

# Managing Relationships Between Restaurants and Food Delivery Platforms: Conflict, Contracts, and Coordination

Pnina Feldman, Andrew E. Frazelle, and Robert Swinney\*

February, 2022

Forthcoming in *Management Science*

## Abstract

Restaurant delivery platforms collect customer orders via the internet, transmit them to restaurants, and deliver the orders to customers. They provide value to restaurants by expanding their markets, but critics claim they destroy restaurant profits by taking a percentage of revenues and generating congestion that negatively impacts dine-in customers. We consider these tensions using a model of a restaurant as a congested service system. We find that the predominant industry contract, in which the platform takes a percentage cut of each delivery order (a “commission”), fails to coordinate the system because the platform does not internalize its effect on dine-in revenues; this leads to prices that are too low, reducing the restaurant’s margins and leaving money on the table for both firms. Two commonly proposed remedies to this problem (commission caps and allowing the restaurant to set a price floor on the platform) can increase restaurant revenue but do not solve the coordination issue. We thus propose an alternative, practical coordinating contract that is a variation of the current industry standard: for each delivery order, the platform pays the restaurant a percentage revenue share *and* a fixed fee. We show that this contract, appropriately designed, coordinates the system, protects restaurant margins by ensuring a lower bound on its revenue per delivery order, and allocates revenue between the restaurant and the platform with a high degree of flexibility.

**Keywords:** on-demand services, delivery platforms, supply chain coordination

## 1 Introduction

Delivery platforms such as Grubhub, Seamless, DoorDash, Postmates, and Uber Eats are a large and rapidly growing part of the restaurant industry. These platforms are third-party service providers connecting customers and restaurants: they typically combine a convenient and easy-to-use web-based ordering system with the capability to physically deliver orders to customers. For restaurants, delivery platforms generate value by allowing them to enter the delivery market without building their own ordering platform or investing in their own delivery drivers, expanding their customer base with minimal up-front investment or fixed operating costs. Because of this, platforms often pitch themselves to restaurants as a source of “incremental” revenue, i.e., additional profit on top of their existing dine-in revenue (Meyersohn, 2018).

Despite their seeming advantages, delivery platforms are controversial with restaurant owners (Houck, 2017; Dunn, 2018; Meyersohn, 2018). Among the potential pitfalls of delivery platforms, two issues in

---

\*Feldman: Questrom School of Business, Boston University, 595 Commonwealth Avenue, Boston, MA 02215, pninaf@bu.edu. Frazelle: Jindal School of Management, The University of Texas at Dallas, 800 W Campbell Road, Richardson, TX 75080, andrew.frazelle@utdallas.edu. Swinney: Fuqua School of Business, Duke University, 100 Fuqua Drive, Durham, NC 27708, robert.swinney@duke.edu. The authors thank the Department Editor, the Associate Editor, and three anonymous reviewers for helpful comments. The authors are grateful to one anonymous reviewer in particular for helpful suggestions that led to our identification of a simple and practically implementable coordinating contract. Additionally, the authors are grateful to seminar participants at Baruch College, Boston College, the University of Texas at Austin, and the University of Utah, as well as participants at multiple conference presentations.

particular stand out. The first derives from the fact that the standard contractual relationship offered by most platforms is one of simple revenue sharing: revenue on each order is split between the platform and restaurant according to some pre-negotiated rate. The platform’s share of the revenue, also called a commission, is typically around 15-30% (Restaurant Engine, 2021), leaving the restaurant with only 70-85% of its normal revenue on each item sold.<sup>1</sup> Given the already thin margins in the restaurant industry, which average 4-5% or less (Lunden, 2020), relinquishing any share to a third party can drastically reduce—or eliminate—the profitability of an order.

The second key issue is that a large volume of delivery orders may place a strain on restaurant operations, adding complexity for staff and deteriorating service for dine-in customers. Restaurant staff are faced with the daunting task of juggling orders from multiple customer streams, such as dine-in, take-out, and a stream for each delivery platform that the restaurant supports. Incremental orders coming from the platform can place greater demand on the kitchen—a shared resource that serves both dine-in and delivery customers—and can lead to increased errors, longer wait times for customers, and an overall worse impression of service (Buell, 2017; Houck, 2017). In the long-run, this can harm the restaurant’s reputation and cause even loyal dine-in customers to switch to delivery (where, as noted above, they generate less profit for the restaurant) or simply stop patronizing the restaurant altogether.

Thus, incremental orders generated by the delivery platform not only earn a lower profit rate than orders generated by dine-in customers, they may also deteriorate service for dine-in customers, leading to fewer (profitable) dine-in orders and more (unprofitable) delivery orders; see the case study by Buell (2017) for a vivid example. This illustrates a key source of conflict in the relationship between platforms and restaurants, one that can lead to a downward spiral for restaurants consisting of both increasing demand and decreasing profit margins, causing some to limit, or even eliminate, their partnerships with delivery platforms (Houck, 2017). As the owners of two popular New York eateries put it, “we know for a fact that as delivery increases, our profitability decreases,” and as a result, “sometimes it seems like we’re making food to make Seamless profitable” (Dunn, 2018).

In this paper, we investigate precisely these issues. Our chief goal is to understand how to resolve this potential source of conflict between restaurants and platforms via the use of appropriate contracts, giving customers a service that they value while avoiding the negative effects described above. To accomplish this, we model a restaurant as a congested service system. The restaurant serves two separate channels of demand: dine-in customers and delivery customers. The restaurant has full power over the price in the dine-in channel, while the platform determines the price in the delivery channel. Reflecting the issues described above, the platform keeps a share of the delivery revenue, and incremental orders from the delivery channel generate congestion that is costly for dine-in customers.

Using this model, we show that the contract that currently prevails in the industry ignores the negative

---

<sup>1</sup>Precise terms of contracts between restaurants and platforms are not usually publicly disclosed; however, Restaurant Engine (2021) describes typical commission rates around 30% for each of the four major US platforms (Grubhub and its subsidiary Seamless, DoorDash, Postmates, and UberEats), although this can vary depending on options chosen by the restaurant (e.g., how prominently to display the restaurant in search results) as well as the restaurant’s negotiating power.

interaction between channels and gives both the platform and the restaurant incentive to set their price too low. As a result, joint revenue is not maximized—the system is uncoordinated—which could cause the relationship between platform and restaurant to break down. We also show that common attempts to remedy this problem, commission caps and price floors for the platform, may help the restaurant somewhat, but cannot coordinate the system, serving as a mere “band-aid” on an inadequate contract that fails to fix the underlying structural issues with simple revenue sharing.

Our central contribution is to show that a modification of the current industry standard contract, which we call generalized revenue sharing, achieves coordination with a simple and practically implementable structure. Under this contract, for each delivery order, the platform pays the restaurant both a percentage revenue share *and* a fixed fee that is independent of the delivery price. Such a contract achieves a wide (but not total) range of allocations of the delivery revenue between the parties, and ensures that for each delivery order, the restaurant receives some minimum revenue level which, under certain conditions, is at least as high as the restaurant’s revenue on each dine-in order, protecting restaurant margins.

Our results thus illustrate that the popular discourse around platform-restaurant relationships, which is mostly centered on the idea that revenue sharing contracts are damaging because the commissions chosen by platforms are too high, may be focused on the wrong lever: the *structural form* of simple revenue sharing, the industry standard contract, is deficient. However, our proposed contract with a fixed fee and a percentage revenue share offers a simple and practically implementable way to alleviate a key source of conflict between restaurants and platforms, maximizing aggregate revenue and coordinating the system.

## 2 Related Literature

The literature on food delivery platforms is nascent but growing rapidly. A number of recent papers study operational issues faced by delivery platforms (Mao *et al.* , 2019; Oh *et al.* , 2019; Liu *et al.* , 2020; Li & Wang, 2021b; Niu *et al.* , 2021). The closest works to ours in this stream are Baron *et al.* (2019) and Chen *et al.* (2020). Baron *et al.* (2019) consider an omnichannel retail context, similar to ours, in which one channel generates a negative externality on the other; however, they do not consider a decentralized system and instead, they have a single firm that controls multiple channels. Hence, the issue of incentive conflict and miscoordination between different firms is not addressed, nor do they examine mechanisms to mitigate this (commission caps, price floors, and generalized revenue sharing) as we do.

Chen *et al.* (2020) study contracts between restaurants and delivery platforms and, like us, they find that simple revenue sharing fails to coordinate. Our models differ, however, in their assumptions about pricing power and the form of the coordinating contract. They assume that the menu price is the same in both the dine-in and delivery channels; we make no such assumption. They also find that a simple revenue sharing contract with a “price ceiling” on the delivery menu price—which, because of the identical price assumption, amounts to a constraint on the restaurant’s dine-in price as well—coordinates the system. We propose a

different coordinating contract that places no constraints on the restaurant’s dine-in operations and involves only adjustments on payments made from the platform to the restaurant for delivery orders, which may be more appealing to some restaurateurs.

Our work is also related to the broader supply chain contracting literature, in particular work on revenue sharing contracts. Cachon & Lariviere (2005), for example, show that a revenue sharing contract where a retailer shares some of its revenue with its supplier can coordinate the supply chain and assign arbitrary fractions of revenue to each party. See Cachon (2003) and Lariviere (2015) for reviews and perspectives on the supply chain contracting literature. Our work differs from this literature because of the existence of dual sales channels—the delivery channel and the dine-in channel—and the interaction between them. As a result, while the form of our proposed coordinating contract bears some structural similarity to those proposed in this literature (e.g., Cachon & Lariviere 2005 also consider a revenue sharing contract with both a revenue share and a linear wholesale price, analogous to our fixed fee), the contract parameters play a different role and must be set in a different way in our setting; more on this distinction will be discussed in §5.

Our work is also related to models of supplier encroachment in the broader supply chain literature. Encroachment refers to a supplier, with an existing retail partner, who supplements his indirect (retail) sales with direct-to-consumer sales, and work in this literature focuses on when it is profitable to add a direct channel and how to induce the retailer to accept this. Hence, the “status quo” in encroachment models is typically the opposite of ours: in our case, the supplier (the restaurant) can always sell direct to consumers (i.e., via dine-in), and the question is whether to add the indirect (delivery) channel and how to structure the contract such that the restaurant accepts this.

Many papers in this literature do not consider coordination and focus on the impact of a supplier adding a direct sales channel on firm and supply chain profit (e.g., Arya *et al.* , 2007; Li *et al.* , 2013; Ha *et al.* , 2015). Some do propose coordinating contracts (Tsay & Agrawal, 2004; Boyaci, 2005; Cai, 2010; Xu *et al.* , 2014; Li *et al.* , 2015; Yang *et al.* , 2018); however, these proposed contracts are impractical in our setting, requiring complicated non-linear pricing schemes, sharing of direct sales channel revenue with the retailer (i.e., dine-in revenue with the platform), or removing pricing power from either the supplier or retailer. By contrast, our proposed coordinating contract requires only simple modifications to the form of the current industry standard contract, and is linear in the order volume. Hence, a key contribution of our work is showing that a simple and practically implementable contract can coordinate under the specific type of dual channel structure and interactions between channels representative of the restaurant delivery industry.

## 3 Model

### 3.1 The Restaurant and the Platform

A restaurant sells a single identical item via two channels: a *delivery channel* that serves customers via a third-party platform, and a *dine-in* channel that serves customers directly at the restaurant. The restaurant

has full control over the price in the dine-in channel,  $p$ , and sets that price to maximize its revenue, all of which goes directly to the restaurant. The platform has full control over the price in the delivery channel,  $\theta$ , and sets that price to maximize its revenue. The price  $\theta$  is the “all-in” price for delivery channel customers, including the price of the food as well as any service or delivery fees.<sup>2</sup>

Motivated by practice at most restaurants, both channels are served via a shared kitchen; hence, demand from both channels generates congestion in this shared resource, which can lead to delays and poor service as described in the introduction. We ignore congestion or delays generated by either the dining room (for the dine-in channel) or delivery process (for the delivery channel). Each of these sources of congestion only affects a single channel at a time, and therefore has straightforward consequences on system performance; we focus instead on the more interesting shared step in the process that affects both channels simultaneously, i.e., the kitchen. We also assume that the capacity of the kitchen is fixed and exogenous because, in practice, during peak hours (i.e., lunch and dinner) restaurants are often fully staffed and kitchen capacity is limited by physical space; in the short- or medium-term, it will be difficult to expand capacity further. Moreover, even if staffing is a bottleneck in the kitchen, hiring quality kitchen staff at restaurants can be costly, slow, and challenging (Deliso, 2021). Lastly, as we will show, the delivery channel reduces the restaurant’s margins; it can be difficult to justify capacity expansion as margins deteriorate; a first order concern will be to coordinate the system and protect the restaurant’s margins, after which capacity expansion may be considered.

### 3.2 Customer Segments

The customer population has total size normalized to one and consists of two customer types. *Dine-in* customers only consider using the dine-in restaurant channel. These customers comprise a fraction  $\alpha$  of the total market, and are denoted by the subscript  $R$  for “restaurant” customers. *Delivery* customers only consider using the delivery channel. These customers comprise a fraction  $1 - \alpha$  of the total market, and are denoted by the subscript  $H$  for “home,” since their consumption (usually) occurs at their homes. We assume  $0 < \alpha < 1$  to avoid trivial cases where there is only one customer type and hence all traffic flows through a single channel. Upon arrival, customers choose whether to purchase or not, and do not consider switching between channels. Customer types are thus exogenous and derive from an earlier stage customer decision process, reflective of the fact that customer preferences for dine-in or delivery often result from factors outside of our model (e.g., a customer may be ill or tired and strongly prefer delivery, or may be planning a special evening out and strongly prefer to dine-in at the restaurant). Both dine-in and delivery customers are heterogeneous in their valuation for service on their respective channels. Dine-in customers’

---

<sup>2</sup>In practice, the delivery channel price can be set either entirely by the platform or by a combination of the restaurant and the platform (e.g., the restaurant may set the menu price of the food while the platform may set service and delivery fees). For our base model, we consider the former scenario; this reflects the fact that, even if the restaurant sets the menu price of the food, the platform has the power to both add fees and offer discounts and promotions that could effectively give it full pricing power. For example, if the restaurant sets a price of \$10 for its food, the platform can set a service or delivery fee to raise the final price above \$10, or can offer a promotional discount (e.g., 25% off, \$5 off, free delivery, etc.) to lower the final price below \$10. In §4.3, we consider the latter scenario, where the restaurant sets the price of the food in the delivery channel, effectively establishing a “price floor” on top of which the platform may raise the price further by adding service and delivery fees.

valuations are uniformly distributed on  $[0, \bar{v}_R]$ , with  $\bar{v}_R > 0$ , CDF  $F_R$ , and complementary CDF  $\bar{F}_R$  (i.e.,  $\bar{F}_R(\cdot) = 1 - F_R(\cdot)$ ). Delivery customers' valuations are uniformly distributed on  $[0, \bar{v}_H]$ , with  $\bar{v}_H > 0$ , CDF  $F_H$ , and complementary CDF  $\bar{F}_H$ .

As described above, greater demand placed on the shared resource of the kitchen can increase delays experienced by customers in both channels. Specifically, customers are assumed to experience a linear congestion cost proportional to the total volume of customers who purchase across both channels. Let  $c_t$  be the marginal waiting cost for customers of type  $t \in \{H, R\}$ , and suppose that the total mass of customers who purchase equals  $x$ . The congestion cost incurred by customers is equal to  $c_t x$ . A linear cost function captures the fact that customer utility is decreasing in the volume of customers who use the resource, while still maintaining expositional clarity. It is appropriate if, for example, customers arrive faster than they can be processed by the kitchen, a reasonable assumption during peak hours for restaurants, forming a queue that is proportional in length to the number of arrivals.<sup>3</sup> However, in §6 we extend our results to the case of general convex waiting costs.

We assume that dine-in customers should be more sensitive to congestion than delivery customers, since delivery customers wait for their food in relatively comfortable locations (e.g., their homes or hotel rooms), while dine-in customers wait in less comfortable locations (e.g., the restaurant's waiting area or on the street). Moreover, most delivery platforms allow customers to preorder food for arrival at a specific time, allowing them to avoid (or mostly avoid) waits altogether. Hence, delivery customers have *lower* marginal waiting costs than dine-in customers, i.e.,  $c_H < c_R$ . For tractability, in our base model, we assume dine-in customers incur a strictly positive marginal cost ( $c_R > 0$ ), while delivery customers incur zero cost ( $c_H = 0$ ). In §6, we discuss positive marginal waiting costs for both segments, i.e.,  $0 < c_H < c_R$ .

Let  $\gamma_R$  be the fraction of dine-in customers who purchase and let  $\gamma_H$  be the fraction of delivery customers who purchase. Given  $\alpha$ ,  $\gamma_R$ , and  $\gamma_H$ , the dine-in volume is  $\alpha\gamma_R$ , and the delivery volume is  $(1 - \alpha)\gamma_H$ . Hence, the total number of customers who purchase is  $\alpha\gamma_R + (1 - \alpha)\gamma_H$ , and the congestion cost incurred by dine-in customers is  $c_R(\alpha\gamma_R + (1 - \alpha)\gamma_H)$ . The purchase utility of a dine-in customer with valuation  $v_R$  is  $U_R(p; v_R, \gamma_R, \gamma_H) = v_R - p - c_R(\alpha\gamma_R + (1 - \alpha)\gamma_H)$ , while the purchase utility of a delivery customer with valuation  $v_H$  is  $U_H(\theta; v_H) = v_H - \theta$ . Customers from both segments are assumed to purchase if they have non-negative utility. The sequence of events is as follows. First, the restaurant publicly announces the dine-in price  $p$ . Then, the platform announces the delivery price  $\theta$ . Finally, customers in both channels make their purchasing decisions simultaneously.

---

<sup>3</sup>To see this, suppose  $\lambda$  customers arrive uniformly over 1 hour, and the kitchen is capable of deterministically processing  $\mu < \lambda$  customers per hour. Then, by the end of the 1 hour arrival window,  $L = \lambda - \mu$  customers have accumulated in a queue. On average,  $L/2 = (\lambda - \mu)/2$  customers are waiting in this queue. From Little's Law, since customers are departing the system at rate  $\mu$ , the average wait time in the queue is  $W = L/\mu = (\lambda/\mu - 1)/2$  hours. The average waiting cost for a type  $t$  customer is thus  $c_t W = c_t(\lambda/\mu - 1)/2$ , i.e., a linear function of the total volume of customers who seek service,  $\lambda$ .

### 3.3 Centralized System

We next analyze the centralized system, in which a single firm sets both the delivery and dine-in price to maximize total revenue across both channels. This will serve as a benchmark for comparison in our subsequent analysis. The total revenue for prices  $p$  and  $\theta$  and purchase fractions  $\gamma_R$  and  $\gamma_H$  is  $Z(p, \theta) = p\alpha\gamma_R + \theta(1-\alpha)\gamma_H$ , where the dine-in revenue is  $p\alpha\gamma_R$  and the delivery revenue is  $\theta(1-\alpha)\gamma_H$ . Lemmas A.1 and A.2 in §A of the online supplement derive expressions for the equilibrium purchase fractions of dine-in and delivery customers given prices in each channel. We can substitute these values into  $Z(p, \theta)$  to give

$$Z(p, \theta) = p\alpha \left[ \frac{\bar{v}_R - p - c_R(1-\alpha)\bar{F}_H(\theta)}{\bar{v}_R + c_R\alpha} \right]^+ + \theta(1-\alpha)\bar{F}_H(\theta). \quad (1)$$

In general, it may not be optimal to set a price such that a strictly positive fraction of customers in both channels purchase. However, as shown in Lemma A.3 in the online supplement, a condition can be derived that guarantees this, implying that the bracketed term in (1) is strictly positive. This condition is:

**Assumption 1.** *The parameter values satisfy  $c_R(1-\alpha)/4 < \bar{v}_R/2 < \bar{v}_H$ .*

For the remainder of the paper, we assume Assumption 1 holds, since this implies a centralized firm will operate both channels simultaneously, and as a result, in a decentralized system, achieving coordination requires that the platform and restaurant coexist. Given this, we may next derive the centrally optimal prices:

**Proposition 1.** *The optimal price pair  $(p^*, \theta^*)$  that maximizes total revenue (1) uniquely solves the system  $p^* = (\bar{v}_R - c_R(1-\alpha)\bar{F}_H(\theta^*)) / 2$  and  $\theta^* = (\bar{v}_H + p^*c_R\alpha / (\bar{v}_R + c_R\alpha)) / 2$ .*

*Proof.* All proofs appear in §B of the online supplement. □

Proposition 1 will allow us to compare firm decisions in a decentralized system under various contract forms to that of a centralized system. Note that at optimality, the centralized system sets both the dine-in and delivery price to account for congestion costs generated in the dine-in channel by delivery customers, i.e., taking into account  $c_R$ . If the waiting cost is zero, i.e.,  $c_R = 0$ , the prices are set to independently maximize the revenue in each channel. However, if  $c_R > 0$ , the optimal dine-in price is *lower* to account for reduced customer utility due to waiting, while the optimal delivery price is *higher* to reduce the amount of congestion—a negative externality—that the delivery channel generates for dine-in customers.

## 4 Simple Revenue Sharing Contracts

In this section, we analyze a decentralized system, focusing on the performance of the prevalent industry contract: a simple revenue sharing (RS) agreement in which the platform shares with the restaurant a fraction of revenue from delivery orders, while the restaurant keeps all dine-in revenue. We denote by  $a$  the share of the revenue from delivery customers that the restaurant receives. The platform's fraction of

revenue,  $1 - a$ , is often called a commission. Under RS, the restaurant's revenue for each delivery customer is  $a\theta$ , and the platform's revenue per delivery customer is  $(1 - a)\theta$ . The contract parameter  $a$  is determined exogenously outside of our model, e.g., via a take-it-or-leave-it offer from the platform (as is common with smaller, independent restaurants) or a negotiation process (as is typical with larger restaurant chains).

In what follows, we consider three variations of RS, all of which can be found in practice: §4.1 analyzes a base version of the contract with no constraints placed on firm decisions, §4.2 considers a variation with a cap placed on the platform's commission, and §4.3 considers a variation in which the restaurant is allowed to set a floor on the platform's delivery channel price.

## 4.1 Base Contract

In a decentralized system with RS, from Lemma A.1, the platform's revenue is given by  $z_{P,RS}(\theta) = (1 - a)\theta(1 - \alpha)\bar{F}_H(\theta)$ . Maximizing this over  $\theta$  gives the optimal delivery price  $\theta_{RS}$ . Note that  $\theta_{RS}$  will clearly be independent of the restaurant's dine-in price,  $p$ , since this appears nowhere in  $z_{P,RS}(\theta)$ . Hence, the restaurant can treat the platform's price (and therefore also the restaurant's total revenue from delivery customers) as constant. From Lemma A.2, the restaurant's revenue can then be written as

$$z_{R,RS}(p) = p\alpha \left[ \frac{\bar{v}_R - p - c_R(1 - \alpha)\bar{F}_H(\theta_{RS})}{\bar{v}_R + c_R\alpha} \right]^+ + a\theta_{RS}(1 - \alpha)\bar{F}_H(\theta_{RS}). \quad (2)$$

Maximizing  $z_{P,RS}(\theta)$  and  $z_{R,RS}(p)$  yields the equilibrium prices under RS:

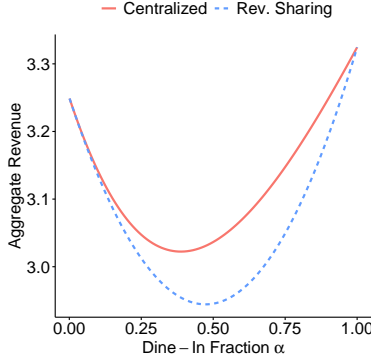
**Proposition 2.** *Under RS, the platform's optimal price is  $\theta_{RS} = \bar{v}_H/2$  and the restaurant's optimal price is  $p_{RS} = (\bar{v}_R - c_R(1 - \alpha)\bar{F}_H(\theta_{RS}))/2$ .*

Observe that, since a higher delivery price reduces congestion generated in the dine-in channel, the dine-in price  $p_{RS}$  is increasing in the delivery price  $\theta_{RS}$ . This observation leads to the following corollary:

**Corollary 1.** *Under RS, if  $c_R > 0$ , the delivery price, dine-in price, and dine-in purchase fraction are too low compared to the central optimum ( $\theta_{RS} < \theta^*$ ,  $p_{RS} < p^*$ , and  $\gamma_R^{RS} < \gamma_R^*$ ).*

Comparing Propositions 1 and 2, we see that under RS, if  $c_R = 0$  the centralized optimum is achieved. However, if  $c_R > 0$ , the optimal delivery price does not take into account the negative externality it generates on the dine-in channel as in the centralized system. Instead, the platform maximizes delivery revenue ignoring its impact on dine-in congestion cost, and as a result, sets a price that is too low. Because the dine-in price is increasing in the delivery price, the restaurant also lowers its price to induce dine-in customers to purchase. Despite the fact that prices are too low, excess congestion crowds out enough dine-in customers to yield a lower dine-in volume than the centrally optimal solution. Hence, although the platform does give the restaurant access to incremental revenue from delivery orders, it both lowers the restaurant's margins and reduces its dine-in demand, consistent with the real-world examples of Dunn (2018).





**Figure 1.** Revenue in the centralized system and a decentralized system with simple revenue sharing. In the example,  $\bar{v}_R = 25$ ,  $\bar{v}_H = 13$ ,  $c_R = 22$ , and  $a = 0.85$ .

To understand the cost of this lack of coordination, we conducted a comprehensive numerical study consisting of 30,400 parameter combinations (not counting variations in the revenue share  $a$ , which do not affect the central optimum or decentralized system revenue) including values of  $\alpha$  from .05 to .95, values of  $c_R$  from 0 to 30, values of  $\bar{v}_R$  from 0 to 40, and values of  $\bar{v}_H$  from 0 to 40.<sup>4</sup> The percentage revenue loss under RS ranges from 0% to 26.2%, with an average of 1.1% and a median of .2%. Because the restaurant industry has very small margins, on the order of 4-5% of sales or even less (Lunden, 2020), even a single-digit percentage reduction in the top line can have an outsize impact on the bottom line. Hence, we conclude that the revenue losses associated with RS can be substantial.

We also observe that the loss under RS is greatest when the dine-in fraction  $\alpha$  is intermediate; see Figure 1. If  $\alpha \rightarrow 0$ , almost all customers are in the delivery channel; any level of congestion in the dine-in channel is thus inconsequential to total revenue, and by comparing Propositions 1 and 2, we see that the platform under RS sets a price close to the central optimum. On the other hand, if  $\alpha \rightarrow 1$ , almost all customers are in the dine-in channel; in this case, the delivery channel cannot generate enough congestion to have a significant impact on dine-in customers, so both firms under RS set prices close to the central optimum. Intermediate  $\alpha$ , on the other hand, is a “worst case scenario” for the restaurant and the platform: there are many potential customers on both channels, and hence they can have a significant impact on one another, causing aggregate revenue to suffer most.

This suggests that RS performs well if  $\alpha$  is very small or very large, or if  $c_R = 0$ . During the COVID-19 pandemic, for example,  $\alpha$  was reduced to essentially zero, implying that RS is sufficient to generate good performance; indeed, during the height of the pandemic, restaurants were grateful for delivery platforms as one of their only ways to generate revenue. However, as society emerges from the pandemic and  $\alpha$  increases, the inefficiency of RS contracts will worsen, potentially increasing the conflict between platforms and restaurants as they jockey over commissions and profitability in a post-pandemic landscape (Elejalde-Ruiz, 2021).

Finally, we note that not only is RS sub-optimal in the sense that it cannot achieve the centralized optimal

<sup>4</sup>In 69.7% of these parameter combinations, Assumption 1 holds, while in the remainder it does not; thus, our numerical study verifies that coordination is valuable in the general case when Assumption 1 may not be satisfied.

revenue, but it can even perform worse than if the platform and restaurant used no contract at all. Absent a formal agreement, the platform can act like a regular customer, paying menu price for each order to the restaurant and then delivering to customers with a surcharge (i.e., essentially placing a take-out order on behalf of each delivery customer). This approach, which has been controversial among restaurateurs due to the fact that platforms can deliver their food without their explicit cooperation, was at one time used by Grubhub and Postmates (Saxena, 2019), but has since been outlawed in some localities such as California (Batey, 2021). We discuss this scenario in the online supplement (§C), and show that the aggregate revenue can in fact be *higher* without a contract than under RS. The reason for this is that, by forcing the platform to set the delivery price at least as high as the restaurant’s dine-in price, the restaurant avoids the vicious cycle that occurs under RS wherein the platform sets too low a price for delivery, forcing the restaurant to set too low a price for dine-in. However, the no contract scenario can also be significantly worse than RS depending on the contract and problem parameters, underscoring the need for a more effective contract.

## 4.2 Commission Caps

In an effort to protect restaurant margins, recent regulations in some US cities and states have capped the commission charged by delivery platforms at 15% or similar (Lucas, 2020), initially in response to the COVID-19 pandemic but potentially becoming permanent in some municipalities (Forman, 2021). It is not clear, however, that these caps have their intended effect, namely protecting small independent restaurants, and recent findings in Li & Wang (2021a) have shown that commission caps help larger chain restaurants more than smaller ones. Below, we discuss the implications of commission caps in the context of our model.

Commission caps are a special case of simple revenue sharing in which the commission  $1 - a$  paid to the platform for delivery orders is limited below a maximum value, or equivalently, there is a lower bound  $\underline{a}$  on the restaurant’s share. In the context of our model, a commission cap could potentially benefit the restaurant by increasing its share of the revenue above the level that emerges from the initial (not modeled) contract determination stage. However, because Corollary 1 holds for any revenue share  $a$ , we have the following:

**Corollary 2.** *Under RS with a commission cap, for any cap  $\underline{a} < 1$ , the equilibrium prices in both channels and the equilibrium dine-in volume are strictly less than centrally optimal.*

In other words, because the underlying contract is still RS, both parties still price too low under a commission cap, and the restaurant still sees less dine-in volume than centrally optimal. Because of this, while a commission cap can help allocate a larger share of the system revenue to the restaurant, aggregate revenue will still be lower than in a centralized system, i.e., money is still left on the table. Combined with the insight that the efficiency of RS is greatest when  $\alpha \rightarrow 0$  or 1, this suggests that commission caps may be effective in protecting restaurant revenues when almost all customers come from one channel; however, commission caps fail to fix the underlying problem with RS, namely that the platform does not account for the congestion effects of delivery customers on dine-in revenue and, as a result, cannot help increase system

profit for intermediate  $\alpha$ .

### 4.3 Delivery Price Floor

We next consider another variation of RS that introduces a potential remedy to the problem of delivery prices that are too low. In practice, platforms sometimes allow restaurants to set the menu price for delivery orders, on top of which the platform adds its own fees: this changes the structure of RS contracts by allowing the restaurant to set two separate menu prices, one for dine-in customers and another for delivery customers. Since the platform can further raise the price by adding its own fees, this amounts to allowing the restaurant to set a price floor in the delivery channel.

To model this scenario, we suppose that the restaurant sets a floor  $\underline{\theta}$  for the delivery channel price, and the platform subsequently chooses a delivery price  $\theta \geq \underline{\theta}$ . Other than the price floor, the setting is equivalent to RS, i.e., the restaurant receives a fraction  $a$  of delivery revenues.<sup>5</sup> From Proposition 2, we know that under RS, the platform's revenue is unimodal in  $\theta$  and that the optimal  $\theta$  does not depend on  $p$ . Hence, with a price floor  $\underline{\theta}$  determined by the restaurant, the equilibrium delivery price chosen by the platform will be  $\max\{\theta_{RS}, \underline{\theta}\}$ . The restaurant thus chooses a price floor that is not binding in equilibrium (in which case the outcome is identical to Proposition 2), or one that binds in equilibrium, in which case the restaurant effectively chooses the price in both channels. We focus on the latter scenario, as this turns out to be always optimal. The restaurant receives all of the dine-in revenue and a fraction  $a$  of the delivery revenue, hence its revenue function is the same as equation (2), but now  $\theta$  is a decision variable, i.e.,

$$z_{R,PF}(p, \theta) = p\alpha \left[ \frac{\bar{v}_R - p - c_R(1 - \alpha)\bar{F}_H(\theta)}{\bar{v}_R + c_R\alpha} \right]^+ + a\theta(1 - \alpha)\bar{F}_H(\theta). \quad (3)$$

This is similar to the centralized system revenue function in (1), with one difference: because the second term of (3) is multiplied by  $a < 1$ , the restaurant will, relative to the centralized system, overvalue dine-in revenue and undervalue delivery revenue. This will cause the platform to set a delivery channel price that is too high relative to the centralized optimum, because it feels the full cost of increased delivery volume (reflected in dine-in congestion cost) but only part of the benefit of increased delivery volume (reflected in its  $a < 1$  share of delivery revenue). The dine-in price will also be too high, since the amount of congestion generated in the dine-in channel is less than centrally optimal. This is formalized in the following proposition, in which the equilibrium prices with a price floor are denoted by  $p_{PF}$  and  $\theta_{PF}$ :

**Proposition 3.** *Under RS with a delivery price floor, for  $0 \leq a < 1$ , the equilibrium prices in both channels are higher than the central optimum ( $\theta_{PF} > \theta^*$  and  $p_{PF} > p^*$ ).*

<sup>5</sup>We note that, sometimes in practice, the platform only splits revenue of the menu price with the restaurant, i.e., the platform keeps 100% of the customer delivery fees. In other words, for a delivery fee  $f$ , menu price  $\theta'$ , and restaurant revenue share  $a'$ , platform revenue is  $f + (1 - a')\theta'$  while restaurant revenue is  $a'\theta'$ . Our formulation is equivalent to this if we define a new menu price  $\theta \equiv f + \theta'$  and a new revenue share  $a \equiv a'\theta'/(f + \theta')$ , i.e., for any  $f$ ,  $a'$ , and  $\theta'$ , we can provide a  $\theta$  and  $a$  such that our model results in equivalent payments.

Having the power to set a delivery price floor leads the restaurant to set a price floor that is higher than the centrally optimal delivery price, which leads to delivery and dine-in prices that are higher than centrally optimal. This implies that, under revenue sharing with a price floor, customers in *both* channels pay higher prices than under simple revenue sharing or at the central optimum; indeed, it is possible for the restaurant to set the price floor so high that no customers purchase from the platform, causing the delivery channel to shut down. Hence, allowing the restaurant to set a price floor for the delivery channel does give it the power to protect its margins, although it is perhaps too effective in doing so: the result is that the restaurant maximizes its own revenue to the detriment of the entire system, failing to achieve coordination.

## 5 Coordination with Generalized Revenue Sharing

In this section, we consider how to remedy the problem with RS and coordinate the system. We first note that an obvious way to coordinate the system is for the restaurant to share dine-in revenue with the platform, in addition to the platform sharing delivery revenue with the restaurant, i.e., a “two-way revenue sharing” contract. It is straightforward to show (details omitted) that with appropriately chosen parameters, such a contract coordinates the system because it makes both revenue functions into affine transformations of the aggregate revenue (see Xu *et al.*, 2014). However, requiring the restaurant to share *dine-in* revenue with the platform is impractical, and restaurants are unlikely to agree to this.<sup>6</sup> On the other hand, in the literature on supplier encroachment several contract types have been shown to coordinate a system with direct and indirect channels wherein competition generates negative externalities between channels. These mechanisms include pricing constraints on one or both parties (e.g., Cai, 2010) and non-linear pricing (e.g., Tsay & Agrawal, 2004); as noted earlier, these are impractical for our setting.

Because of this, we turn next to devising a simple and practically implementable coordinating contract. Ideally, such a contract will cause the platform to appropriately account for the congestion that its customers create in a way that only affects payments between the platform and restaurant on delivery orders. In addition, such a contract should avoid non-linear payment schemes or constraints placed upon the pricing power of either firm. Finally, it is also desirable to protect restaurant revenue (i.e., to ensure such revenue is not too low) on delivery orders, and to allocate revenue between the restaurant and platform with a high degree of flexibility. In this section, we propose just such a contract, which we call *generalized revenue sharing* (GRS), and demonstrate that it accomplishes all of these goals.

In a GRS contract, for each order, the platform pays the restaurant a fixed fee  $\tau$  and a percentage  $a$  of the delivery price  $\theta$ . Under this contract,  $\tau$  is analogous to a linear wholesale price paid on each order, and  $a$  is the restaurant’s share of delivery revenue. Using Lemma A.1, the platform’s revenue is  $z_{P,G}(p, \theta) = \left( (1 - a)\theta - \tau \right) \bar{F}_H(\theta)(1 - \alpha)$ , where the subscript  $G$  denotes the GRS contract. Using Lemma

---

<sup>6</sup>Two-way revenue sharing also has a verifiability problem: the restaurant has no incentive to accurately disclose its dine-in revenue to the platform, i.e., it prefers to under-report dine-in revenue. Note that the reverse cannot happen: the platform cannot under-report delivery volume or revenue to the restaurant, since the restaurant prepares each meal that the platform delivers.

A.2, the restaurant's revenue is

$$z_{R,G}(p, \theta) = p\alpha \left[ \frac{\bar{v}_R - p - c_R(1 - \alpha)\bar{F}_H(\theta)}{\bar{v}_R + c_R\alpha} \right]^+ + (a\theta + \tau)(1 - \alpha)\bar{F}_H(\theta).$$

Even though platform revenue does not directly include the dine-in congestion cost, an appropriately chosen  $\tau$  can induce the platform to choose the centrally optimal delivery price, as the following result shows.

**Proposition 4.** *With a generalized revenue sharing contract, for  $0 \leq a < 1$ , if  $\tau = (1-a)p^*(c_R\alpha/(\bar{v}_R + c_R\alpha))$ , the platform sets  $\theta = \theta^*$  and the restaurant sets  $p = p^*$ , i.e., the system is coordinated.*

GRS coordinates the system because  $\tau$  charges the platform for the marginal negative externality generated by a delivery order on the dine-in channel. With linear waiting costs, the coordinating  $\tau$  takes an appealingly simple form: it is a  $1 - a$  share of a fraction  $c_R\alpha/(\bar{v}_R + c_R\alpha) < 1$  of the centrally optimal dine-in price. This represents the marginal loss in revenue that occurs from an incremental increase in dine-in congestion, multiplied by the platform's share of the delivery revenue,  $1 - a$ . By charging this amount to the platform for each delivery order, the platform will both enjoy a  $1 - a$  share of the marginal delivery revenue *and* incur a  $1 - a$  share of the marginal congestion cost it generates for the dine-in channel, and hence choose the centrally optimal delivery price; anticipating this, the restaurant chooses the centrally optimal dine-in price as well.

The coordinating contract is, essentially, a linear wholesale price contract plus a revenue share. This resembles the revenue sharing contract for physical inventories discussed in Cachon & Lariviere (2005), but the contract works in a different way, and the wholesale price plays a different role, in our setting. In Cachon & Lariviere (2005), incentive misalignment arises because of double marginalization, and the coordinating contract is a profit sharing contract because the retailer pays the same fraction of the production cost as the fraction of revenue it retains, i.e., the channel cost is shared in the same proportion as revenue. In our setting, incentive misalignment does not occur because of double marginalization within a single channel, but rather due to a negative externality *between* channels arising from the specific features of the restaurant industry. Because of this, the coordinating revenue sharing contract in Cachon & Lariviere (2005) would assign a per-order fee of zero when applied to our setting (since we have assumed production cost is zero) and would not coordinate our system. By contrast,  $\tau$  in our coordinating contract depends not on marginal production cost but on the drivers of the negative externality, specifically customer valuations, waiting costs, and the relative sizes of the dine-in and delivery channels.

The simplicity of the GRS contract—observe that it only involves adjustments made to the payment between the platform and the restaurant for each delivery order, and moreover, the resulting payment is linear in the number of orders—means that it is practically implementable in our setting. As discussed previously, it is also desirable to ensure restaurant revenues are not too low on delivery orders, and to flexibly allocate revenue between parties. The next result shows that GRS possesses both properties:

**Proposition 5.** *Under a coordinating generalized revenue sharing contract:*

(i) *For each delivery order, the restaurant receives at least  $p^*(c_R\alpha)/(\bar{v}_R + c_R\alpha)$ . If  $\bar{v}_R < \bar{v}_H$  and  $a \geq \bar{v}_R/\bar{v}_H$ ,*

the restaurant earns at least as much for a delivery order as for a dine-in order, i.e.,  $a\theta^* + \tau \geq p^*$ .

(ii) By varying  $a$ , the contract can allocate to the platform any fraction between 0 and  $\delta$  of the delivery revenue, where  $\delta \geq 1 - (\bar{v}_R/\bar{v}_H)(c_R\alpha)/(\bar{v}_R + c_R\alpha)$ . Moreover, if  $c_R\alpha \leq \bar{v}_R \leq \bar{v}_H$ , then  $\delta \geq 1/2$ .

Proposition 5(i) gives a lower bound on the revenue per order that a restaurant can expect under a GRS contract, and shows this is a fraction of the dine-in price that is increasing in the congestion cost ( $c_R$ ) and dine-in fraction ( $\alpha$ ). This shows that if congestion is very costly to the restaurant ( $c_R$  is high or there are many dine-in customers), a GRS contract ensures the restaurant receives significant revenue on each delivery order. In fact, in some cases—if delivery customers have sufficiently high valuations, and if the restaurant’s share of delivery revenue is high enough—GRS results in a *higher* revenue per order to the restaurant for delivery orders than for dine-in orders. In an industry with many small, independent restaurants with little bargaining power, these are highly desirable features, protecting restaurant revenue (like commission caps or price floors) while *also* coordinating the system.

Part (ii) also shows that while GRS cannot achieve full flexibility in allocating delivery revenue, it does achieve a wide range of allocations. (Note it is impossible to offer full flexibility while also protecting restaurant revenue.) In particular, when dine-in customers and delivery customers value the respective services highly enough, the contract can achieve anywhere from a 50-50 split of delivery revenue (at one extreme, a wholesale price contract with  $a = 0$ ) to full allocation to the restaurant (the other extreme with  $a = 1$ ). Hence, while the precise contract terms are a matter of negotiation between restaurant and platform, GRS possesses the flexibility to accommodate a wide range of scenarios while also achieving coordination.

In short, GRS contracts simultaneously protect the restaurant and increase aggregate revenue, while ensuring a wide range of revenue allocations are possible. The existence of a simple coordinating contract that is practically implementable in this setting is good news for restaurants and platforms. In addition, policymakers interested in regulating this market can take note. After commission caps were instituted in some cities in 2020, platforms disputed them with legal action and lobbying efforts, and Uber Eats even tacked on an additional fee for customers to offset lost commissions (Carson, 2020; Brown, 2020). Our results show that rather than attempting to adjust the parameters of an inherently flawed contract—simple revenue sharing—policymakers and firms alike might consider a different contractual form, generalized revenue sharing, that can mitigate the incentive conflict between platforms and restaurants and maximize joint revenue.

## 6 Robustness

### 6.1 Positive Delivery Channel Waiting Costs

In our base model, we assumed that congestion was costly for dine-in customers (i.e.,  $c_R > 0$ ) but costless for delivery customers ( $c_H = 0$ ). This assumption simplifies the analysis and allows us to derive clean insights; however, in §D of the online supplement, we numerically relax this assumption to consider the case where

delivery customers experience positive waiting costs, i.e.,  $0 < c_H < c_R$ . When  $c_H > 0$ , a second negative externality is generated: dine-in customers cause delays that are costly to delivery customers. We find that, as one might expect, RS continues to be incapable of achieving coordination in this case, since the platform still fails to account for the negative externality it generates for dine-in customers; hence, our insights about the detrimental effects of RS persist.

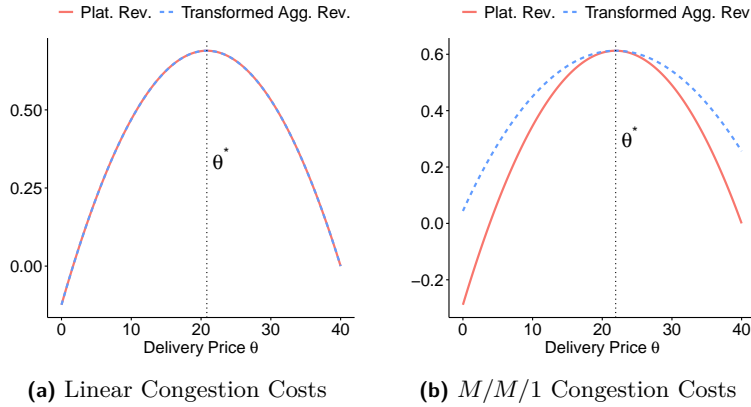
While it can be shown that the GRS contract proposed in Proposition 4 (i.e., assuming  $c_H = 0$ ) does not always coordinate when  $c_H > 0$ , we numerically observe that it performs exceptionally well, even when using contract parameters optimized for the  $c_H = 0$  case as in the proposition: in 2,700 parameter combinations, we find that the average revenue loss relative to the central optimum is just 0.006% (compared to an average loss of 1.2% with RS) while the maximum loss is 0.3% (compared to a maximum of 11.1% with RS). GRS performs extremely well because the restaurant, in receiving a share of the delivery revenue, already has a strong incentive to account for the second externality it generates for the delivery channel; consequently, we conclude that our key insights are robust to this assumption, and even under the more general setting in which *both* delivery and dine-in customers are congestion sensitive, the GRS contract we propose exhibits excellent performance.

## 6.2 Non-Linear Waiting Costs

Our base model also assumes waiting costs for dine-in customers are linear and equal to  $c_R x$ , where  $x$  is the total purchase volume. Linearity is not necessary to generate our key insights, as shown in §E of the online supplement. Specifically, we show that if dine-in customers have a waiting cost function  $W(x)$  that is increasing and convex in  $x$ , it remains true that RS cannot coordinate, and moreover, GRS coordinates the system if the platform pays the restaurant a fraction  $a$  of delivery revenue and a fixed fee of  $\tau = (1 - a)\alpha\bar{F}_R(v_R^*)W'(x^*)$  on each delivery order, where  $v_R^*$  is the centrally optimal dine-in valuation threshold. The quantity  $\alpha\bar{F}_R(v_R^*)$  is the centrally optimal dine-in purchase volume, and  $W'(x^*)$  is the marginal congestion cost of a dine-in customer at the centrally optimal total purchase volume  $x^*$ . Hence, the coordinating  $\tau$  is the platform’s share of the marginal congestion cost that would be added to the system by an additional delivery customer, although it is no longer a simple fraction of the centrally optimal dine-in price as in the linear congestion cost case.

This shows that for important practical classes of cost functions, such as quadratic or queueing-derived congestion costs, our insights continue to hold. The fact that a contract with a linear per-order fee coordinates even when waiting costs are non-linear is not *a priori* obvious. Figure 2(a) and 2(b) illustrate why linear fees work with non-linear costs.<sup>7</sup> With linear waiting costs (a), the linear fee essentially makes the platform’s revenue function an affine transformation of the centralized system revenue at every point, ensuring the

<sup>7</sup>In the figure, the curve labeled “Transformed Agg. Rev.” represents a particular affine transformation of the aggregate revenue function, while the curve labeled “Plat. Rev.” represents the platform’s revenue function under a coordinating generalized revenue sharing contract. In both (b) and (c), the specific non-linear congestion cost function used derives from the time in queue for an  $M/M/1$  system, i.e.,  $W(x) = c_R(1/(\mu - \lambda x) - 1/\mu)$ , where  $\mu = \lambda = 1$  and  $x$  equals the total purchase fraction.



**Figure 2.** Platform revenue and an affine transformation of centralized revenue for  $c_R = 8$ ,  $\bar{v}_R = 30$ ,  $\bar{v}_H = 40$ ,  $\alpha = .5$ , and  $a = .85$ .

platform chooses the centrally optimal delivery channel price; because of this, the curves in (a) overlap. With non-linear costs (b), a linear fee can no longer accomplish this. However, it can make the platform’s revenue function *tangent* to an affine transformation of the centralized system revenue, precisely at the centrally optimal delivery price; hence, the curves in (b) touch only at the centrally optimal price. This induces the platform to choose the proper delivery price, and conditional on this, the restaurant will choose the centrally optimal dine-in price. Consequently, the structure of the coordinating GRS contract—a linear fee for each order plus a fraction of delivery revenues—is the same even with non-linear costs.

## 7 Conclusion

In this paper, we have considered how to best structure relationships between food delivery platforms and the restaurants with whom they partner. Our results show that the most common contractual relationship between platforms and restaurants has inherent flaws. Simple revenue sharing does not appropriately charge the platform for the negative externality it generates on the dine-in channel, and as a result, cannot coordinate the system and can reduce the profitability of restaurants, many of whom already experience very thin margins. Neither commission caps nor delivery channel price floors can remedy this shortcoming.

This supports the popular view (Houck, 2017; Dunn, 2018; Meyersohn, 2018) that delivery platforms may *not* benefit restaurants, and partnering with a third-party delivery service can lead to a vicious cycle in which service quality and profitability deteriorate despite the increase in volume. On the other hand, a generalized revenue sharing contract can coordinate the system using a simple and practically implementable contract form involving shared delivery order revenue and a fixed fee per order paid by the platform to the restaurant, in addition to protecting restaurant revenues and flexibly allocating delivery revenue.

Given the relative novelty of this business model and the rapidity with which it is evolving, there are many aspects of the relationship between platforms and restaurants that warrant further study. Future work



could incorporate other factors that may influence the profitability of a delivery platform or the restaurants that partner with it, such as competition between restaurants or platforms, or the capacity decisions of restaurants (e.g., should a restaurant add more shared kitchen capacity or should it invest in dedicated delivery kitchen capacity, a so-called “ghost kitchen” model, see Hawley, 2020). More broadly, as delivery platforms continue to grow in popularity, it will become increasingly important for them to effectively manage relationships with restaurants in order to avoid conflict that threatens the viability of their business model. Our work illustrates some of the important issues that can arise in these relationships, and offers a simple and practically implementable way to alleviate them and improve coordination of the food delivery supply chain.

## References

- Arya, Anil, Mittendorf, Brian, & Sappington, David EM. 2007. The bright side of supplier encroachment. *Marketing Science*, **26**(5), 651–659.
- Baron, Opher, Chen, Xiaole, & Li, Yang. 2019. The paradox of choice: The false premise of omnichannel services and how to realize it. Available at SSRN 3444772.
- Batey, Eve. 2021. *New California Law Raptures Thousands of Restaurants From Postmates, DoorDash, and Grubhub*. <https://sf.eater.com/2021/1/4/22213402/restaurants-removed-postmates-grubhub-california-law-2021>.
- Boyaci, Tamer. 2005. Competitive stocking and coordination in a multiple-channel distribution system. *IIE transactions*, **37**(5), 407–427.
- Brown, H. Claire. 2020. Cities capped food delivery platform fees during the pandemic. Grubhub and Postmates resisted these limits, citing confusion. Accessed July 10, 2021. *The Counter*. <https://thecounter.org/food-delivery-platform-fee-caps-grubhub-postmates-covid-19/>.
- Buell, Ryan. 2017. Breakfast at the Paramount. *Harvard Business School Case Study*.
- Cachon, Gérard P. 2003. Supply chain coordination with contracts. *Handbooks in Operations Research and Management Science*, **11**, 227–339.
- Cachon, Gérard P, & Lariviere, Martin A. 2005. Supply chain coordination with revenue-sharing contracts: Strengths and limitations. *Management Science*, **51**(1), 30–44.
- Cai, Gangshu George. 2010. Channel selection and coordination in dual-channel supply chains. *Journal of Retailing*, **86**(1), 22–36.
- Carson, Biz. 2020. Delivery companies are fighting city commission caps. Does anybody win? Accessed July 10, 2021. *Protocol*. <https://www.protocol.com/delivery-commission-caps-uber-eats-grubhub>.
- Chen, Manlu, Hu, Ming, & Wang, Jianfu. 2020. Food delivery service and restaurant: Friend or foe? Available at SSRN 3469971.
- Deliso, Meredith. 2021. *Why restaurants are having a hard time finding staff right now*. <https://abcnews.go.com/Business/restaurants-hard-time-finding-staff-now/story?id=77562018>.
- Dunn, Elizabeth. 2018. How Delivery Apps May Put Your Favorite Restaurant Out of Business. Accessed April 24, 2019. <https://www.newyorker.com/culture/annals-of-gastronomy/are-delivery-apps-killing-restaurants>.
- Elejalde-Ruiz, Alexia. 2021. *Restaurants say delivery has been both a blessing and a curse during the pandemic. What happens as eateries reopen?* <https://www.chicagotribune.com/business/ct-biz-future-of-food-delivery-20210301-gep3h5kbzcc7hiuj3zobvawoyu-story.html>.
- Forman, Laura. 2021. Food-Delivery Regulation Is a Drama Worth Watching. Accessed April 12, 2021. <https://www.wsj.com/articles/food-delivery-regulation-is-a-drama-worth-watching-11614859201>.

- Ha, Albert, Long, Xiaoyang, & Nasiry, Javad. 2015. Quality in supply chain encroachment. *Manufacturing & Service Operations Management*, **18**(2), 280–298.
- Hawley, Kristen. 2020. *Ghost Kitchens Are the Wave of the Future. But Is That a Good Thing?* <https://www.eater.com/21540765/ghost-kitchens-virtual-restaurants-covid-19-industry-impact>.
- Houck, Brenna. 2017. Why Some Restaurants Are Cutting Ties With Mobile Ordering Apps. Accessed April 24, 2019. <https://www.eater.com/2017/8/29/16214442/restaurant-order-pay-apps-seamless-postmates-uber-eats>.
- Lariviere, Martin A. 2015. OM Forum—Supply Chain Contracting: Doughnuts to Bubbles. *Manufacturing & Service Operations Management*, **18**(3), 309–313.
- Li, Zhuoxin, & Wang, Gang. 2021a. Regulating Powerful Platforms: Evidence from Commission Fee Caps in On-Demand Services. *Available at SSRN*.
- Li, Zhuoxin, & Wang, Gang. 2021b. The role of on-demand delivery platforms in restaurants. *Working Paper, Available at SSRN 3813891*.
- Li, Zhuoxin, Gilbert, Stephen M, & Lai, Guoming. 2013. Supplier encroachment under asymmetric information. *Management Science*, **60**(2), 449–462.
- Li, Zhuoxin, Gilbert, Stephen M, & Lai, Guoming. 2015. Supplier encroachment as an enhancement or a hindrance to nonlinear pricing. *Production and Operations Management*, **24**(1), 89–109.
- Liu, Sheng, He, Long, & Max Shen, Zuo-Jun. 2020. On-time last-mile delivery: Order assignment with travel-time predictors. *Management Science*.
- Lucas, Amelia. 2020. Local Lawmakers Provide Struggling Restaurants with Temporary Relief From Food Delivery Fees. Accessed April 12, 2021. <https://www.cnbc.com/2020/05/15/city-lawmakers-provide-restaurants-with-relief-from-delivery-fees.html>.
- Lunden, Ingrid. 2020. Restaurant rewards booking app Seated nabs 30M, acquires VenueBook to add events. Accessed May 4, 2021. <https://techcrunch.com/2020/08/18/restaurant-rewards-booking-app-seated-nabs-30m-acquires-venuebook-to-add-events>.
- Mao, Wenzheng, Ming, Liu, Rong, Ying, Tang, Christopher S, & Zheng, Huan. 2019. Faster deliveries and smarter order assignments for an on-demand meal delivery platform. *Available at SSRN 3469015*.
- Meyersohn, Nathaniel. 2018. Why Uber Eats and GrubHub partnerships are risky for restaurants. Accessed April 24, 2019. <https://money.cnn.com/2018/03/28/news/companies/uber-eats-grubhub-delivery-apps/index.html>.
- Niu, Baozhuang, Li, Qiyang, Mu, Zihao, Chen, Lei, & Ji, Ping. 2021. Platform logistics or self-logistics? Restaurants cooperation with online food-delivery platform considering profitability and sustainability. *International Journal of Production Economics*, **234**, 108064.
- Oh, Jaelynn, Su, Xuanming, & Glaeser, Chloe Kim. 2019. Restaurant Delivery Platforms, Commission Rates, and Delivery Fees. *Working Paper, University of Utah*.
- Restaurant Engine. 2021. *Ultimate Guide to Restaurant Delivery Services: UberEats, GrubHub, PostMates and DoorDash*. <https://restaurantengine.com/ultimate-guide-to-restaurant-delivery-services-ubereats-grubhub-postmates-and-doordash/>.
- Saxena, Jaya. 2019. *Grubhub's New Strategy Is to Be an Even Worse Partner to Restaurants*. <https://www.eater.com/2019/10/30/20940107/grubhub-to-add-restaurants-without-permission-like-postmates>.
- Tsay, Andy A, & Agrawal, Narendra. 2004. Channel conflict and coordination in the e-commerce age. *Production and operations management*, **13**(1), 93–110.
- Xu, Guangye, Dan, Bin, Zhang, Xumei, & Liu, Can. 2014. Coordinating a dual-channel supply chain with risk-averse under a two-way revenue sharing contract. *International Journal of Production Economics*, **147**, 171–179.
- Yang, Huixiao, Luo, Jianwen, & Zhang, Qinhong. 2018. Supplier encroachment under nonlinear pricing with imperfect substitutes: Bargaining power versus revenue-sharing. *European Journal of Operational Research*, **267**(3), 1089–1101.