol{\alpha}^*, \boldsymbol{\alpha}_k^*, \boldsymbol{\lambda}_k^*)_{k\in\mathcal{K}}$ is a feasible solution for (40)–(42). Consequently, we have feasible solutions for both the primal problem and the dual problem. When we consider the equalities in (10)-(15) and (43)–(46), we obtain for $k \in \mathcal{K}$ that

$$\begin{aligned} \mathbf{v}_k^* &= \mathbf{c}_k + \mathbf{B}_k\boldsymbol{\eta}_k - \mathbf{B}_k\mathbf{z}_k^* = \mathbf{c}_k - \mathbf{B}_k\mathbf{x}_k^*, \\ \boldsymbol{\theta}_k^* &= \mathbf{A}_k^\intercal\boldsymbol{\beta}^* + \mathbf{B}_k^\intercal\boldsymbol{\beta}_k^* + \boldsymbol{\nu}_k^* - \mathbf{r}_k - \mathbf{B}_k^\intercal\boldsymbol{\xi}_k = \mathbf{A}_k^\intercal\boldsymbol{\alpha}^* + \mathbf{B}_k^\intercal\boldsymbol{\alpha}_k^* + \boldsymbol{\lambda}_k^* - \mathbf{r}_k. \end{aligned} \tag{49}$$

Recall that the strong duality of linear programming implies

$$\sum_{k\in\mathcal{K}} (\mathbf{r}_k + \mathbf{B}_k^\intercal\boldsymbol{\xi}_k)^\intercal\mathbf{z}_k^* + \sum_{k\in\mathcal{K}} \boldsymbol{\xi}_k^\intercal\mathbf{v}_k^* = (\mathbf{c} + \sum_{k\in\mathcal{K}} \mathbf{A}_k\boldsymbol{\eta}_k)^\intercal\boldsymbol{\beta}^* + \sum_{k\in\mathcal{K}} (\mathbf{c}_k + \mathbf{B}_k\boldsymbol{\eta}_k)^\intercal\boldsymbol{\beta}_k^* + \sum_{k\in\mathcal{K}} (\mathbf{1} + \boldsymbol{\eta}_k)^\intercal\boldsymbol{\nu}_k^* - \sum_{k\in\mathcal{K}} \boldsymbol{\eta}_k^\intercal\boldsymbol{\theta}_k^*.$$

Rewriting this equality with (47), (48) and (49) shows that

$$\sum_{k\in\mathcal{K}} \mathbf{r}_k^\intercal \mathbf{x}_k^* = \mathbf{c}^\intercal \boldsymbol{\alpha}^* + \sum_{k\in\mathcal{K}} \mathbf{c}_k^\intercal \boldsymbol{\alpha}_k^* + \sum_{k\in\mathcal{K}} \mathbf{1}^\intercal \boldsymbol{\lambda}_k^*.$$

This establishes that $(\mathbf{x}_k)_{k\in\mathcal{K}}$ and $(\boldsymbol{\alpha}^*, \boldsymbol{\alpha}_k^*, \boldsymbol{\lambda}_k^*)_{k\in\mathcal{K}}$ are the primal and dual optimal solutions for (2)–(5), respectively. The desired equalities in the hypothesis follow from our construction. □

Theorem 3.1 *Let $(\mathbf{u}_k^*, \mathbf{w}_k^*)_{k\in\mathcal{K}}$ and $(\boldsymbol{\gamma}^*, \boldsymbol{\gamma}_k^*)_{k\in\mathcal{K}}$ be the primal and dual optimal solutions of (29)-(34). Using again the primal and dual optimal solutions, $(\mathbf{x}_k^*)_{k\in\mathcal{K}}$ and $(\boldsymbol{\alpha}^*, \boldsymbol{\alpha}_k^*)_{k\in\mathcal{K}}$ of the original problem (2)–(5), we obtain*

$$\begin{array}{rll} Z = \bar{Z} - \sum_{k\in\mathcal{K}} \mathbf{r}_k^\intercal \boldsymbol{\eta}_k - \sum_{k\in\mathcal{K}} (\mathbf{c}_k + \mathbf{B}_k\boldsymbol{\eta}_k)^\intercal \boldsymbol{\xi}_k, & & \\ \mathbf{x}_k^* = \mathbf{D}_k^\intercal \mathbf{u}_k^* - \boldsymbol{\eta}_k, & & k\in\mathcal{K}, \\ \boldsymbol{\alpha}^* = \boldsymbol{\gamma}^*, & & \\ \boldsymbol{\alpha}_k^* = \mathbf{F}_k^\intercal \boldsymbol{\gamma}_k^* - \boldsymbol{\xi}_k, & & k\in\mathcal{K}. \end{array}$$

Proof. To obtain the linear programming model (29)-(34), we apply for $k\in\mathcal{K}$ the change of variables $\mathbf{D}_k^\intercal \mathbf{u}_k = \mathbf{z}_k$ and $\mathbf{E}_k^\intercal \mathbf{w}_k = \mathbf{v}_k$ to the model (10)-(15). Likewise, multiplying both sides of the equality constraints (12) with $\mathbf{F}_k$ leads for $k\in\mathcal{K}$, to the change of variables $\boldsymbol{\beta}_k = \mathbf{F}_k^\intercal \boldsymbol{\gamma}_k$. Note that both sides of the constraints (13)-(15) are multiplied by M-matrices, and hence, feasibility is not affected. Using next Lemma 3.1 implies

$$\begin{array}{ll} \mathbf{x}_k^* = \mathbf{z}_k^* - \boldsymbol{\eta}_k = \mathbf{D}_k^\intercal \mathbf{u}_k^* - \boldsymbol{\eta}_k, & k\in\mathcal{K}, \\ \boldsymbol{\alpha}^* = \boldsymbol{\beta}^* = \boldsymbol{\gamma}^*, & \\ \boldsymbol{\alpha}_k^* = \boldsymbol{\beta}_k^* - \boldsymbol{\xi}_k = \mathbf{F}_k^\intercal \boldsymbol{\gamma}_k^* - \boldsymbol{\xi}_k, & k\in\mathcal{K}. \end{array}$$

Mangasarian (2011, Proposition 1) has shown that the optimal objective function values of (10)-(15) and (29)-(34) are the same. Recall from the proof of Lemma 3.1 that (10)-(15) is obtained from (2)–(5) by applying for $k\in\mathcal{K}$, the transformations $\mathbf{z}_k = \mathbf{x}_k + \boldsymbol{\eta}_k$ and $\boldsymbol{\beta}_k = \boldsymbol{\alpha}_k + \boldsymbol{\xi}_k$. Using the first transformation, the constant term $\sum_{k\in\mathcal{K}} \mathbf{r}_k^\intercal \boldsymbol{\eta}_k$ is subtracted from the objective function. Moreover, the same transformation also alters the right-hand-side of (4) as $\mathbf{c}_k + \mathbf{B}_k\boldsymbol{\eta}_k$, $k\in\mathcal{K}$. The second transformation with this new right-hand-side subtracts additionally the constant term $\sum_{k\in\mathcal{K}} (\mathbf{c}_k + \mathbf{B}_k\boldsymbol{\eta}_k)^\intercal \boldsymbol{\xi}_k$ from the objective function. Adding both constant terms establishes the required equality:

$$\bar{Z} = Z + \sum_{k\in\mathcal{K}} \mathbf{r}_k^\intercal \boldsymbol{\eta}_k + \sum_{k\in\mathcal{K}} (\mathbf{c}_k + \mathbf{B}_k\boldsymbol{\eta}_k)^\intercal \boldsymbol{\xi}_k.$$

This completes the proof. □

Lemma 3.2 *Suppose for $k\in\mathcal{K}$ that $1\le m < n_k \le s_k$, $1 < m_k \le t_k$, and both $\mathbf{A}_k$ and $\mathbf{B}_k$ have full rank. Even if all private information of party $k\in\mathcal{K}$ are known, then finding any one of $\mathbf{D}_k$, $\mathbf{F}_k$, $\boldsymbol{\eta}_k$ or $\boldsymbol{\xi}_k$ requires obtaining a particular solution to a system of linear equations with infinitely many solutions.*

Proof. Using (28), we first check the relations that involve $\mathbf{D}_k$ as the only unknown. This leaves us with $\bar{\mathbf{A}}_k = \mathbf{A}_k\mathbf{D}_k^\intercal$, where $\mathbf{A}_k$ is $m\times n_k$ matrix. Since $m < n_k$ and $\operatorname{rank}(A_k) = m$, this system has infinitely many solutions. In all other relations listed in (28), $\mathbf{D}_k$ is placed along with another private random matrix. However, we note that when $n_k = s_k$, we have

$$\bar{\mathbf{1}}_k - \bar{\mathbf{G}}_k\bar{\mathbf{H}}_k^{-1}\bar{\boldsymbol{\eta}}_k = \mathbf{G}_k\mathbf{1} + \mathbf{G}_k\boldsymbol{\eta}_k - \bar{\mathbf{G}}_k\bar{\mathbf{H}}_k^{-1}\bar{\boldsymbol{\eta}}_k = \mathbf{G}_k\mathbf{1}.$$

When $n_k = 1$, the random matrix $\mathbf{G}_k$, and consequently, $\mathbf{D}_k$ can be obtained. However, we have assumed that $n_k > 1$ leading to an underdetermined system. Thus, $\mathbf{D}_k$ cannot be obtained without solving a system with infinitely many solutions. In a similar vein, $\boldsymbol{\eta}_k$ appears only in $\tilde{\boldsymbol{\eta}}_k = \mathbf{A}_k\boldsymbol{\eta}_k$ as the sole unknown, but again this system has infinitely many solution for $m < n_k$. Matrices $\mathbf{F}_k$ and $\boldsymbol{\xi}_k$ do not directly appear in any one of the equations without being multiplied with another random matrix. Again we note that when $n_k = s_k$, we have

$$\bar{\mathbf{c}}_k - \bar{\mathbf{B}}_k\bar{\mathbf{H}}_k^{-1}\bar{\boldsymbol{\eta}}_k = \mathbf{F}_k\mathbf{c}_k + \mathbf{F}_k\mathbf{B}_k\boldsymbol{\eta}_k - \bar{\mathbf{B}}_k\bar{\mathbf{H}}_k^{-1}\bar{\boldsymbol{\eta}}_k = \mathbf{F}_k\mathbf{c}_k.$$

This system is also undetermined, since $m_k > 1$. Finally, when $m_k = t_k$, we have

$$\bar{\mathbf{r}}_k - \bar{\mathbf{B}}_k^{\intercal}\bar{\mathbf{F}}_k^{-\intercal}\bar{\boldsymbol{\xi}}_k^{\intercal} = \mathbf{D}_k\mathbf{r}_k + \mathbf{D}_k\mathbf{B}_k^{\intercal}\boldsymbol{\xi}_k - \bar{\mathbf{B}}_k^{\intercal}\bar{\mathbf{F}}_k^{-\intercal}\bar{\boldsymbol{\xi}}_k^{\intercal} = \mathbf{D}_k\mathbf{r}_k.$$

Given $1 < n_k \leq s_k$, this last system has infinitely many solutions as well. □