Flash Pass*

Yuichiro Kamada[†] Zihao Zhou[‡]

November 4, 2022

Preliminary Draft

This paper is frequently updated. The latest version can be found at

http://ykamada.com/pdf/flash.pdf

Abstract

We consider a model in which an amusement park sells different priority passes to customers in a queue whose utilities depend on positions in the queue. A customer's valuation of a priority pass depends on the distribution of customers buying each pass. Hence, other customers' purchase decisions affect the customer's valuation, which differentiates our model from the standard screening models. We discuss the implementability of selling multiple passes and show that the externality makes the implementation of multi-pass schemes difficult. This issue can persist even when customers have heterogeneous utilities of positions in a queue.

*We are grateful to Ganesh Iyer, Michihiro Kandori, Kei Kawai, Tony Ke, Fuhito Kojima, Chakravarthi Narasimhan, Philipp Strack, and Luyi Yang for their valuable comments and suggestions on this paper. We would also like to thank the feedback from the conference and seminar participants at the 40th Annual ISMS Marketing Science Conference, the 17th Summer Institute in Competitive Strategy (SICS), Hitotsubashi University, Osaka University, University of Tokyo, and Yokohama National University. All errors remain with us.

[†]Yuichiro Kamada: Haas School of Business, University of California, Berkeley, y.cam.24@gmail.com

[‡]Zihao Zhou: UCL School of Management, zihao.zhou@ucl.ac.uk

Contents

1 Introduction

When access to a service facility is congested, service providers commonly implement a type of queue called priority queue, where each person/entity in the queue has an associated priority such that those with a higher priority will be ahead of those with a lower priority in the queue. For example, an amusement park can sell two types of priority passes to its customers, one called regular and the other called flash pass. A customer holding a flash pass is ahead of every regular pass holder in the queue. Event organizers hosting many visitors, such as an exhibition, can let VIP ticket holders skip the queue and enter the venue ahead of regular ticket holders. Airport security checks sometimes set up an express lane where eligible travelers can skip the queue to expedite security checks. In cloud computing, different computing requests queue for the computing resources; within a computer, various programs need to queue for access to the CPU. In this paper, we consider the pricing problem faced by a seller that manages a priority queue, e.g., an amusement park.

When a park sells different priority passes, other customers' purchase decisions affect a customer's valuation of a pass, which differentiates the pricing problem of a priority queue from the problem where a customer's valuation of a pass is fixed. For example, consider a customer buying a flash pass. If she is the only customer buying that pass, then she would be ahead of everyone else, and hence her valuation of the flash pass is high. If, in contrast, everyone else buys the flash pass as well, then she would be in the middle of the queue on average, and thus her valuation of the flash pass would not be so high. This is caused by the externalities that other customers' purchase decisions impose on the customer, and this paper provides insights into the implications of these externalities.

We observe that parks usually sell only a small number of priority passes. For example, Six Flags, a large amusement park corporation in the US, sells only three tiers of priority passes: THE FLASH Pass, THE FLASH Pass Gold, and THE FLASH Pass Platinum. Motivated by this observation, this paper focuses on the implementability of multi-pass schemes, i.e., whether a park can price many priority passes so that each pass has at least one willing customer. For example, we show that when customers have the same utility function concerning the positions in a queue, the park cannot sell a different priority pass to each customer. We further show that, under some conditions on each customer's utility function, implementing a multi-pass scheme is impossible unless different customer utility types are sufficiently different. We formalize that the existence of externalities contributes to the difficulty of selling many priority passes.

The main objective of this paper is to show that externalities can contribute to the constrained number of priority passes that are implementable, and we do not claim to explain the constraint on the number of passes thoroughly. Other factors could play a role in the constraint at the same time. For example, selling many priority passes may incur considerable logistic costs, prompting the park to sell a small number of passes. Too many choices may also overwhelm customers (Park and Jang 2013; Kuksov and Villas-Boas 2010).

Queuing literature, such as Balachandran (1972), Adiri and Yechiali (1974), Hassin and Haviv (1997), and Alperstein (1988), has studied managing priority queues with a setup different from ours. In these papers, each customer sequentially arrives at a queue, observes the queue's state with a forecast about the waiting time in each priority pass, and then chooses a pass to maximize her expected utility. In contrast, our model is static: the customers make purchase decisions simultaneously, so they do not observe the state of the entire queue when they make a purchase decision. We believe these two types of models fit to different situations. Our models may fit better in the applications we have in mind, such as the queues in the amusement park: For example, many customers of Six Flags purchase their priority passes online before their realized arrival. At the purchase time, the customers do not observe the purchase behavior of the customers who have already bought, nor do they expect that their purchase behavior would be observed by other customers to affect their purchase behavior.

As for the number of passes, Alperstein (1988) discussed the optimal pricing as well as the number of passes to sell. The paper finds it profit-optimal to have at most one customer in each priority pass, which implies a large number of passes when many customers are in the queue at the same time. However, the number of priority passes is usually much smaller than the number of customers, and our paper provides one factor contributing to the disparity. Adiri and Yechiali (1974) showed that in equilibrium, early-arriving customers would purchase the lower-priority passes and later-arriving customers will purchase higher-priority passes as the queue gets large, which leads each priority pass to have at least one customer.

Pricing priority passes is like a screening problem. While we use the same approach as in the standard problems such as Guesnerie and Laffont (1984) and Maskin and Riley (1984), we show that implementability in our setup does not extend from those models because of the externalities.[1] In Section 6.2, we discuss how the existence of externalities makes the pricing of priority passes harder.

Several existing studies considered externalities in screening problems. Early works by Bergstrom, Blume, and Varian (1986) and Rasmusen, Ramseyer, and Wiley Jr (1991) considered applications with positive and negative externalities, respectively. Segal (1999) considered a general model of bilateral contracting between a principal and multiple agents.[2]

[1] Specifically, we will show that our setup does not permit the constraint reduction as in the standard screening models.

[2] A follow-up study, Segal and Whinston (2003), also considered a screening problem with externalities. Gomes (2005) and Bloch and Gomes (2006) analyzed multilateral contracting problems with externalities.

He provided results on when the principal-optimal aggregate trade quantity would be above or below the socially optimal aggregate trade quantity. These results, however, would not cover our results due to the conditions he imposes.[3]

For existing applied literature that includes externalities, the nature of externalities varies by application. For example, Katz and Shapiro (1986) analyzed the adoption of new technology under the presence of network effects that greater adoption of the technology increases the utility of adoption; Csorba (2008) considered the externalities when the utility of using a product increases with the rise in demand; Shi, Zhang, and Srinivasan (2019) and Kamada and Öry (2020) both looked at product-line design and pricing questions in which new customers boost the utility of existing customers. None of these papers dealt with externalities in queues.

The paper proceeds as follows. Section 2 and 3 introduce the main model. Section 4 discusses selling multiple priority passes with one utility type. Section 5 extends the main model to cases with multiple types of customer utility functions. Section 6 provides a more detailed discussion of the derived results, with Section 7 concluding the paper.

2 Model

An amusement park chooses $K \geq 1$ and $p = (p_1, \ldots, p_K) \in \mathbb{R}_+^K$, where K is the number of types of priority passes that it sells and p_k is the price for the k-th pass. We denote by θ_k the k-th pass. Let θ_0 denote the option of staying at home and set $p_0 = 0$ for completeness. There are $N \geq 1$ customers who observe the price vector p and then make purchase decisions simultaneously: Each customer either buys some priority pass (i.e., chooses θ_k for some $k = 1, \ldots, K$) or does not buy any pass and leaves the park (i.e., chooses θ_0).

After the purchase decisions, the customers that purchase a priority pass form a queue, with the possible positions in the queue being $1, 2, \ldots, N'$, where N' is the number of customers who bought some pass. For every $k \in \{1, \ldots, K\}$, every customer buying θ_k is guaranteed to be ahead of every customer buying θ_j if $j > k$ and behind every customer buying θ_j if $1 \leq j < k$. For customers in the same priority pass, each order of these customers happens

[3] Specifically, in Segal (1999)'s model, the principal offers a take-it-or-leave-it contract to the agent that specifies the trade quantity between them. The contracting outcomes of other agents can affect the reserve utility of an agent. He found that when the externalities are positive (negative), i.e., an agent's reserve utility is non-decreasing (non-increasing) if other agents are trading more with the principal, then the principal-optimal aggregate trade quantity is below (above) the socially optimal aggregate trade quantity. If we wish to fit our model to the setup of Segal (1999), given a priority queue, we can treat the customers' choices as the "trade profile" (which specifies the trade quantity of each customer, and dependent on which each customer has a utility function) and the optimal deviation payoff of the customer as a customer's reserve utility. The externalities in the fitted model do not satisfy the conditions for the results of Segal (1999), and hence that paper's results do not cover ours.

with the same probability. Hence, each customer's position is uniformly distributed over the possible positions of customers buying the same pass.[4]

A **base utility function** $u : \mathbb{N} \to \mathbb{R}$ assigns a utility to each position in the queue where $u(n)$ denotes the utility from being at the n-th position in the queue.[5] To simplify the notations, we write u_n in place of $u(n)$ in what follows. If a customer buys pass θ_k and receives position n, then her payoff is $u_n - p_k$. If the customer chooses θ_0, her payoff is $u_0 - p_0$ (which equals u_0 since $p_0 = 0$), where u_0 is set to zero unless otherwise specified.[6]

Given any choice by the amusement park, the above setup where customers make a choice can be modeled as a strategic-form game. Define a strategic-form game $G(N, K, p, u) = \langle I, (A_i)_{i\in I}, (\pi_i)_{i\in I}\rangle$, where $I = \{1, \ldots, N\}$ is the set of players, $A_i = \{\theta_0, \theta_1 \ldots, \theta_K\}$ is i's action set, and $\pi_i : A \to \mathbb{R}$ is each customer's payoff function where $A = \times_{i=1}^{N} A_i$: For every $a \in A$, $\pi_i(a) = \overline{v}_i(a) - p_k$, where $\overline{v}_i(a)$ is customer i's expected utility from action profile a and we have $a_i = \theta_k$.

Given an action profile $a \in A$, define $\overline{q}(a) = (\overline{q}_k(a))_{k=0}^{K}$, where $\overline{q}_k(a) = |\{i : a_i = \theta_k\}|$ denotes the number of customers choosing θ_k. Given N and K, define $\mathcal{Q}(N, K) = \{q \in (\{0\} \cup \mathbb{N}) \times \mathbb{N}^K : \sum_{k=0}^{K} q_k = N\}$ to be the set of **schemes** for (N, K). The interpretation is that, for each $q = (q_0, \ldots, q_K) \in \mathcal{Q}(N, K)$, the first coordinate of q (i.e., q_0) denotes the number of customers staying at home; for each $k \in \{1, \ldots, K\}$, the $(k+1)$-th coordinate of q (i.e., q_k) is the number of customers that buy θ_k. By setting $q_k \geq 1$ for $k \geq 1$, we require that each priority pass has at least one customer for q to be a scheme.[7] See Figure 1 for an example of a scheme.

We now define implementability, the main concept of this paper. In short, a scheme is implementable if each customer's purchase decision is optimal given other customers' decisions.

Definition 1 (Implementation)**.** Fix (N, K, u). A price vector p **implements** a scheme $q \in \mathcal{Q}(N, K)$ if $G(N, K, p, u)$ has a pure-strategy Nash equilibrium $a^* \in A$ such that $\overline{q}(a^*) = q$. A scheme q is **implementable** if there exists a price vector p that implements q.

[4] Although it does not affect any of our results, for completeness, one can assume that the randomization of customer orders within a priority pass is independent across different passes.

[5] To clarify, $\mathbb{N}$ denotes the set of strictly positive integers.

[6] We only let u_0 be non-zero in Section 6.3, where we consider the limit as the queue length becomes long.

[7] This restriction is without loss in the following sense: an action profile a such that there are some priority passes without any customer is a Nash equilibrium in the game under some price vector p if and only if the action profile that induces the scheme given by "deleting" these passes is a Nash equilibrium of the game under another price vector p'. In particular, given p. the p' can be obtained by deleting the prices for priority passes that had no customers under p.

Customer	Pass Bought	Price Paid	Expected Utility
A	θ_1	p_1	$\frac{u_1+u_2}{2} - p_1$
B	θ_1	p_1	$\frac{u_1+u_2}{2} - p_1$
C	θ_2	p_2	$\frac{u_3+u_4+u_5}{3} - p_2$
D	θ_2	p_2	$\frac{u_3+u_4+u_5}{3} - p_2$
E	θ_2	p_2	$\frac{u_3+u_4+u_5}{3} - p_2$
F	θ_3	p_3	$u_6 - p_3$
G	θ_0	$p_0(=0)$	0

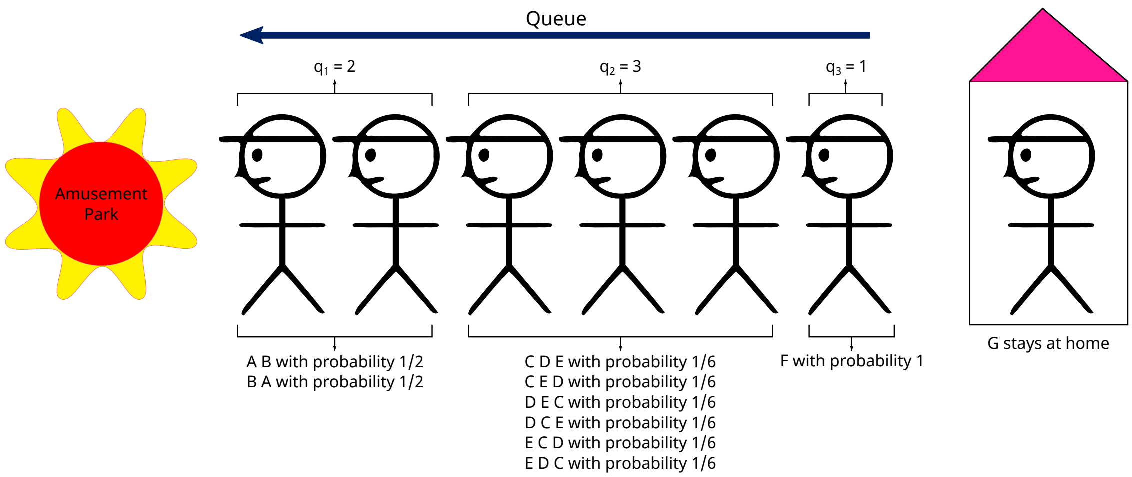

Figure 1: Example of a scheme: $(q_0, q_1, q_2, q_3) = (1, 2, 3, 1)$.

3 Preliminaries

3.1 Pass Utility Function

We begin this section with an equivalent formulation of implementability based on incentive constraints. We use this formulation throughout the paper to verify implementability as it helps to make our discussions more intuitive. Towards this objective, we first define the utility from a deviation. Fix (N, K, u) and a scheme $q \in \mathcal{Q}(N, K)$. For $j, k \in \{0, \ldots, K\}$ such that $q_j > 0$, define $v(\theta_k; \theta_j; q)$ to be the utility (before payment) that a customer who would buy θ_j in scheme q receives if she instead buys θ_k.[8] We call v the **pass-utility function constructed from** u. When without ambiguity, such as when the scheme in consideration is

[8]That is, we define $v(\theta_k; \theta_j; q) := \overline{v}_i(a)$, where a is any action profile such that $\overline{q}_j(a) = q_j - 1$, $\overline{q}_k(a) = q_k + 1$, and $\overline{q}_l(a) = q_l$ for every l such that $l \neq j, k$ and $a_i = \theta_k$.

fixed, q is omitted and $v(\theta_k; \theta_j)$ is written instead. Abuse notation to write $v(\theta_k) := v(\theta_k; \theta_k)$ for each $k \in \{0, 1, \dots, K\}$. In words, $v(\theta_k)$ denotes the utility of a customer choosing θ_k in a scheme. The following observation about the pass utility function is fundamental to the reduction of incentive constraints (as well as the lack of it), which we formalize later.

Claim 1 (Properties of pass-utility function)**.** *Fix* (N, K, u) *and* $q \in \mathcal{Q}(N, K)$*. Fix* $k \in \{1, \dots, K\}$*. If* $j_1, j_2 \in \{1, \dots, k-1\}$ *and* $l_1, l_2 \in \{k+1, \dots, K, 0\}$*, then*

$$v(\theta_k; \theta_{j_1}) = v(\theta_k; \theta_{j_2}) > v(\theta_k) > v(\theta_k; \theta_{l_1}) = v(\theta_k; \theta_{l_2}).$$

To explain the intuition, first consider the case where $l_1, l_2 \neq 0$. The first equality implies that the utility of a downgrade does not depend on "how much" higher priority the higher-priority pass has than the lower-priority pass; similarly, the second equality implies that the utility of an upgrade does not depend on "how much" lower priority the lower-priority pass has than the higher-priority pass. To see this, consider a scheme with three priority passes. When a customer in the second priority pass upgrades to the first pass, the distribution of positions in the first pass after the switch (equal probability on the first and the second positions) is the same as when a customer in the third pass upgrades to the first pass.

Meanwhile, the two inequalities in Claim 1 indicate that a downgrade improves the utility of the lower-priority pass and an upgrade lowers the utility of the higher-priority pass. When a customer downgrades to a lower-priority pass, the first position of that pass improves by one, with the last position unchanged, leading to an improved average utility; in an upgrade, the first position of the higher-priority pass is unchanged, and its last position moves down by one, lowering the average utility of positions in that pass.

Lastly, the case where $l_1 = 0$ or $l_2 = 0$ shows that joining a priority pass from outside the queue is like upgrading to this priority pass from a lower-priority pass, which decreases the utility of the higher-priority pass.

3.2 Individual Rationality, Incentive Compatibility, and Externality

For fixed (N, K, u), pick $q \in \mathcal{Q}(N, K)$ and a price vector p. For every $j \in \{1, \dots, K\}$, (p, q) is said to satisfy the **individual-rationality constraint of** θ_j (henceforth IR_j) if every customer buying θ_j in q has no incentive to leave the queue, i.e.,

$$v(\theta_j) - p_j \geq u_0 = 0. \tag{IR_j}$$

Let the **set of IR constraints** be the collection of IR_j over $j \in \{1, \dots, K\}$. For every $j \in \{0, 1, \dots, K\}$ such that $q_j > 0$ and $k \in \{1, \dots, K\}$, (p, q) is said to satisfy the **incentive-compatibility constraint from** θ_j **to** θ_k (henceforth IC_{jk}) if every customer choosing θ_j in q has no incentive to switch to θ_k, i.e.,

$$v(\theta_j) - p_j \geq v(\theta_k; \theta_j) - p_k. \tag{IC$_{jk}$}$$

Let the **set of IC constraints** be the collection of IC_{jk} over $j \in \{0, 1, \dots, K\}$ such that $q_j > 0$ and $k \in \{1, \dots, K\}$.[9] With these definitions, implementability can be characterized with respect to IC and IR constraints.

Claim 2 (Implementation with respect to incentive constraints)**.** *A scheme $q \in \mathcal{Q}(N, K)$ is implementable if and only if there exists a price vector p such that (p, q) satisfies every constraint in the set of IC and IR constraints.*

Fix an IC constraint IC_{jk} where $j, k \in \{1, \dots, K\}$. The constraint IC_{jk} is said to be a **downward IC constraint** if $j < k$, a **local downward IC constraint** if $k = j + 1$, an **upward IC constraint** if $j > k$, and a **local upward IC constraint** if $j = k + 1$. In standard screening models such as Guesnerie and Laffont (1984) and Maskin and Riley (1984), for implementability, it is sufficient for a price vector to satisfy every local downward IC constraint and IR_K.[10] Similarly, we will show that for each downward IC constraint, there are local downward IC constraints that together imply it: IC_{jk} holds if $\text{IC}_{l,l+1}$ holds for every $l \in \{j, \dots, k-1\}$. Unlike in the standard screening models, however, in our model, no non-local upward IC constraint is implied by any combination of local upward IC constraints (We will detail this point shortly). The following lemma summarizes the constraint reduction results in our model and provides a useful condition for proving some negative results about implementability.

Lemma 1 (Constraint reduction)**.** *Fix $q \in \mathcal{Q}(N, K)$ and a price vector $p \in \mathbb{R}^K_+$. Fix $j, k \in \{1, \dots, K\}$ such that $j < k$.*

(a) If (p, q) satisfies $IC_{l,l+1}$ for every $l \in \{j, \dots, k-1\}$, then (p, q) satisfies IC_{jk}.

[9] Since $q_j > 0$ holds for every $j \in \{1, \dots, K\}$ if q is a scheme, the condition of $q_j > 0$ is relevant only when $j = 0$.

[10] Carroll (2012) provided a comprehensive discussion about when local incentive constraints are sufficient for global incentive constraints. Our setup is most closely related to the case with transfers and interdependent preferences in his paper. Carroll (2012) showed that in that case, if each agent's utility is linear in the reported type and each agent's type space is convex, then local incentives are sufficient. The conditions cover screening studies like Guesnerie and Laffont (1984) and Maskin and Riley (1984), but not ours because the linearity condition does not hold in our model.

(b) If (p,q) satisfies IR_k and IC_{jk}, then (p,q) satisfies IR_j.

(c) If (p,q) satisfies both IC_{jk} and IC_{kj}, then

$$v(\theta_j;\theta_k) - v(\theta_k) \leq v(\theta_j) - v(\theta_k;\theta_j). \tag{ID$_{jk}$}$$

The reduction of downward IC constraints and the lack of it for upward IC constraints can be understood by using the inequalities in Claim 1. Pick $j,k \in \{1,\ldots,K\}$ such that $j+1<k$, i.e., j and k are at least two apart. The downward IC constraint IC_{jk} implies an upper bound on $p_j - p_k$ as follows:

$$\begin{aligned} p_j - p_k \leq v(\theta_j) - v(\theta_k;\theta_j) &= v(\theta_j) - v(\theta_k) - \underbrace{[v(\theta_k;\theta_j) - v(\theta_k)]}_{\text{Downgrade externality}} \\ &= v(\theta_j) - v(\theta_k) - \underbrace{[v(\theta_k;\theta_{k-1}) - v(\theta_k)]}_{\text{Local downgrade externality}}, \end{aligned}$$

where the last inequality follows from Claim 1. We call the two differences in the square brackets the downgrade externality and the local downgrade externality, respectively. The existence of the downgrade externalities decreases the upper bound $p_j - p_k$ further. If we bind and combine all the local downward IC constraints between θ_j and θ_k, the resulting upper bound on $p_j - p_k$ would decrease from $v(\theta_j) - v(\theta_k)$ by the sum of the local downgrade externalities (i.e., $\sum_{l=j+1}^{k} v(\theta_l;\theta_{l-1}) - v(\theta_l)$). The sum of these local downgrade externalities is more than the downgrade externality in IC_{jk} $(v(\theta_k;\theta_j) - v(\theta_k))$ because by Claim 1, $v(\theta_k;\theta_j) = v(\theta_k;\theta_{k-1})$ holds for every $j,k \in \{1,\ldots,K\}$ such that $j<k$. Therefore, binding every local downward IC constraint between θ_j and θ_k satisfies IC_{jk}, hence the simplification for downward IC constraints.

The analogous simplification for the upward IC constraints, however, does not hold because an upward IC constraint provides a lower bound, not an upper bound, of the price difference. Specifically, IC_{kj} implies

$$\begin{aligned} p_j - p_k \geq v(\theta_j;\theta_k) - v(\theta_k) &= v(\theta_j) - v(\theta_k) - \underbrace{[v(\theta_j) - v(\theta_j;\theta_k)]}_{\text{Upgrade externality}} \\ &= v(\theta_j) - v(\theta_k) - \underbrace{[v(\theta_j) - v(\theta_j;\theta_{j+1})]}_{\text{Local upgrade externality}}, \end{aligned}$$

where the last equality again follows from Claim 1. Binding all the local upward IC constraint between θ_k and θ_j decreases the lower bound on $p_j - p_k$ by the sum of the local upgrade externalities (i.e., $\sum_{l=j}^{k-1} v(\theta_l) - v(\theta_l;\theta_{l+1})$). The sum of these externalities decreases the lower bound on $p_j - p_k$ by more than what is allowed by IC_{kj}. Therefore, binding every local upward IC constraint between θ_k and θ_j violates IC_{kj}, hence the lack of simplification for upward IC constraints.

3.3 Increasing Difference

A scheme q is said to satisfy ID_{jk} if the inequality (ID_{jk}) in Lemma 1 holds. Let the set of **increasing difference (ID)** conditions be the collection of ID_{jk} for $j, k \in \{1, \dots, K\}$ such that $j < k$. Note that ID_{jk} holds if IC_{jk} and IC_{kj} hold simultaneously. Thus, the set of ID conditions is necessary for implementability. Specifically, IC_{jk}, under which every customer holding θ_j has no incentive to downgrade to θ_k, implies an upper bound on $p_j - p_k$, whereas IC_{kj}, under which every customer holding θ_k has no incentive to upgrade to θ_j, implies a lower bound on $p_j - p_k$. Implementability necessitates that the upper bound be weakly higher than the lower bound, hence ID_{jk}.

Note that (ID_{jk}) can be manipulated as:

$$\underbrace{v(\theta_k;\theta_j) - v(\theta_k)}_{\text{Downgrade externality}} \leq \underbrace{v(\theta_j) - v(\theta_j;\theta_k)}_{\text{Upgrade externality}}.$$

In words, an ID condition is equivalent to the requirement that the upgrade externality is no less than the downgrade externality.

The ID conditions are commonly assumed in the screening literature. In existing studies such as Guesnerie and Laffont (1984) and Maskin and Riley (1984), the ID conditions are necessary and sufficient for implementability if every IC constraint is implied by some combination of local IC constraints.[11] In our model, while the ID conditions are necessary for implementability as part (c) of Lemma 1 shows, they are not sufficient for implementability (we show this in Theorem 3). The only special case where the ID conditions are necessary and sufficient for implementability is when $K = 2$, every customer buys some pass, and $v(\theta_2) \geq 0$. This is because that is the only case where every IC constraint is a local constraint and every IR constraint holds with some price vector. For $K \geq 3$, the ID conditions are not sufficient for implementability. For example, when the base utility function is linear, the ID conditions hold (Proposition 1), but in this case, no schemes with more than two priority passes are implementable (Theorem 2). Therefore, implementability calls for more conditions, which we will discuss next.

[11] See Chapter 2 of Bolton and Dewatripont (2005) for a summary of these results.

4 Implementability

This section discusses the implementability of schemes, where our general message is that implementability is hard to obtain in our setting with externalities. For this purpose, we formalize our results under different patterns of base utility functions. Section 4.1 introduces three types of base utility functions and explains their applications. Section 4.2 examines implementability under each of these three possibilities for the base utility function. Section 4.3 considers a special scheme that was shown to be optimal in the queuing literature (in a different setting) and shows that it is not implementable under any base utility function in our setting.

4.1 Examples of Base Utility Functions

Some of our results pay special attention to the following three patterns of base utility functions. As we argue below, each of these patterns has reasonable applications.

- Concave base utility function ($u_n - u_{n+1} \leq u_{n+1} - u_{n+2}$ for each n): The concave case applies when queuing incurs an opportunity cost to a customer. Specifically, a customer's utility from a park depends on the time spent in the park. Assume the customer has a total amount of available time $T > 0$. If the customer spends $x \in [0, T]$ units of time in the park, then her utility is $y(x)$, where $y' > 0$ and $y'' \leq 0$, i.e., the customer enjoys spending time in the park but faces diminishing marginal utility. If being at the n-th position in the queue means n units of wait time, then the customer's utility is $u_n = y(T - n)$, which is decreasing and concave in n.

- Convex base utility function ($u_n - u_{n+1} \geq u_{n+1} - u_{n+2}$ for each n): The convexity assumption is applicable when queuing is inherently unpleasant to a customer, but the customer becomes less sensitive to longer queuing time. Specifically, assume that being at the n-th position in the queue means n units of wait time. The customer obtains a fixed utility U from visiting the park but incurs a disutility of $c(n)$ from queuing when the customer waits for n units of time. We assume $c(\cdot)$ to be increasing and concave to capture the diminishing sensitivity to wait time. In this case, the customer's base utility function is $u_n = U - c(n)$, which is decreasing and convex in n. Additionally, the convexity assumption holds when upon entry, the customer obtains a fixed instantaneous utility that is exponentially discounted with the wait time in the queue. Specifically, assume the customer again obtains a fixed utility $U > 0$ from visiting the park and being at the n-th position in the queue means n units of wait time. Waiting in the queue discounts the utility by rate $r > 0$. The customer's base utility function is then $e^{-r(n-1)}U$ for some $r > 0$, which is decreasing and convex in n.

- Linear base utility function ($u_n - u_{n+1} = u_{n+1} - u_{n+2}$ for each n): The linear case is commonly assumed in queuing literature such as Balachandran (1972), Adiri and Yechiali (1974), and Alperstein (1988). Since a linear function is both convex and concave, some applications of the previous two cases can be adapted to the linear case. For example, assuming $y'' = 0$ in the above concave case and $c'' = 0$ in the above convex case makes the base utility function linear. Additionally, the case with the linear base utility function is arguably the most tractable.

4.2 Implementability and Base Utility Functions

We start with the concave base utility functions. It turns out that if the base utility function is strictly concave, i.e., $u_n - u_{n+1} < u_{n+1} - u_{n+2}$ for every position n, then no scheme with more than one pass and more than two customers is implementable.

Theorem 1 (Implementation with strictly concave utility)**.** *Fix (N, K, u) where $K > 1$ and u is strictly concave. If $q \in \mathcal{Q}(N, K)$ has more than two customers buying a priority pass, then q is not implementable.*

The proof of Theorem 1 shows that when the base utility function is strictly concave, the downgrade externality is always strictly higher than the upgrade externality, violating ID_{jk}, i.e., there exists no price vector that satisfies both IC_{jk} and IC_{kj}. The following example illustrates the intuition.

Example 1 (Strict concavity violates ID)**.** Consider the case where $N = 3$, $K = 2$, and u is strictly concave. Fix a scheme $(q_0, q_1, q_2) = (0, 1, 2)$. Writing both ends of ID_{12} in (ID_{jk}) with respect to the base utility function, we have

$$\frac{u_1 + u_2 + u_3}{3} - \frac{u_2 + u_3}{2} \leq u_1 - \frac{u_1 + u_2}{2},$$

which we solve to get $u_1 - u_2 \geq u_2 - u_3$, contracting the strict concavity assumption of u. Hence, ID_{12} is violated and the scheme is not implementable.

The existence of externalities contributes to the lack of implementability. If there were no externalities, then we would have $v(\theta_k; \theta_j) = v(\theta_k)$ for every j and k. In this hypothetical case, IC_{12} for the aforementioned example implies $p_1 - p_2 \leq u_1 - \frac{u_2+u_3}{2}$ and IC_{21} implies

$p_1 - p_2 \geq u_1 - \frac{u_2+u_3}{2}$, making the scheme implementable with $p_1 - p_2 = u_1 - \frac{u_2+u_3}{2}$, as in the standard screening models.[12]

In contrast to the concave case, when u is convex, i.e., $u_n - u_{n+1} \geq u_{n+1} - u_{n+2}$ for every position n, all ID conditions are satisfied. Moreover, when u is convex and $K = 2$, the (unique) ID condition is equivalent to the existence of a price vector with which the two IC constraints hold. These facts imply the following result.

Proposition 1 (Two-pass implementation with convex utility)**.** *Fix (N, K, u) where $K = 2$ and u is a convex base utility function. Every $q \in \mathcal{Q}(N, 2)$ such that $v(\theta_2) \geq 0$ is implementable.*

The contrast between Theorem 1 and Proposition 1 shows that implementability depends on the shape of u. It turns out that to check the implementability of a scheme, it is necessary and sufficient to check whether the price vector binding all local downward IC constraints and the lowest IR constraint implements the scheme. The next lemma makes this point and provides further simplification.

Lemma 2 (Implementability condition)**.** *Fix (N, K, u) where $1 \leq K \leq N$. Let $q \in \mathcal{Q}(N, K)$ be such that $q_0 = 0$ and $v(\theta_K) \geq 0$. Let p^* be the price vector such that $p^*_K = v(\theta_K)$ and $p^*_k - p^*_{k+1} = v(\theta_k) - v(\theta_{k+1}; \theta_k)$ for every $k \in \{1, \ldots, K-1\}$. The scheme q is implementable if and only if (p^*, q) satisfies every upward IC constraint in $\{IC_{Kj} : 1 \leq j \leq K-1\}$.*

Lemma 1 shows that if IR_K and every local downward IC constraint hold, then all downward IC and IR constraints hold. Hence, to check that p^* (which by definition satisfies IR_K and every local downward IC constraint) implements q, it only remains to check whether (p^*, q) satisfies every upward IC constraint. Lemma 2 says that it suffices to check only the subset of upward IC constraints in the lemma's statement.

In the standard screening models, with a fixed scheme q, when p^* as in Lemma 2 binds every local downward IC constraint and IR_K, every IC and IR constraint holds thanks to the ID conditions and constraint reduction. However, in our model, no combination of local upward IC constraints implies a non-local upward IC constraint, and this is why p^* in Lemma 2 does not always implement the scheme q. The lemma also illustrates how a scheme fails to be implementable. For every $j, k \in \{1, \ldots, K\}$ such that $j < k$,

$$p^*_j - p^*_k = \sum_{l=j}^{k-1} p^*_l - p^*_{l+1} = v(\theta_j) - v(\theta_k) - \sum_{l=j}^{k+1} \underbrace{[v(\theta_{l+1}; \theta_l) - v(\theta_{l+1})]}_{\substack{\text{Downgrade externality} \\ \text{from } \theta_l \text{ to } \theta_{l+1}}}, \tag{1}$$

where p^* is defined in Lemma 2. The summation on the right-hand side is the accumulated externalities from downgrading to the next-lower-priority pass, which can be interpreted as

[12] We will have more in-depth discussion on this hypothetical case in Section 6.2.

the extra surplus the park gives to each customer in θ_j (relative to customers in θ_k). At the same time, IC_{kj} is equivalent to

$$p_j - p_k \geq v(\theta_j;\theta_k) - v(\theta_k) = v(\theta_j) - v(\theta_k) - \underbrace{[v(\theta_j) - v(\theta_j;\theta_{j+1})]}_{\substack{\text{Upgrade externality} \\ \text{from } \theta_{j+1} \text{ to } \theta_j}}, \tag{2}$$

where the difference in the bracket is the upgrade externality from θ_{j+1} to θ_j, which is interpreted as the allowance for the surplus the park can give to each customer in θ_j. Whenever the surplus given to customers in θ_j, expressed as the sum of downgrade externalities in (1), exceeds the surplus allowance, which is the upgrade externality in (2), the scheme is not implementable. Note that if additionally $k < K$, from $p^*_j - p^*_k$ to $p^*_j - p^*_{k+1}$, the extra surplus given to customers in θ_j (relative to customers in θ_{k+1}) increases, whereas the surplus allowance stays the same by Claim 1. Therefore, if $\text{IC}_{k+1,j}$ holds with p^*, then $\text{IC}_{k,j}$ also holds.

When $K \geq 3$, not every scheme is implementable even if the base utility function is convex. For example, the next theorem shows that when the base utility function is linear, which is a special case of convexity, no schemes with more than two passes are implementable.

Theorem 2 (Implementation with linear utility)**.** *Fix (N, K, u), where u is linear. Fix $q \in \mathcal{Q}(N, K)$ such that $v(\theta_K) \geq 0$. The scheme q is implementable if and only if $K \leq 2$.*

The proof shows that when $K \geq 3$, IC_{31} implies $p_1 - p_3 \geq v(\theta_1) - v(\theta_3) - \frac{d}{2}$ where $d = u_1 - u_2$, but binding IC_{12} and IC_{23} implies $p_1 - p_3 = v(\theta_1) - v(\theta_3) - d$, arriving at a contradiction. By (1), IC_{12} and IC_{23} implies that the park needs to give a minimum amount of surplus d (i.e., $u_1 - u_2$) to customers in θ_1. This surplus, however, exceeds the surplus allowance in (2), which is $\frac{d}{2}$ in this case. In words, the externalities created by downgrades would lower p_1 relative to p_3 so much that customers buying θ_3 have an incentive to upgrade to θ_1.

4.3 One Customer in Each Priority Pass

In Alperstein (1988), it is profit-maximizing for the park to have at most one customer in each priority pass. The following result, which holds for any base utility function u, shows that such a scheme is not implementable with more than two priority passes.

Theorem 3 (Implementation with $N = K$)**.** *Fix (N, K, u) and $q \in \mathcal{Q}(N, K)$. If $K > 2$ and $q_k = 1$ for every $k \in \{1, \ldots, K\}$, then q is not implementable.*

The result is shown by contradiction, comparing the sum of downgrade externalities and the surplus allowance given by the upgrade externality: Suppose that such a scheme is implementable and consider the incentives of the first three customers, each of whom buys a different priority pass. For the first customer, it is tempting to switch to θ_2 from θ_1 since the customer pays less but still has the chance of being at the same position after switching. To incentivize the first customer against downgrading, $p_1 - p_2$ needs to be small. Similarly, $p_2 - p_3$ needs to be small so that the second customer does not want to downgrade to θ_3 from θ_2. However, to eliminate the third customer's incentive to upgrade to θ_1 from θ_3, the park also needs to set $p_1 - p_3$ to be high enough. The proof shows that the upper bound on $p_1 - p_3$ derived from the upper bounds on $p_1 - p_2$ and $p_2 - p_3$ (the sum of downgrade externalities) is strictly less than the lower bound on $p_1 - p_3$ (surplus allowance implied by the upgrade externality), which is a contradiction.

5 Two Utility Types

So far, we have considered the case where there is a single base utility function for all buyers. Name this case the **single-type case**. Now consider the case where there are two types of base utility functions, which we call the **two-type case**. This section discusses how this heterogeneity affects the implementability of a scheme. We will show that although the heterogeneity in the base utility functions sometimes makes it possible to implement a scheme with more than two passes as Six Flags does, the conflict between the upgrade and downgrade incentives can persist.

We first introduce the setup of the two-type case in Section 5.1 and then provide some analyses of implementability (and the lack of it) for the case with two concave or linear base utility functions in Section 5.2. Overall, we find that implementability requires the two customer types to be sufficiently different. In Section 5.3, we discuss how the number of customers in each priority pass affects implementability. Lastly, in Section 5.4, we briefly describe some new insights we obtain if there are more than two customer types, such as the possibility of the park incentivizing some customer types not to buy any priority pass.

5.1 Notation and Setup

There are two customer types, h (high) and l (low). Denote the high type's base utility function by u^h and that of the low type's by u^l. For $t \in \{h, l\}$, let $N^t > 0$ be the number of type-t customers and define $N = N^h + N^l$ to be the total number of customers. Assume $u_0^l = u_0^h = 0$ for $t \in \{h, l\}$ and $u_n^h > u_n^l$ for every $n \in \{1, \dots, N\}$. Moreover, for every

$n \in \{1, \ldots, N-1\}$, assume $u_n^h - u_{n+1}^h > u_n^l - u_{n+1}^l$. For each customer type $\tau \in \{h, l\}$, construct the type-specific pass utility function v^t from u^t.

For q to be a scheme, we require that every priority pass has at least one customer, that is, $q_k^h + q_k^l > 0$ for every $k \in \{1, \ldots, K\}$, where q_k^τ denotes the number of customers of type τ in θ_k. Given (N, K), let $\mathcal{Q}((N^h, N^l), K)$ denote the set of schemes.[13] Given $j \in \{0, \ldots, K\}$ such that $q_j^\tau > 0$ and $k \in \{1, \ldots, K\}$, let IC_{jk}^t and IR_k^t be the type-specific IC and IR constraints, whose definitions are taken directly from the single-type case.[14]

Fix $\big((N^h, N^l), K\big)$ and a scheme $q \in \mathcal{Q}((N^h, N^l), K)$. Fix $j \in \{0, \ldots, K\}$ such that $q_j^h + q_j^l > 0$[15] and $k \in \{1, \ldots, K\}$. If $q_j^h, q_j^l > 0$, then because both IC_{jk}^h and IC_{jk}^l provide upper bounds on $p_j - p_k$, one of the two upper bounds must imply the other. The same reasoning goes for every IR constraint. Define $\text{IC}_{jk} \in \{\text{IC}_{jk}^\tau : \tau \in \{h, l\}, q_j^\tau > 0\}$ to be the type-specific IC constraint that implies every constraint in $\{\text{IC}_{jk}^\tau : \tau \in \{h, l\}, q_j^\tau > 0\}$.[16] Define $\text{IR}_k \in \{\text{IR}_k^h, \text{IR}_k^l\}$ analogously.[17] Let the set of IC constraints be the collection of IC_{jk} over $j \in \{0, \ldots, K\}$ and $k \in \{1, \ldots, K\}$ such that $q_j^h + q_j^l > 0$; let the set of IR constraints be the collection of IR_k over $k \in \{1, \ldots, K\}$. With this notation, we can define implementability in the two-type case with respect to the set of IR and IC constraints just like in the single-type case.

5.2 Implementability in the Two-Type Case

Now we consider implementability in the two-type case. We first define the notions of monotonicity and regularity in Section 5.2.1 that we use in stating our results. Then Sections 5.2.2 and 5.2.3 characterize the environments in which multi-pass schemes are implementable in the two-type case where the base utility function is strictly concave and linear, respectively. We do not pursue analogous results in the case with strictly convex base utility function because the results from Section 6.3 on large queue implementation suggest that the strictly convex case could admit many priority passes in the single-type case, which would make it hard to derive conditions about type differences for implementability like we will do for the strictly concave and linear cases.

[13]We provide the formal definition of a scheme for the case with multiple customer types in Appendix B.

[14]We provide a more detailed setup for the type-specific constraints in Appendix B, where we discuss the case with more than two customer types.

[15]This inequality may not hold if $j = 0$.

[16]When IC_{jk} is a downward IC constraint, IC_{jk} is equivalent to IC_{jk}^l if $q_j^l > 0$ and to IC_{jk}^h if not; when IC_{jk} is an upward IC constraint, IC_{jk} is equivalent to IC_{jk}^h if $q_j^h > 0$ and to IC_{jk}^l if not.

[17]We provide the complete characterization of this constraint reduction between customer types in Appendix B, where we discuss the general case with multiple customer types.

5.2.1 Monotonicity and Regularity

The added customer heterogeneity introduces some complications for constraint reduction. In the single-type case, Lemma 1 shows that it is sufficient to check every local downward IC constraint and IR_K for implementability. In general, in the two-type case, we cannot reduce the set of downward IC and IR constraints to the set of local downward IC constraints and IR_K, but there are sufficient conditions to guarantee that such a reduction works. Specifically, constraint reduction similar to that in the single-type case can be obtained if we require the scheme in consideration to be such that $q^l_j > 0$ implies $q^l_k > 0$ for every $j, k \in \{1, \ldots, K\}$ such that $j \leq k$.[18] This restriction eliminates schemes where a lower-priority pass has only high-type customers, but a higher-priority pass has at least one low-type customer. With this restriction, we can similarly check the implementability of a given scheme by binding the lowest IR constraint and every local downward IC constraint, as we did in Lemma 2.[19] It turns out that the restriction is necessary for implementability with two concave utilities. To be precise, implementability with two concave base utility functions implies a condition stronger than the restriction which we call the monotonicity property: We say that a scheme q is **monotone** if, for every $j, k \in \{1, \ldots, K\}$ such that $j < k$, $q^l_j > 0$ implies $q^h_k = 0$.

Proposition 2 (Monotonicity with concave utilities)**.** *Fix $\big((N^h, N^l), K, (u^h, u^l)\big)$ in the two-type case and assume both u^h and u^l are concave. If $q \in \mathcal{Q}\big((N^h, N^l), K\big)$ is implementable, then q is monotone.*

The monotonicity property does not generally hold in our setup because of externalities. Consider the following example.

Example 2 (High-type in lower priority)**.** Consider $\big((N^h, N^l), K, (u^h, u^l)\big)$ in the two-type case, where $N^h = 2$, $N^l = 1$, and $K = 2$. Let $u^l = (11, 1, 0)$ and $u^h = (14, 3, 1)$ be the base utility functions of the two types. Consider the scheme $q = (q^h, q^l)$ such that $(q^h_0, q^h_1, q^h_2) = (0, 0, 2)$ and $(q^l_0, q^l_1, q^l_2) = (0, 1, 0)$. It can be verified that setting $p_1 = 9$ and $p_2 = 2$ implements the scheme. In this scheme, the high-type customers buy the lower-priority pass while the low-type customer buys the higher-priority pass. Calculations are shown in Figure 2. In this example, monotonicity fails because when a high type attempts to buy the higher-priority pass, the customer creates congestion to this pass, which lowers the high-type valuation of the high-priority pass so much that the high-type customer buying θ_2 has no incentive to switch to θ_1.

In the standard screening model, a "monotonicity" condition would also require that if a lower type buys some priority pass then all higher types buy some priority pass. The next

[18] See Lemma 4 and Lemma 5 in Appendix A.3.1 for the formal statements of the conditions.

[19] See Lemma 6 in Appendix A.3.1 for the formal statement.

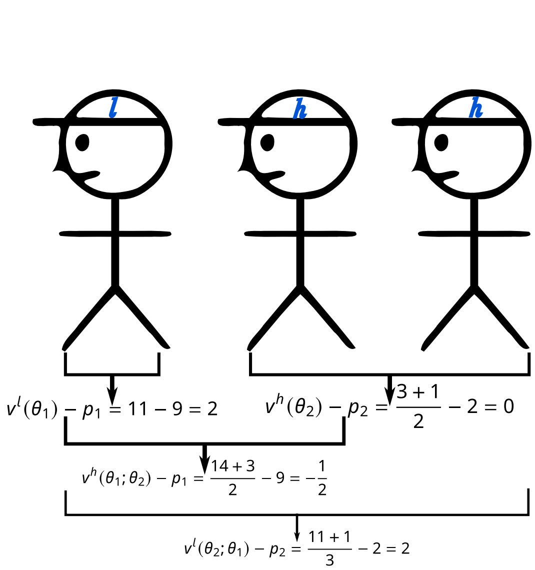

Figure 2: Example 2 where high type buys some strictly lower-priority pass than low type.

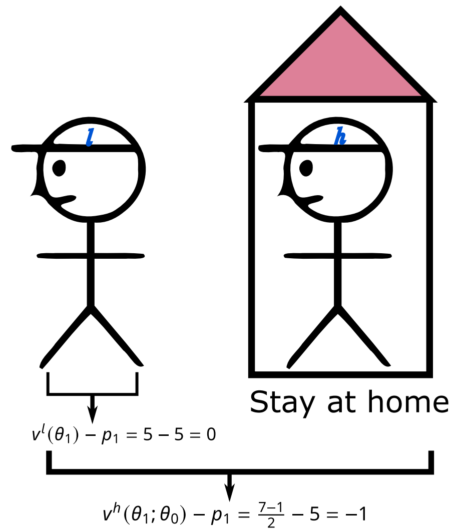

Figure 3: Example 3 where low type buys some pass, but high type does not.

example illustrates that this property does not hold in our model even if the utility function is concave (it is a two-position example).

Example 3 (High-type not buying)**.** In the two-type case, let $\big((N^h, N^l), K, (u^h, u^l)\big)$ be such that $N^h = N^l = 1$ and $K = 1$. Let $u^l = (5, 0)$ and $u^h = (7, 1)$ be the base utilities of the two types. Consider the scheme q with only the low-type customer in the queue, i.e., the high-type customer buys no priority pass. It can be verified that $p_1 = 5$ implements q, with calculations shown in Figure 3. In this example, monotonicity fails because, given the price, the second position is bad enough even for the high type, incentivizing the high-type customer who is staying at home not to join the queue.

When monotonicity does not hold, by definition either the high type does not buy any pass, or a high type buys a strictly lower-priority pass. In Example 3, a low type buys some pass, but the high type does not buy any pass; in Example 2, the low type buys the higher-priority pass, whereas the high type buys the lower-priority pass.

We now define further terminology used in the results in the rest of this section. If a two-type case $((N^h, N^l), K, (u^h, u^l))$ is such that $u^h = \beta u^l$ for some $\beta > 1$, then we call this case the **multiplicative** two-type case. A two-type case is **concave** if both base utility functions are concave, and **linear** if the functions are linear.

Definition 2 (Regular scheme)**.** In the two-type case, a scheme $q \in \mathcal{Q}((N^h, N^l), K)$ is said to be **regular** if it satisfies the following conditions:

(i) Every customer buys some priority pass, and $v^t(\theta_K) \geq 0$ for every t such that $q^t_K > 0$.

(ii) Each customer type is in at most two priority passes.

(iii) The scheme q is monotone.

Condition (i) lets us ignore the IR constraints and focus on IC constraints. Condition (ii) is necessary for implementability in the concave two-type case by Theorems 1 and 2.[20] Lastly, condition (iii) is the monotonicity condition, which is necessary for implementability in the concave two-type case by Proposition 2.

In what follows, we consider the strictly concave case and the linear case in turn.

5.2.2 Strictly Concave Case

In the strictly concave single-type case, an implementable scheme can have at most one priority pass. When there are two customer types, upper hemicontinuity of pure-strategy Nash equilibria for utility functions implies that when the two customer types are only slightly different, a two-pass scheme with two strictly concave base utility functions is still not implementable. However, if the two types are sufficiently different, we can separate the two types. The following proposition provides a cutoff of this closeness for a fixed scheme so that separation is feasible if and only if the two customer types are no closer than the cutoff.

Proposition 3 (Two-pass with two strictly concave utilities)**.** *Consider the strictly concave multiplicative two-type case where $K > 1$ and $N > 2$. There exists $\underline{\beta} > 1$ such that a regular scheme q is implementable if and only if $K = 2$, $q_1^h = N^h$, $q_2^l = N^l$, and $\beta \geq \underline{\beta}$.*

In the strictly concave multiplicative two-type case, the only implementable regular scheme must have every high-type customer in the high-priority pass and every low-type customer in the low-priority pass. The condition that $K = 2$ is necessary since, by Theorem 1, a strictly concave customer type can be in at most one priority pass. The condition that every high-type customer buys the high-priority pass and every low-type customer buys the low-priority pass is implied by Proposition 2.

We now turn to the cutoff on β, which characterizes the two customer types' difference in the proposition. Consider a scheme q such that $q_1^h = N^h$ and $q_2^l = N^l$. We can generalize Lemma 2 to the two-type case to show that q is implementable if and only if the following price vector implements q:

$$p_2 = v^l(\theta_2), \quad p_1 = v^h(\theta_2) - \left[v^h(\theta_2; \theta_1) - p_2\right] = v^h(\theta_2) - \left[v^h(\theta_2; \theta_1) - v^l(\theta_2)\right],$$

[20] One further implication is that each implementable scheme in the concave two-type case has at most four priority passes.

where the term in the bracket on the right-hand side of p_1 is the surplus given to the high-type customers.[21] In this price vector, each high-type customer holding the high-priority pass is indifferent between the two passes, whereas each low-type customer holding the low-priority pass gets zero surplus. This pattern is consistent with the optimal pricing formula in the standard screening models. However, the existence of externalities plays a role in the above pricing formula, leading to different results of implementability. To see this, note that the price vector implies,

$$p_1 - p_2 = v^h(\theta_1) - v^h(\theta_2;\theta_1) = v^h(\theta_1) - v^l(\theta_2) - \overbrace{\Big[\underbrace{\big(v^h(\theta_2;\theta_1) - v^h(\theta_2)\big)}_{\text{Downgrade externality}} + \underbrace{\big(v^h(\theta_2) - v^l(\theta_2)\big)}_{\substack{\text{Information rent in} \\ \text{standard screening}}}\Big]}^{\text{Total surplus giveaway}}, \quad (3)$$

where the second parenthesized difference is often called the information rent of the high-type customers in the standard screening models. In our model, the additional term $v^h(\theta_2;\theta_1) - v^h(\theta_2)$ is the externality created when a high-type customer unilaterally downgrades to the low-priority pass. The externality is strictly positive and hence the park needs to give away more surplus to the higher-type customer, which further lowers p_1. To check whether q is implementable, note that IC_{21} of q implies

$$p_1 - p_2 \geq v^l(\theta_1;\theta_2) - v^l(\theta_2) = v^h(\theta_1) - v^l(\theta_2) - \overbrace{\Big[\underbrace{\big(v^l(\theta_1) - v^l(\theta_1;\theta_2)\big)}_{\text{Upgrade externality}} + \big(v^h(\theta_1) - v^l(\theta_1)\big)\Big]}^{\text{Surplus allowance}}, \quad (4)$$

where the first parenthesized difference in the bracket is the externality created when the low-type customer upgrades to the high-priority pass. Since $v^h(\theta_1) - v^l(\theta_1) > v^h(\theta_2) - v^l(\theta_2) > 0$, in the standard screening models without externalities, the price difference from the pricing formula would satisfy IC_{21}, making q implementable.[22] However, from the proof of Theorem 1, when the base utility functions are strictly concave and $N > 2$, $v^h(\theta_2;\theta_1) - v^h(\theta_2) > v^h(\theta_1) - v^h(\theta_1;\theta_2) > v^l(\theta_1) - v^l(\theta_1;\theta_2)$, which could potentially make the price difference in (3) violate IC_{21}. For example, in the hypothetical case where $u^h = u^l$, we obtain $v^h(\theta_1) - v^l(\theta_1) = v^h(\theta_2) - v^l(\theta_2) = 0$. In this case, the right-hand side of (3) is strictly below the right-hand side of (4), making the scheme not implementable as in Theorem 1. In the two-type case, as the customer type difference gets larger, e.g., β in Proposition 3 gets larger, both the total surplus giveaway in (3) and the surplus allowance in (4) increase. However, because the allowance grows faster than the giveaway, the scheme

[21] The generalization appears in Lemma 6 in Appendix A.3.1.

[22] The inequality holds because $u^h_n - u^h_{n+1} > u^l_n - u^l_{n+1} > 0$ for every position n.

becomes implementable with large enough β. Lastly, to see that $\underline{\beta} > 1$ in Proposition 3 is necessary, we observe that when $\beta \leq 1$, $v^h(\theta_2) - v^l(\theta_2) \leq 0$ holds, and hence (4) does not hold with the price difference in (3).

5.2.3 Linear Case

In the linear single-type case, an implementable scheme can have at most two priority passes. As in the strictly concave case, the upper hemicontinuity implies that at most two priority passes can be implemented in the linear multiplicative two-type case if the two types are sufficiently close to each other. Again, as in the strictly concave case, however, it is possible to implement a scheme with more than two priority passes when the types are sufficiently apart from each other. The following result characterizes the implementability condition as a requirement on the difference between the two customer types.

Theorem 4 (Multi-pass scheme with two types)**.** *Fix a linear multiplicative two-type case where $K > 2$. For each regular scheme q, there exists $\underline{\beta} > 1$ such that q is implementable if and only if $\beta \geq \underline{\beta}$.*

Note that in the concave two-type case, the regularity condition excludes the case where $K > 4$. Towards a straightforward intuition for the theorem, consider the special case of the linear multiplicative two-type case where $K = 3$ and there are only high-type customers in the first two priority passes and only low-type customers in the lowest-priority pass. Assume also that the scheme has exactly m customers in each priority. In this case, IC_{31} holds if and only if

$$\underbrace{\frac{\beta}{2}(2m-1)}_{\substack{\text{Upper bound of} \\ p_1 - p_2}} + \underbrace{\frac{\beta}{2}(2m-1)}_{\substack{\text{Upper bound of} \\ p_2 - p_3}} \geq \underbrace{\frac{1}{2}(4m-1)}_{\substack{\text{Lower bound of} \\ p_1 - p_3}} \tag{5}$$

The inequality holds if and only if β is sufficiently larger than 1, hence the condition in the theorem that the two customer types be adequately different. The intuition is similar to that of Proposition 3: A larger difference between the two customer types enlarges the price difference between θ_1 and θ_3 when IC_{12} and IC_{23} are binding, giving customers in θ_3 less incentive to upgrade to θ_1.

5.3 Queue Size and Implementability

Note that, for fixed $\beta > 1$, (5) also holds if m is sufficiently large: In the linear multiplicative two-type case, more customers lead to higher price differences, making it easier to satisfy IC_{31}. One could conjecture that a scheme is implementable with sufficiently many customers in each priority pass. We formalize this conjecture in the following two results. The first result confirms that, in the linear multiplicative two-type case, having more customers helps with implementation.

Proposition 4 (Sufficiently many customers for implementation)**.** *Fix a linear multiplicative two-type case where $K \leq 4$. There exists $\underline{M} < \infty$ such that every regular scheme q with at least $\underline{M}$ customers in every priority pass is implementable. Moreover, if $\beta \geq 2$, then the previous statement holds for $\underline{M} = 1$.*

When there are sufficiently many customers in each priority pass, the price difference between a higher and lower priority pass is large enough to eliminate the upgrade incentives of customers in the lower-priority pass. It turns out that when $\beta \geq 2$, the two customer types are sufficiently different such that every IC constraint holds even if there is only one customer in each pass.

Proposition 4 does not clarify how large the required number of customers ($\underline{M}$) is to obtain implementability. The following result shows that when the number of customers in each priority pass of a scheme is bounded, the scheme is not implementable when the two customer types are too close.

Proposition 5 (Customer types are too close for implementation)**.** *Fix a linear multiplicative two-type case where $K > 2$. For each $\overline{m} \geq 1$, there exists $\delta > 0$ such that no regular scheme with at most $\overline{m}$ customers in each priority pass is implementable if $\beta < 1 + \delta$.*

One implication of the result is that as the two customer types get closer towards 1, $\underline{M}$ in Proposition 4 gets unboundedly higher. In this case, the price differences between different passes are not large enough to resolve the customers' upgrade and downward incentives.

5.4 Extension to Multiple Customer Types

Results in the two-type case can be extended to the case with more than two customer types, which we call the **multi-type case**. Because the two cases have similar core intuitions, we relegate the formal analysis of the multi-type case to Appendix B.

In the multi-type case where each base utility function is concave, implementability needs customer types in different priority passes to be sufficiently different, which is a generalization of the implementability condition in Theorem 4.[23] For example, if every customer type is strictly concave and buys some pass, which is a generalized case of Proposition 3, then the existence of an implementable multi-pass scheme implies the existence of a large enough "gap" between customer types in two adjacent priority passes.

[23] The required conditions are formally characterized in Theorem 5 in Appendix B.

Propositions 4 and 5, both of which can be generalized to the multi-type case, together have some insights about the number of passes the park can implement when there are more than two customer types. By Proposition 5, a scheme with many priority passes is not implementable if customer types from different priority passes are too close. On the other hand, given a scheme that is not implementable, if we "merge" some priority passes such that the number of customers in each priority pass of the new scheme is large, then by Proposition 4, the new scheme with a reduced number of priority passes may be implementable.

One complication of the implementability condition that customer types in different priority passes need to be sufficiently different is that the park could potentially exclude some customer types from the queue to create the "gaps" between customer types in different priorities. For example, suppose there are five customer types (ordered by their utility decrease along a queue), with each adjacent pair of customer types being very close with regard to their rate of utility decrease. If the park would like to implement a three-pass scheme, the park may consider excluding the second and the fourth customer types from the queue so that there is enough difference between the customer types remaining in the queue. However, through a monotonicity result that generalizes Proposition 2, we show that there are restrictions to this type of customer exclusion,[24] which further strengthens the difficulty with implementing multi-pass schemes.

6 Discussions

Section 6.1 looks at the park's profit. We first explain how the one-pass scheme maximizes the park's profit in the single-type case, and then revisit the two-type case to show that a multi-pass scheme can be more profitable than a single-pass scheme. Section 6.2 discusses how the externality affects implementability in a model that generalizes our assumption about how the purchased pass affects the utility. Section 6.3 discusses implementability in large queues where the size of externalities approaches zero.

6.1 Profits

In the single-type case, having more passes weakly hurts the park's profit: When $K = 1$, and $q_0 = 0$, setting $p_1 = v(\theta_1)$ implements $(q_0, q_1) = (0, N)$ and extracts all the customer surplus. This is because when everyone buys the same priority pass, each customer's valuation of

[24]For example, Proposition 10 and its Corollary 2 in Appendix B show that, if a customer's base utility function decreases strictly faster than that of a customer in the lowest-priority pass in an implementable priority queue, then that customer must have purchased some priority pass.

the priority pass is the same and adds up to the sum of utility over all the positions in the scheme, enabling the park to extract all the customer surplus.

In the strictly concave case, Theorem 1 implies that any implementable scheme has $K = 1$, so the optimal profit is attained under the uniquely implementable scheme. In the linear and strictly convex case, an implementable scheme can have multiple passes. Because of the externality created when a customer downgrades to a lower priority pass, implementing a multi-pass scheme means that the park needs to give some surplus to customers with higher-priority passes, making a multi-pass scheme suboptimal for profits. Thus, in the linear case, in which any implementable scheme must have one or two passes by Theorem 2, having one more priority pass always decreases profits.

Furthermore, in the single-type convex case, for each implementable scheme with K priority passes with $K > 1$, the firm can strictly improve its profit from an implementable scheme with $K - 1$ priority passes.

Proposition 6 (Profit decreasing in K)**.** *Fix (N, K, u) where $K \geq 2$. Assume $q \in \mathcal{Q}(N, K)$ is implementable. If $K = 2$ or u is convex, then there exists $\tilde{q} \in \mathcal{Q}(N, K-1)$ such that $\tilde{q}$ is implementable and the firm's optimal profit (by pricing p^* as in Lemma 2) from $\tilde{q}$ is strictly higher than the profit from q.*

The proposition is proved by showing that the firm's profit always improves by merging the lowest two priority passes. Indeed, conditional on implementability, when the last two priority passes are merged, the park can extract all the surplus of customers in the merged priority pass, improving the park's profit. As for the implementability of merging the last two priority passes, the proof shows that the downgrade externality to the merged priority pass in $\tilde{q}$ is less than the sum of downgrade externalities to the two priority passes in q. Given $j, k \in \{1, \ldots, K\}$ such that $j < k$, by (1) and (2) in Section 4, the implementability of q implies that the upgrade externality to θ_j is more than the sum of downgrade externalities to priority passes from θ_{j+1} to θ_k. Therefore, if the downgrade externality to the merged priority pass in $\tilde{q}$ is less than the sum of downgrade externalities to θ_{K-1} and θ_K in q, then merging the lowest two priority passes preserves the scheme's implementability.[25]

When there is more than one type, however, the single-pass scheme does not always dominate a multi-pass one. For the two-type case with strictly concave utilities, the two-pass scheme could be more profitable than a single-pass scheme. The next proposition focuses on **all-serving** schemes, which we define to be the schemes where every customer of every type

[25] Merging the last two priority passes does not always preserve implementability. For example, fix $N = 6$ and a base utility function $(u_n)_{n=0}^{N} = (95, 62, 60, 59, 1, 0)$. Consider the scheme $(q_k)_{k=0}^{3} = (0, 2, 3, 1)$. It can be calculated that q is implementable, whereas the scheme $\tilde{q}$ obtained by merging the last two priority passes is not.

buys some priority pass.

Proposition 7 (Profitability of two-pass schemes)**.** *Consider the strictly concave two-type case. There exists $\underline{\beta} > 1$ such that the profit from the unique regular two-pass scheme is higher than that of the all-serving one-pass scheme if and only if $\beta \geq \underline{\beta}$.*

Note that the uniqueness of a regular two-pass scheme follows from Proposition 3. When the one-pass scheme serves both types, the park needs to respect the low type's IR constraint, lowering the price. If the two types are sufficiently different, the park will have an incentive to add a higher-priority pass to extract more surplus from the high-type customers. When the park adds a second pass and sells it to the high-type customers, it gets lower profit from the low-type customer. If the two types are too close, then the increase in profit from the high-type customers does not cover the decrease from the low-type customers, making the one-pass scheme more profitable.

6.2 Implementation and Externality

The lack of upward IC constraint reduction makes the implementation of multi-pass schemes non-trivial. In the standard screening models, the ID conditions imply that binding local downward IC constraints satisfies all upward IC constraints, and the local IC constraints together imply all the IC constraints, making the ID conditions sufficient for implementability in the standard setup. However, in our model, as any non-local upward IC constraint is generally not implied by any combination of local upward IC constraints, the ID conditions do not guarantee implementability.

The IC constraints in our model differ from those in the standard screening models because of the existence of externalities from switching: When one customer switches to another pass, the customer creates congestion if switching to a higher-priority pass and improves the waiting time if switching to a lower-priority pass. It turns out that the existence of this type of externalities makes implementation harder than in a model without externalities.

We focus on the single-type case, formalize the notion of externalities in a general model and discuss its impact on implementability. For this purpose, we use the notation in Section 2 and consider a game $\tilde{G}(N, K, p, \tilde{v}) = \langle N, A, (\tilde{\pi}_i)_{i=1}^N \rangle$, where $\tilde{\pi}_i(a) = \tilde{v}(a_i; \overline{q}(a)) - p_k$ is each customer's payoff function over the action set $A = (\{\theta_0, \dots, \theta_K\})^N$.[26] Recall that $\overline{q}(a) = (\overline{q}_k(a))_0^K$ such that $\overline{q}_k(a) = |\{i : a_i = \theta_k\}|$. Here, $\tilde{v}$ tells the expected utility for each $\theta_j \in \{\theta_0, \dots, \theta_K\}$ in each action profile a and only depends on the customer's action a_i and the number of customers for each possible action, which is characterized by $\overline{q}(a)$. The

[26] To complete the definition of $\tilde{v}$, for each θ_j such that $a_i \neq \theta_j$ for every customer i, set $\tilde{v}(\theta_j; \overline{q}(a)) = 0$ without loss of generality.

function $\tilde{v}$ need not be constructed as an expected utility over positions in the queue and we take it as exogenous the provision of the function $\tilde{v}$. The function $\bar{v}$ that we constructed in Section 2 is a special case of $\tilde{v}$. Thus, the game G in Section 2 is a special case of $\tilde{G}$ if we set $\tilde{v}(a_i; \overline{q}(a)) = \overline{v}_i(a)$. The definition of implementation with respect to $\tilde{G}$ is analogous to the one with respect to G.

Definition 3. Fix a game $\tilde{G}(N, K, p, \tilde{v})$. We say that $\tilde{v}$

(a) **creates more downgrade externality** if for each $q \in \mathcal{Q}(N, K)$, every $a \in A$ such that $\overline{q}(a) = q$, and every $j, k \in \{1, \ldots, K\}$ such that $j < k$, $0 < \tilde{v}(\theta_j; \overline{q}(a)) - \tilde{v}(\theta_j; \overline{q}(a')) < \tilde{v}(\theta_k; \overline{q}(a'')) - \tilde{v}(\theta_k; \overline{q}(a))$ for all $a' \in A$ such that there is i with $a_i = \theta_k$, $a'_i = \theta_j$ and $a_n = a'_n$ for every $n \neq i$, and all $a'' \in A$ such that there is i with $a_i = \theta_j$, $a'_i = \theta_k$, and $a_n = a'_n$ for every $n \neq j$.

(b) **creates zero externality** if for every $a, a' \in A$ and $j \in \{0, \ldots, K\}$, $\tilde{v}(\theta_j; \overline{q}(a)) = \tilde{v}(\theta_j; \overline{q}(a'))$, i.e., $\tilde{v}$ only depends on the customer's choice of priority pass.

The definition of creating zero externality corresponds to the setup in the standard screening models, where each customer's utility depends only that customer's action.[27]

In the game $G(N, K, p, u)$ defined in Section 2, if u is strictly concave, then in the game expressed as a special case of $\tilde{G}$, whose setup we have just described, $\tilde{v}$ creates more downgrade externality. It turns out that if $\tilde{v}$ in game $\tilde{G}(N, K, p, \tilde{v})$ creates zero externality, then q is implementable for every $q \in \mathcal{Q}(N, K)$. In contrast, if $\tilde{v}$ creates more downgrade externality and $K \geq 2$, then no scheme is implementable.

Proposition 8 (Externality vs. no externality)**.** *Fix $\tilde{G}(N, K, p, \tilde{v})$ and $q \in \mathcal{Q}(N, K)$.*

(a) If $\tilde{v}$ creates zero externality, then q is implementable by setting $p_k = v(\theta_k)$ for each pass θ_k.

(b) If $\tilde{v}$ creates more downgrade externality and $K \geq 2$, then q is not implementable.

By Part (a), since every pass is implementable when $\tilde{v}$ creates zero externality, the set of implementable schemes when $\tilde{v}$ creates more downgrade externality is a subset of the set when $\tilde{v}$ creates zero externality. Moreover, Part (b) implies that the externalities can strictly shrink the set of implementable schemes. Specifically, when $\tilde{v}$ creates more downgrade externality, the externality created by downgrading to a lower-priority pass are large relative to the externality created by upgrading to a different pass, to the extent that the difference

[27]The definition is equivalent to a condition that is more wordy yet perhaps more comparable to the definition of creating more downgrade externality: For each $j, k \in \{0, \ldots, K\}$ and $a, a' \in A$ such that $a_i = \theta_j$ for some customer i and a' is the strategy profile where i switches to θ_k unilaterally, $\tilde{v}(\theta_k; \overline{q}(a')) = \tilde{v}(\theta_k; \overline{q}(a))$.

brings about an unresolvable conflict between incentivizing against upgrading and against downgrading when $K \geq 2$.

6.3 Implementation in Large Queues

This section analyzes how implementability changes when the queue grows. The implementation results so far depend on the externalities customers impose on each other. By p^* in Lemma 2, a scheme fails to be implementable when the externalities from downgrading are too large: To incentivize the customers in the high-priority pass not to downgrade, to the extent that a customer in a lower-priority pass will have an incentive to upgrade. If the downgrade externalities can be sufficiently small with enough customers, which limits the price decrease of a higher-priority pass, then implementing schemes with many passes is possible.

Proposition 9 (Implementation with fixed K and large N)**.** *Fix K and a strictly decreasing sequence $(u_n)_{n=1}^{\infty}$. Assume $u_0 = -\infty$. If $\lim_{n\to\infty} \frac{u_n}{n} = 0$, then there exists $M < \infty$ such that $N \geq M$ implies the existence of some implementable $q \in \mathcal{Q}(N, K)$.*

The condition $\lim_{n\to\infty} \frac{u_n}{n} = 0$ ensures that the downgrade externality $v(\theta_k; \theta_{k-1}) - v(\theta_k)$ converges to 0 when $q_k \to \infty$ for every $k \in \{2, \dots, K\}$. The condition is mild and admits an unbounded utility function such as $u_n = -\log n$ while implying a rate of decrease slower than that of the linear and concave base utility functions. Proposition 9 shows that a scheme becomes implementable if the externalities are made arbitrarily small (by having sufficiently many customers). This result reinforces the message emphasized throughout this paper: The existence of externalities contributes to the difficulty of implementing schemes with many priority passes.

7 Conclusions

This paper has shown the difficulty of implementing a multi-pass scheme under a static setting where customers make purchase decisions simultaneously and have uncertainty about the final position within each priority pass. The difficulty with implementing many passes derives from the conflict between incentivizing customers against upgrading and downgrading. When the base utility function is strictly concave, implementing a multi-pass scheme cannot be incentive compatible if there is only one customer type. This paper has shown that such incentive conflicts can persist even when there are multiple types of base utility functions.

This paper used a stylized model to deliver clear and concise insights about implementing multi-pass schemes, and there are many avenues for future research. For example, how does implementability change if the model has more dynamic components? The extreme case in which customers arrive sequentially is considered in the queuing literature as we discussed in the Introduction. The reality would probably be in the middle of these two extreme settings. Another topic to study is a profit-maximization problem in the multi-type case. Yet another possibility that would be worth an investigation is a realistic situation in which other perks such as free drinks or better seating are added to the flash pass benefits. We leave the analysis of those setups to the future research.

Appendix A Proofs of Results in Main Text

We define a new notation repeatedly used in this section. In the single-type case, for any (N, K), $q \in \mathcal{Q}(N, K)$, and $k \in \{1, \ldots, K\}$, define $Q_k = \sum_{j=1}^{k} q_j$, which is the last position of θ_k in q. For completeness, define $Q_0 = 0$.

A.1 Proofs for Section 3

A.1.1 Proof of Claim 1

Proof. Fix $k \in \{1, \ldots, K\}$ and consider any $j_1, j_2 \in \{1, \ldots, k-1\}$ and $l_1, l_2 \in \{k+1, \ldots, K, 0\}$. When a customer switches from θ_{j_1} to θ_k, the distribution of her resulting position is uniform over $\{Q_{k-1}, \ldots, Q_k\}$, and this is the same as when the customer switches from θ_{j_2} to θ_k. Therefore, $v(\theta_k; \theta_{j_1}) = v(\theta_k; \theta_{j_2})$. For l_1 and l_2, the same reasoning implies that the distribution is now uniform over $\{Q_{k-1}+1, \ldots, Q_k+1\}$, and thus we have $v(\theta_k; \theta_{l_1}) = v(\theta_k; \theta_{l_2})$. Finally, note that the distribution of the position of the customer buying θ_k is uniform over $\{Q_{k-1} + 1, \ldots, Q_k\}$. Since this last distribution is first-order stochastically dominated by the distribution given by the downgrade while it first-order stochastically dominates the distribution given by the upgrade, we have $v(\theta_k; \theta_{l_1}) < v(\theta_k) < v(\theta_k; \theta_{j_2})$ since u is decreasing. This completes the proof. □

A.1.2 Proof of Claim 2

Proof. Let $a \in A$ be an action profile such that $\overline{q}(a) = q$ and fix a price vector p. We look at the incentives of customer i. If $a_i = \theta_0$, then the customer does not gain by deviating to $\theta_j \neq \theta_0$ if and only if IC_{0j} holds.

Now assume $a_i = \theta_j \neq \theta_0$. The customer does not gain by does not gain by deviating to θ_0 if and only if IR_j holds. Lastly, for every $k \in \{1, \ldots, K\}$ such that $k \neq j$, the customer does not gain by deviating to θ_k if and only if IC_{jk} holds.

Therefore, q is implementable if and only if there exists some price p such that (p, q) satisfies every IC and IR constraint. □

A.2 Proofs for Section 4

A.2.1 Proof of Lemma 1

Proof. Fix $j, k \in \{1, \dots, k\}$ such that $j < k$.

Part (a) Assume $\mathrm{IC}_{l,l+1}$ holds for every $l \in \{j, \dots, k-1\}$. We prove the result by induction. First, if $k = j+1$, we obviously have IC_{jk}. Second, for any $k > j+1$, suppose that $\mathrm{IC}_{j,k-1}$ holds. Then,

$$\begin{aligned} v(\theta_j) - p_j &\geq v(\theta_{k-1}; \theta_j) - p_{k-1} && \text{(by } \mathrm{IC}_{j,k-1}\text{)} \\ &\geq v(\theta_{k-1}) - p_{k-1} && \text{(by Claim 1)} \\ &\geq v(\theta_k; \theta_{k-1}) - p_k && \text{(by } \mathrm{IC}_{k,k-1}\text{)} \\ &= v(\theta_k; \theta_j) - p_k && \text{(by Claim 1).} \end{aligned}$$

Thus, IC_{jk} holds.

Part (b) Assume IC_{jk} and IR_k hold. We have

$$\begin{aligned} v(\theta_j) - p_j &\geq v(\theta_j; \theta_k) - p_k && \text{(by } \mathrm{IC}_{jk}\text{)} \\ &\geq v(\theta_k) - p_k && \text{(by Claim 1)} \\ &\geq 0 && \text{(by } \mathrm{IR}_k\text{).} \end{aligned}$$

Thus, IR_j holds.

Part (c) Assume IC_{jk} and IC_{kj} hold. We have

$$\begin{aligned} v(\theta_j; \theta_k) - v(\theta_k) &\leq p_j - p_k && \text{(by } \mathrm{IC}_{kj}\text{)} \\ &\leq v(\theta_j) - v(\theta_k; \theta_j) && \text{(by } \mathrm{IC}_{jk}\text{).} \end{aligned}$$

Thus, ID_{jk} holds. □

A.2.2 Proof of Theorem 1

Proof. We will use the following lemma to prove the theorem.

Lemma 3. *Fix (N, K, u) where $K \geq 2$. Fix $q \in \mathcal{Q}(N, K)$. If u is concave, then for every $j, k \in \{1, \ldots, K\}$ such that $j < k$,*

$$v(\theta_k; \theta_j) - v(\theta_k) \geq v(\theta_j) - v(\theta_j; \theta_k),$$

and the inequality is strict if u is strictly concave and either $q_j + q_k > 2$ or $j + 1 < k$.

Proof of lemma. For $m \in \{0, \ldots, q_j\}$, define $x_m = u_{Q_j - m + 1}$. Similarly, for $m \in \{0, \ldots, q_k\}$, define $y_m = u_{Q_{k-1}+m}$. By the concavity of u, we have

$$x_{q_j} - x_{q_j - 1} \leq \cdots \leq x_1 - x_0 \leq y_0 - y_1 \leq \cdots \leq y_{q_k - 1} - y_{q_k}.$$

Hence, we have $\sum_{m=1}^{q_j}(x_m - x_0) \leq \frac{q_j(q_j+1)(x_1 - x_0)}{2}$ and $\sum_{m=1}^{q_k}(y_0 - y_m) \geq \frac{q_k(q_k+1)(y_0 - y_1)}{2}$, which together imply

$$\frac{\sum_{m=1}^{q_j}(x_m - x_0)}{q_j(q_j+1)} \leq \frac{x_1 - x_0}{2} \leq \frac{y_0 - y_1}{2} \leq \frac{\sum_{m=1}^{q_k}(y_0 - y_m)}{q_{j+1}(q_{j+1}+1)}. \tag{6}$$

The left-most side of (6) being no greater than the right-most side is equivalent to

$$\frac{\sum_{m=0}^{q_j} x_m}{q_j + 1} - \frac{\sum_{m=1}^{q_k} y_m}{q_k} \geq \frac{\sum_{m=1}^{q_j} x_m}{q_j} - \frac{\sum_{m=0}^{q_k} y_m}{q_k + 1},$$

which is equivalent to $v(\theta_k; \theta_j) - v(\theta_k) \geq v(\theta_j) - v(\theta_j; \theta_k)$.

If u is strictly concave, the first inequality in (6) is strict if $q_j > 1$, the second inequality is strict if $j + 1 < k$, and the last inequality is strict if $q_k > 1$. Thus, at least one inequality must be strict. The proof for the lemma is complete. □

Assume $q \in \mathcal{Q}(N, K)$ for some $K > 1$ is implementable and more than two customers buy a priority pass in q. To arrive at a contradiction, it is without loss of generality to assume $q_0 = 0$ because if q is implementable with respect to (N, K, u), then when we subtract q_0 from N, the scheme in which the customers who stay at home under q are removed is still implementable.

When $K > 1$, for each $q \in \mathcal{Q}(N, K)$ where more than two customers buy a priority pass, $K > 2$ or $q_l > 1$ for some $l \in \{1, \ldots, K\}$. In both cases, Lemma 3 implies that if u is strictly concave, then we can find some $j, k \in \{1, \ldots, K\}$ such that $j < k$ and $v(\theta_k; \theta_j) - v(\theta_k) > v(\theta_j) - v(\theta_j; \theta_k)$. This contradicts ID_{jk} and hence q is not implementable. □

A.2.3 Proof of Proposition 1

Proof. Let $q \in \mathcal{Q}(N,2)$ such that $v(\theta_2) \geq 0$. For $n \in \{0,\ldots,q_1\}$, define $x_n = u_{Q_1-n+1}$. Similarly, for $n \in \{0,\ldots,q_2\}$, define $y_n = u_{Q_1+n}$. By the definition of the base utility function,

$$x_{q_1} > \cdots > x_1 = y_0 > x_0 = y_1 > \cdots y_{q_2}.$$

By the convexity of u, we have

$$x_{q_1} - x_{q_1-1} \geq \cdots \geq x_1 - x_0 = y_0 - y_1 \geq \cdots \geq y_{q_2-1} - y_{q_2},$$

which implies $\sum_{n=1}^{q_1}(x_n - x_0) \geq \frac{q_1(q_1+1)(x_1-x_0)}{2}$ and $\sum_{n=1}^{q_2}(y_0 - y_n) \leq \frac{q_2(q_2+1)(y_0-y_1)}{2}$, which together imply

$$\frac{\sum_{m=1}^{q_1}(x_m - x_0)}{q_1(q_1+1)} \geq \frac{x_1 - x_0}{2} = \frac{y_0 - y_1}{2} \geq \frac{\sum_{m=1}^{q_2}(y_0 - y_m)}{q_2(q_2+1)}. \tag{7}$$

Consider the price vector p such that

$$p = (p_1, p_2) = (v(\theta_2) + v(\theta_1) - v(\theta_2;\theta_1), v(\theta_2)).$$

Because $v(\theta_2) \geq 0$, p is a valid price vector. We are to show that p implements q. By definition of p, (p,q) binds IC_{12} and IR_2. By Lemma 1, (p,q) satisfies IR_1. Note that (p,q) satisfies IC_{21} if and only if $p_1 - p_2 \geq v(\theta_1;\theta_2) - v(\theta_2)$, which by the definition of p is equivalent to

$$v(\theta_1) - v(\theta_1;\theta_2) \geq v(\theta_2;\theta_1) - v(\theta_2). \tag{8}$$

With our definition of $\{x_n\}_{n=1}^{q_1}$ and $\{y_n\}_{n=1}^{q_2}$, (8) is equivalent to

$$\frac{\sum_{n=1}^{q_1} x_n}{q_1} - \frac{\sum_{n=0}^{q_1} x_n}{q_1+1} \geq \frac{\sum_{n=0}^{q_2} y_n}{q_2+1} - \frac{\sum_{n=1}^{q_2} y_n}{q_2}.$$

This is equivalent to

$$\frac{\sum_{n=1}^{q_1}(x_n - x_0)}{q_1(q_1+1)} \geq \frac{\sum_{n=1}^{q_2}(y_0 - y_n)}{q_2(q_2+1)},$$

which holds by (7). Therefore, IC_{21} holds.

Since IR_2 binds and $v(\theta_2;\theta_0) < v(\theta_2)$ holds by Claim 1, we have

$$v(\theta_2;\theta_0) - p_2 < v(\theta_2) - p_2 = 0,$$

and hence IC_{02} holds. For IC_{01}, note that

$$\begin{aligned} v(\theta_1;\theta_0) - p_1 &= v(\theta_1;\theta_2) - p_1 && \text{(by Claim 1)} \\ &= v(\theta_1;\theta_2) - v(\theta_2) - [v(\theta_1) - v(\theta_2;\theta_1)] && \text{(by the definition of } p\text{)} \\ &\leq 0 && \text{(by (8))} \end{aligned}$$

Thus, IC_{01} holds. Since we have now shown that all the constraints hold, we conclude that q is implementable. □

A.2.4 Proof of Lemma 2

Proof. First, $v(\theta_K) \geq 0$ implies that p^* is a valid price vector. Because $q_0 = 0$, IC_{0k} for any $k \in \{1, \ldots, K\}$ is not defined. By the definition of p^*, (p^*, q) binds IR_K. It also binds every local downward IC constraint and hence every downward IC constraint holds by part (a) of Lemma 1. For each $k \in \{1, \ldots, K-1\}$, since IR_K and IC_{kK} holds, IR_k holds by part (b) of Lemma 1.

The "if" part The argument so far implies that, if (p^*, q) satisfies every upward IC constraint, p^* implements q. If $K = 2$, then there is only one upward IC constraint and therefore p^* implements q if and only if IC_{21} holds. Assume $K > 2$ and fix $j, k \in \{1, \ldots, K-1\}$ such that $j < k$. If (p^*, q) satisfies IC_{kj}, then (2) holds with p being set to p^*. We then plug (1) into (2) to get

$$v(\theta_j) - v(\theta_j;\theta_{j+1}) \geq \sum_{l=j}^{k-1} \left[v(\theta_{l+1};\theta_l) - v(\theta_{l+1}) \right]. \tag{9}$$

Note that the left-hand side of this inequality is independent of k as long as $j < k$. Similarly, (p^*, q) satisfies $\text{IC}_{k+1,j}$ if and only if

$$v(\theta_j) - v(\theta_j;\theta_{j+1}) \geq \sum_{l=j}^{k} \left[v(\theta_{l+1};\theta_l) - v(\theta_{l+1}) \right]. \tag{10}$$

Since the right-hand side of (10) is larger than that of (9) by Claim 1, $\text{IC}_{k,j}$ holds whenever (p^*, q) satisfies $\text{IC}_{k+1,j}$. Therefore, if IC_{Kj} holds for every $j \in \{1, \ldots, K-1\}$, then every upward IC constraint holds. Therefore, if (p^*, q) satisfies IC_{Kj} for every $j \in \{1, \ldots, K-1\}$, then p^* implements q and hence, q is implementable.

The "only if" part Now assume that q is implementable and let p be a price vector that implements q. Fix $j, k \in \{1, \ldots, K\}$ such that $j < k$. We have

$$\begin{aligned} p_j - p_k &= \sum_{l=j}^{k-1} p_l - p_{l+1} \\ &\leq \sum_{l=j}^{k-1} v(\theta_l) - v(\theta_{l+1}; \theta_l) && \text{(because } (p,q) \text{ satisfies IC}_{l,l+1}) \\ &= \sum_{l=j}^{k-1} p_l^* - p_{l+1}^* && \text{(by the definition of } p^*) \\ &= p_j^* - p_k^*. \end{aligned}$$

Thus, (p^*, q) satisfies IC_{kj}. Because IC_{kj} is an arbitrary upward IC constraint, (p^*, q) satisfies every upward IC constraint and hence, in particular, (p^*, q) satisfies every upward IC constraint in $\{\mathrm{IC}_{Kj} : 1 \leq j \leq K-1\}$. □

A.2.5 Proof of Theorem 2

Proof. Fix (N, K, u) where u is linear. Fix $q \in \mathcal{Q}(N, K)$ with $v(\theta_K) \geq 0$. If $K \leq 2$, by Proposition 1, every $q \in \mathcal{Q}(N, K)$ such that $v(\theta_K) \geq 0$ is implementable since u is convex. Assume instead $K \geq 3$. Let $d = u_1 - u_2 > 0$. Note that by the linear setup, $u_n = u_1 - (n-1)d$ for every $n \in \{1, \ldots, N\}$. Therefore, for every $l \in \{1, \ldots, K\}$,

$$\begin{aligned} v(\theta_l) = \frac{\sum_{n=Q_{l-1}+1}^{Q_l} u_n}{q_l} &= \frac{\sum_{n=Q_{l-1}+1}^{Q_l} [u_1 - (n-1)d]}{q_l} \\ &= \frac{\sum_{n=1}^{Q_l - Q_{l-1}} [u_1 - (Q_{l-1} + n - 1)d]}{q_l} \\ &= u_1 - \left[Q_{l-1} + \frac{q_l - 1}{2} \right] d. \end{aligned} \tag{11}$$

Fix $j, k \in \{1, \ldots, K\}$ such that $j < k$. To calculate $v(\theta_k; \theta_j)$, we set q_l to $q_k + 1$ and Q_{l-1} to $Q_{k-1} - 1$ in (11) to get

$$v(\theta_k; \theta_j) = u_1 - \left[Q_{k-1} - 1 + \frac{q_k}{2} \right] = v(\theta_k) + \frac{d}{2}. \tag{12}$$

To calculate $v(\theta_j; \theta_k)$, we set q_l to $q_j + 1$ and Q_{l-1} to Q_{j-1} to get

$$v(\theta_j; \theta_k) = u_1 - \left[Q_{j-1} + \frac{q_j}{2} \right] = v(\theta_j) - \frac{d}{2} \tag{13}$$

If p implements q, then IC_{12} implies $p_1 - p_2 \leq v(\theta_1) - v(\theta_2; \theta_1)$ and $p_2 - p_3 \leq v(\theta_2) - v(\theta_3; \theta_2)$. Adding up the two inequalities, we obtain

$$\begin{aligned} p_1 - p_3 &\leq v(\theta_1) - v(\theta_2; \theta_1) + v(\theta_2) - v(\theta_3; \theta_2) \\ &= v(\theta_1) - v(\theta_3) - d. \qquad \text{(by (13))} \end{aligned} \tag{14}$$

However, IC_{31} implies

$$\begin{aligned} p_1 - p_3 &\geq v(\theta_1; \theta_3) - v(\theta_3) \\ &= v(\theta_1) - v(\theta_3) - \frac{d}{2}, \qquad \text{(by (12))} \end{aligned}$$

which contradicts (14) since $d > 0$. Thus q is not implementable. □

A.2.6 Proof of Theorem 3

Proof. Let $q \in \mathcal{Q}(N, K)$ where $q_k = 1$ for every $k \in \{1, \ldots, K\}$. Using p^* defined in Lemma 2, we obtain

$$p_1^* - p_3^* = \sum_{l=1}^{2} v(\theta_l) - v(\theta_{l+1}; \theta_l) = u_1 - \frac{u_1 + u_2}{2} + u_2 - \frac{u_2 + u_3}{2} = \frac{u_1 - u_3}{2}.$$

However, as $p_1^* - p_3^* < \frac{u_1+u_2-2u_3}{2} = v(\theta_1; \theta_3) - v(\theta_3)$, (p^*, q) does not satisfy IC_{31}, and by Lemma 2, q is not implementable. □

A.3 Proofs for Section 5

In this section we first state and prove lemmas that we use in the proofs for results in Section 5. These lemmas are stated in Section A.3.1. The proofs for results in Section 5 appear in Section A.3.2 onward.

A.3.1 Lemmas for Two-Type Case

Proofs for results in Section 5 often use a generalization of Lemma 2 in the single-type case, which we state in this section as Lemma 6. This lemma in the two-type case itself uses the generalization of the constraint reduction results in the single-type case (parts (a) and (b) of Lemma 1), and we state them as Lemmas 4 and 5. Figure 4 provides a road map of how these lemmas in this section contribute to some of the results in the two-type case.

The following two lemmas provide conditions under which the downward IC and IR constraint reductions are valid in the two-type case.

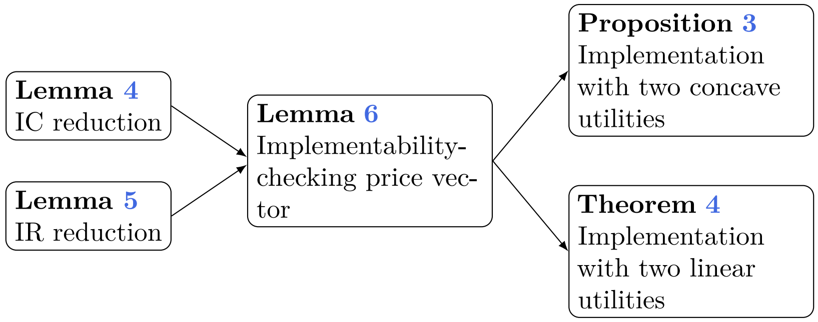

Figure 4: Road map of Appendix A.3.1

Lemma 4 (IC Reduction with two types)**.** *Fix $((N^h, N^l), K, (u^h, u^l))$ with $K \geq 2$, a scheme q, a price vector p, and $j, k \in \{1, \ldots, K\}$ such that $j < k$. Assume that (p, q) satisfies $IC_{m,m+1}$ for every $m \in \{j, \ldots, k-1\}$. If $q^l_j = 0$ or $q^l_m > 0$ for every $m \in \{j, \ldots, k-1\}$, then (p, q) satisfies IC_{jk}.*

Proof. Assume $q^l_j = 0$, which implies that IC_{jk} is equivalent to IC^h_{jk}. For any $m \in \{j, \ldots, k-1\}$, $\text{IC}_{m,m+1}$ implies

$$p_m - p_{m+1} \leq \max_{t \in \{l,h\}} \left[v^t(\theta_m) - v^t(\theta_{m+1}; m)\right] = v^h(\theta_m) - v^h(\theta_{m+1}; \theta_m).$$

Therefore, if $\text{IC}_{m,m+1}$ holds for every $m \in \{j, \ldots, k-1\}$, then

$$\begin{aligned}
p_j - p_k &= \sum_{m=j}^{k-1} (p_m - p_{m+1}) \\
&\leq \sum_{m=j}^{k-1} \left[v^h(\theta_m) - v^h(\theta_{m+1}; \theta_m)\right] && \text{(by IC}_{m,m+1}\text{)} \\
&= v^h(\theta_j) - v^h(\theta_k) - \sum_{m=j}^{k-1} \left[v^h(\theta_{m+1}; \theta_m) - v^h(\theta_{m+1})\right] \\
&\leq v^h(\theta_j) - v^h(\theta_k) - \left[v^h(\theta_k; \theta_{k-1}) - v^h(\theta_k)\right] && \text{(by Claim 1)} \\
&= v^h(\theta_j) - v^h(\theta_k; \theta_j). && \text{(by the proof of Theorem 1)}
\end{aligned}$$

Thus, IC_{jk} holds.

Now assume $q^l_m > 0$ for every $m \in \{j, \ldots, k-1\}$. By the construction of customer types, IC_{jk} is equivalent to IC^l_{jk} and $\text{IC}_{m,m+1}$ is equivalent to $\text{IC}^l_{m,m+1}$. Therefore, by part (a) of Lemma 1, IC_{jk} holds. □

Lemma 5 (IR Reduction with two types)**.** *Fix $((N^h, N^l), K, (u^h, u^l))$ such that $K \geq 2$, $q \in \mathcal{Q}((N^h, N^l), K)$, and a price vector p. Fix $j, k \in \{1, \ldots, K\}$ such that $j < k$ and assume* *that (p,q) satisfies IC$_{jk}$ and IR$_k$. If $q^l_j = 0$ or $q^l_k > 0$, then (p,q) satisfies IR$_j$.*

Proof. If $q^l_j = 0$, then IR$_j$ is equivalent to IR^h_j and IC$_{jk}$ is equivalent to IC$^h_{jk}$. Also, IR$_k$ implies $v^h(\theta_k) - p_k \geq 0$ or $v^l(\theta_k) - p_k \geq 0$, and since the latter implies the former, we have that IR$_k$ implies $v^h(\theta_k) - p_k \geq 0$. Therefore,

$$\begin{aligned} v^h(\theta_j) - p_j &\geq v^h(\theta_k; \theta_j) - p_k && \text{(by IC}_{jk}\text{)} \\ &\geq v^h(\theta_k) - p_k && \text{(by Claim 1)} \\ &\geq 0 && \text{(by IR}_k\text{),} \end{aligned}$$

and thus, IR$_j$ holds.

If $q^l_k > 0$, then IR$_k$ is equivalent to IR^l_k. For any t such that $q^t_j > 0$, we have

$$\begin{aligned} v^t(\theta_j) - p_j &\geq v^t(\theta_k; \theta_j) - p_k && \text{(by IC}^t_{jk}\text{)} \\ &\geq v^l(\theta_k; \theta_j) - p_k && \text{(by the definition of types)} \\ &\geq v^l(\theta_k) - p_k && \text{(by Claim 1)} \\ &\geq 0 && \text{(by IR}_k\text{).} \end{aligned}$$

Thus, IR$_j$ holds. □

The following result extends Lemma 2 in the single-type case and provides a pricing formula to check implementability.

Lemma 6 (Two-type implementation)**.** *Fix $((N^h, N^l), K, (u^h, u^l))$ and $q \in \mathcal{Q}((N^h, N^l), K)$ where every customer buys some pass and $v^l(\theta_K) \geq 0$. Assume for every $j \in \{1, \ldots, K-1\}$, $q^l_j > 0$ implies $q^l_{j+1} > 0$. Let $p^* = (p^*_1, \ldots, p^*_K)$ be such that $p^*_K = v^l(\theta_K)$ and $p^*_j - p^*_{j+1} = v^{\bar{t}_j}(\theta_j) - v^{\bar{t}_j}(\theta_{j+1}; \theta_j)$, where $\bar{t}_j = l$ if $q^l_j > 0$ and otherwise $\bar{t}_j = h$. The scheme q is implementable if and only if (p^*, q) satisfies every upward IC constraint.*

Proof. By the definition of p^*, (p^*, q) satisfies every local downward IC constraint. Pick $j, k \in \{1, \ldots, K\}$ such that $j < k$. If $q^l_j = 0$, then Lemma 4 implies that IC$_{jk}$ holds. If $q^l_j > 0$, then by assumption $q^l_m > 0$ for every $m \in \{j, \ldots, k-1\}$, and hence again by Lemma 4, IC$_{jk}$ holds. Therefore, (p^*, q) satisfies every downward IC constraint. Since $q^l_j > 0$ implies $q^l_{j+1} > 0$ for every $j \in \{1, \ldots, K-1\}$ and every customer buys some pass, we have $q^l_K > 0$. Hence, IR$_K$ is equivalent to IR^l_K, and (p^*, q) satisfies IR$_K$ by the definition of p^*_K. Because in addition IR^l_K holds, by Lemma 5, all the IR constraints also hold. Therefore, q is implementable if (p^*, q) satisfies every upward IC constraint.

Now, assume that q is implementable and let p implement q. Given $j, k \in \{1, \ldots, K\}$ such that $j < k$, (p, q) satisfying IC$_{kj}$ implies $p_j - p_k \geq v^{\underline{t}_k}(\theta_j; \theta_k) - v^{\underline{t}_k}(\theta_k)$, where $\underline{t}_k = h$ if

$q_k^h > 0$ and otherwise $\underline{t}_k = l$. By the definition of customer types, IC_{kj} is equivalent to $\text{IC}_{kj}^{\underline{t}_k}$. On the other hand, we have

$$\begin{aligned}
p_j - p_k &= \sum_{l=j}^{k-1} p_l - p_{l+1} \\
&\leq \sum_{l=j}^{k-1} v^{\bar{t}_l}(\theta_l) - v^{\bar{t}_l}(\theta_{l+1};\theta_l) && \text{(by } \text{IC}_{l,l+1}\text{)} \\
&= \sum_{l=j}^{k-1} p_l^* - p_{l+1}^* = p_j^* - p_k^* && \text{(by definition of } p^*\text{)}
\end{aligned}$$

Therefore, $p_j^* - p_k^* \geq p_j - p_k \geq v^{\underline{t}_k}(\theta_j;\theta_k) - v^{\underline{t}_k}(\theta_k)$, and hence (p^*, q) satisfies IC_{kj}. □

A.3.2 Proof of Proposition 2

Proof. Assume $q \in \mathcal{Q}((N^h, N^l), K)$ is implementable and fix $j, k \in \{1, \ldots, K\}$ such that $j < k$, $q_j^l > 0$ and $q_k^h > 0$ (If there are no such j and k, the proof is done). By the definition of customer types, IC_{jk} is equivalent to IC_{jk}^l and IC_{kj} is equivalent to IC_{kj}^h. Note that

$$\begin{aligned}
v^l(\theta_j) - v^l(\theta_k;\theta_j) &< v^h(\theta_j) - v^h(\theta_k;\theta_j) && \text{(by the definition of types)} \\
&\leq v^h(\theta_j;\theta_k) - v^h(\theta_k), && \text{(by the proof of Theorem 1)}
\end{aligned}$$

However, if a price vector satisfies both IC_{jk}^l and IC_{kj}^h, then we would have

$$v^h(\theta_j;\theta_k) - v^h(\theta_k) \leq p_j - p_k \leq v^l(\theta_j) - v^l(\theta_k;\theta_j),$$

which is a contradiction. Thus, for every $j, k \in \{1, \ldots, K\}$ such that $j < k$, if $q_j^l > 0$, then $q_k^h = 0$. □

A.3.3 Proof of Proposition 3

Proof. Since conditions of Lemma 6 hold, we can check the implementability of q by p^* in the lemma. If $K > 2$, then in every $q \in \mathcal{Q}((N^h, N^l), K)$, at least one customer type has customers in two different priority passes, which makes q unimplementable by Theorem 1.

Assume instead $K = 2$ and fix a regular scheme q. Because $q_0^h = q_0^l = 0$ by the definition of regular schemes, $q_k^h < N^t$ for any $k \in \{1, 2\}$ implies $q_j^h > 0$ for $j \in \{1, 2\}$ such that $j \neq k$, which makes q unimplementable by Theorem 1. Similarly, $q_k^l < N^l$ for any $k \in \{1, 2\}$ implies $q_j^l > 0$ for $j \in \{1, 2\}$ such that $j \neq k$, which again makes q unimplementable. If $q_1^l = N^l$ and

$q_2^h = N^h$, the scheme is not implementable by Proposition 2. Therefore, if a regular scheme $q \in ((N^h, N^l), K)$ for $K > 1$ is implementable, then $K = 2$, $q_1^h = N^h$, and $q_2^l = N^l$.

Now consider the regular scheme $q \in ((N^h, N^l), 2)$ where $q_1^h = N^h$ and $q_2^l = N^l$. With p^* defined in Lemma 6, IC$_{21}$ holds when $p_1^* - p_2^* \geq v^l(\theta_1; \theta_2) - v^l(\theta_2)$. By definition, $p_1^* - p_2^* = \beta^h [v(\theta_1) - v(\theta_2; \theta_1)]$ and $v^l(\theta_1; \theta_2) - v^l(\theta_2) = v(\theta_1; \theta_2) - v(\theta_2)$. Therefore, IC$_{21}$ holds if

$$\beta^h [v(\theta_1) - v(\theta_2; \theta_1)] \geq v(\theta_1; \theta_2) - v(\theta_2),$$

which holds if and only if $\beta^h \geq \frac{v(\theta_1;\theta_2)-v(\theta_2)}{v(\theta_1)-v(\theta_2;\theta_1)}$. We define the right-hand side of the inequality to be $\underline{\beta}$. Lastly, by the proof of Theorem 1, the strict concavity of u implies $\underline{\beta} > 1$ when $N > 2$. □

A.3.4 Proof of Theorem 4

Proof. First assume q is implementable. By Proposition 2, conditions of Lemma 6 hold and p^* defined in the lemma implements q. For (p^*, q) to satisfy IC$_{kj}$ for $j, k \in \{1, \ldots, k\}$ such that $j < k$, we have $v^{\underline{t}_k}(\theta_j; \theta_{j+1}) - v^{\underline{t}_k}(\theta_k) \leq p_j^* - p_k^*$, where $\underline{t}_k = h$ if $q_k^h > 0$ and otherwise $\underline{t}_k = l$. The definition of p^* implies that $p_j^* - p_k^* = \sum_{l=j}^{k-1} v^{\bar{t}_l}(\theta_j) - v^{\bar{t}_l}(\theta_{l+1}; \theta_l) = \sum_{l=j}^{k-1} \beta^{\bar{t}_l} [v(\theta_l) - v(\theta_{l+1;\theta_l})]$, where each $\bar{t}_l$ is defined in Lemma 6. Therefore, IC$_{kj}$ and the definition of p^* imply

$$v^{\underline{t}_k}(\theta_j; \theta_{j+1}) - v^{\underline{t}_k}(\theta_k) \leq \sum_{l=j}^{k-1} \beta^{\bar{t}_l} [v(\theta_l) - v(\theta_{l+1;\theta_l})] \tag{15}$$

Since q is implementable, by Proposition 2, if $\underline{t}_k = h$, then it is necessary that $\bar{t}_l = h$ for every l in (15). Moreover, because by Theorem 2, a customer type can be in at most two priority passes in an implementable scheme, if $\underline{t}_k = h$, then $j = k - 1$ and hence IC$_{kj}$ is equivalent to IC$_{k,k-1}$, a local upward IC constraint, which holds by the ID conditions with respect to u^h.

If $\underline{t}_k = l$ and $\bar{t}_j = l$, then again by Theorem 2, $j = k - 1$, and IC$_{kj}$ is a local upward IC constraint with respect to u^l, which holds by the ID conditions. If $\underline{t}_k = l$ and $\bar{t}_m = h$ for some $m \in \{j, \ldots, k-2\}$, then $\beta^{\bar{t}_m} = \beta$ and IC$_{kj}$ in (15) is a lower bound on β. Therefore, every non-local upward IC constraint provides a lower bound on β. Let $\underline{\beta}$ be the highest lower bound on β over all the lower bounds implied by the non-local upward IC constraints, and hence $\beta \geq \underline{\beta}$ if q is implementable. To see that $\underline{\beta} > 1$, consider IC$_{31}$. Since each customer type can be in at most two priority passes in an implementable scheme with concave base utility functions, Proposition 2 implies that $\bar{t}_1 = h$ and $\underline{t}_3 = l$. In this case, by the definition of p^*, $p_1^* - p_3^* = \beta [v(\theta_1) - v(\theta_2; \theta_1)] + \beta^{\bar{t}_2} [v(\theta_2) - v(\theta_3; \theta_2)]$. Since $\underline{t}_3 = l$, IC$_{31}$ is equivalent to IC^l_{31}, i.e., $v(\theta_1;\theta_2) - v(\theta_3) \le p^*_1 - p^*_3$. Therefore, we have

$$v(\theta_1;\theta_2) - v(\theta_3) \le \beta\left[v(\theta_1) - v(\theta_2;\theta_1)\right] + \beta^{\bar{t}_2}\left[v(\theta_2) - v(\theta_3;\theta_2)\right], \tag{16}$$

which implies a lower bound on β. Because $\beta \ge \beta^{\bar{t}_2} \ge 1$, if $\beta = 1$, then $\beta^{\bar{t}_2} = 1$ and (16) is equivalent to

$$v(\theta_1) - v(\theta_1;\theta_2) \ge v(\theta_2;\theta_1) - v(\theta_1) + v(\theta_3;\theta_2) - v(\theta_3),$$

which does not hold when the base utility function is concave by the proof of Theorem 1 by Theorem 2, (16) does not hold when $\beta = 1$. Therefore, the lower bound on β implied by (16) must be strictly larger than 1, and hence $\underline{\beta} > 1$.

For the other direction, assume $\beta < \underline{\beta}$. Pick j and k such that IC_{kj} implies a lower bound on β and the lower bound is $\underline{\beta}$. However, this would imply that (15) would not hold, i.e., (p^*, q) does not satisfy IC_{kj}. Hence q is not implementable by Lemma 6. Therefore, q is implementable if and only if $\beta \ge \underline{\beta}$. □

A.3.5 Proof of Proposition 4

Proof. The proposition is implied by Proposition 11 in Appendix B for the general multi-type case. To see this, note that every regular scheme in the two-type case is also regular in the multi-type case. Using the notation in Appendix B, $\underline{R}(q) = \beta$ for every regular scheme q in the linear multiplicative two-type case where $K > 2$. Therefore, by Proposition 11 in Appendix B for the multi-type case, we can find some $\underline{M}$ such that every regular scheme in the two-type case is implementable if every priority pass has at least $\underline{M}$ customers. Lastly, by Proposition 11, when $\beta \ge 2$, then $\underline{R}(q) \ge 2$ for every regular scheme q in the linear multiplicative two-type case. Therefore, when $\beta \ge 2$, every regular scheme in the linear multiplicative two-type case is implementable. □

A.3.6 Proof of Proposition 5

Proof. The proposition is an implication of Proposition 12 in Appendix B for the linear multiplicative multi-type case. To see this, note that every regular scheme in the two-type case is regular in the multi-type case. Using the notation in Appendix B, in the linear multiplicative two-type case where $K > 2$, $\underline{R}(q) = \beta$ for every regular scheme q. Therefore, for each $\overline{m} \ge 1$, we can find some $\delta > 0$, such that no regular scheme q with at most $\overline{m}$ customers in each priority pass is implementable if $\beta < 1 + \delta$. □

A.4 Proofs for Section 6

A.4.1 Proof of Proposition 6

Proof. Assume $q \in \mathcal{Q}(N, K)$ is implementable. Consider the scheme $\tilde{q} \in \mathcal{Q}(N, K-1)$, which is obtained by merging the last two priority passes in q is implementable. We are to show first that $\tilde{q}$ is implementable, and then that the profit from $\tilde{q}$, by the price vector in Lemma 2, is higher than the profit from q.

If $K = 2$, the result is clear. Assume $K > 2$ and u is convex. For each symbol, we denote that the symbol is defined with respect to $\tilde{q}$ by accenting it with the tilde sign. For example, $\tilde{\mathrm{IC}}_{jk}$ is the IC constraint between θ_j and θ_k defined with respect to $\tilde{q}$. Symbols without the tilde sign are defined with respect to q.

Note that for every $j \in \{1, \ldots, K-1\}$, $k \in \{1, \ldots, K-2\}$, and $l \in \{1, \ldots, K-3\}$,

$$v(\theta_j) = \tilde{v}(\theta_j) \tag{17}$$

$$v(\theta_k) - v(\theta_k; \theta_{k+1}) = \tilde{v}(\theta_k) - \tilde{v}(\theta_k; \theta_{k+1}) \tag{18}$$

$$v(\theta_{l+1}; \theta_l) - v(\theta_{l+1}) = \tilde{v}(\theta_{l+1}; \theta_l) - \tilde{v}(\theta_{l+1}) \tag{19}$$

By Lemma 2, $\tilde{q}$ is implementable if $\tilde{\mathrm{IC}}_{K-1,j}$ holds with $\tilde{p}^*$ for every $j \in \{1, \ldots, K-2\}$ and $\tilde{\mathrm{IC}}_{0k}$ holds for every $k \in \{1, \ldots, K-1\}$. Plugging (1) into (2), we see that (p^*, q) satisfies IC_{kj} if

$$v(\theta_j) - v(\theta_j; \theta_{j+1}) \geq \sum_{l=j}^{K-1} \left[v(\theta_{l+1}; \theta_l) - v(\theta_{l+1})\right]. \tag{20}$$

Similarly, $\tilde{\mathrm{IC}}_{K-1,j}$ holds with $\tilde{p}^*$ if and only if

$$\tilde{v}(\theta_j) - \tilde{v}(\theta_j; \theta_{j+1}) \geq \sum_{l=j}^{K-2} \left[\tilde{v}(\theta_{l+1}; \theta_l) - \tilde{v}(\theta_{l+1})\right],$$

which by (18) and (19) is equivalent to

$$v(\theta_j) - v(\theta_{j+1}) \geq \sum_{l=j}^{K-3} \left[v(\theta_{l+1}; \theta_l) - v(\theta_{l+1})\right] + \tilde{v}(\theta_{K-1}; \theta_{K-2}) - \tilde{v}(\theta_{K-1}) \tag{21}$$

Comparing (20) and (21), we see that $(\tilde{p}^*, \tilde{q})$ satisfies $\tilde{\mathrm{IC}}_{K-1,j}$ if (p^*, q) satisfies IC_{Kj} and

$$\tilde{v}(\theta_{K-1}; \theta_{K-2}) - \tilde{v}(\theta_{K-1}) < \left[v(\theta_{K-1}; \theta_{K-2}) - v(\theta_{K-1})\right] + \left[v(\theta_K; \theta_{K-1}) - v(\theta_K)\right]. \tag{22}$$

Let $x_n = u_{Q_{K-2}+n}$ for $n \in \{1, \ldots, q_{K-1}\}$ denote the positions of θ_{K-1} in q. Similarly, let

$y_m = u_{Q_{K-1}+m}$ for $m \in \{1, \ldots, q_K\}$ denote the positions of θ_K in q. Lastly, define $x_0 = u_{Q_{K-2}}$. With these new notations, by the definition of the pass-utility functions, we have

$$\begin{aligned}\tilde{\Delta} :=& \tilde{v}(\theta_{K-1}; \theta_{K-2}) - \tilde{v}(\theta_{K-1}) - [v(\theta_{K-1}; \theta_{K-2}) - v(\theta_{K-1})] - [v(\theta_K; \theta_{K-1}) - v(\theta_K)] \\ =& \frac{\sum_{n=1}^{q_{K-1}} (x_0 - x_n) + \sum_{m=1}^{q_K} (x_0 - y_m)}{(q_{K-1} + q_K)(q_{K-1} + q_K + 1)} - \frac{\sum_{n=1}^{q_{K-1}} (x_0 - x_n)}{q_{K-1}(q_{K-1} + 1)} - \frac{\sum_{m=1}^{q_K} (x_{q_{K-1}} - y_m)}{q_K(q_K + 1)}.\end{aligned}$$

Consider the maximization of $\tilde{\Delta}$ subject to the constraint

$$x_0 - x_1 \geq x_1 - x_2 \geq \cdots \geq x_{q_{K-1}} - y_1 \geq y_1 - y_2 \geq \cdots y_{q_K - 1} - y_{q_K} \geq 0,$$

which is implied by the definition and convexity of u. Since $\frac{\partial \tilde{\Delta}}{\partial x_0} < 0$ and $\frac{\partial \tilde{\Delta}}{\partial x_n} > 0$ for $n \in \{1, \ldots, q_{K-1} - 1\}$, $x_0 - x_1 = x_1 - x_2 = \cdots = x_{q_{K-1}-1} - x_{q_{K-1}}$ at maximum. Similarly, since $\frac{\partial \tilde{\Delta}}{\partial y_m} > 0$ for $m \in \{1, \ldots, q_K\}$, we have $x_{q_{K-1}} = y_1 = \cdots = y_{q_K}$ at maximum. Define $d := x_0 - x_1 = \cdots = x_{q_{K-1}-q} - x_{q_{K-1}}$ at maximum. Therefore, at maximum, the value of $\tilde{\Delta}$ is

$$\begin{aligned}\tilde{\Delta}^* :=& \frac{\sum_{n=1}^{q_{K-1}} (x_0 - x_n) + q_K(x_0 - x_{q_{K-1}})}{(q_{K-1} + q_K)(q_{K-1} + q_K + 1)} - \frac{\sum_{n=1}^{q_{K-1}} x_0 - x_n}{q_{K-1}(q_{K-1} + 1)} \\ =& \frac{d\left[\sum_{n=1}^{q_{K-1}} n + q_K q_{K-1}\right]}{(q_{K-1} + q_K)(q_{K-1} + q_K + 1)} - \frac{d \sum_{n=1}^{q_{K-1}} n}{q_{K-1}(q_{K-1} + 1)} \\ =& \frac{d}{2}\left[\frac{q_{K-1}(q_{K-1} + 1) + 2q_{K-1}q_K}{(q_{K-1} + q_K)(q_{K-1} + q_K + 1)} - 1\right]\end{aligned}$$

which is strictly negative for $d > 0$. Thus, (22) holds and $\tilde{\mathrm{IC}}_{K-1,j}$ holds with $\tilde{p}^*$ for every $j \in \{1, \ldots, K-2\}$. Now fix $k \in \{1, \ldots, K-1\}$. Since $\tilde{p}^*_{K-1} = \tilde{v}(\theta_{K-1})$, $\tilde{\mathrm{IR}}_{K-1}$ holds. By Lemma 1, $\tilde{\mathrm{IR}}_k$ holds.

If $k < K - 1$, then we have

$$\begin{aligned}\tilde{v}(\theta_k) - \tilde{p}^*_k &= \sum_{l=k}^{K-2} [\tilde{v}(\theta_{l+1}; \theta_l) - \tilde{v}(\theta_{l+1})] && \text{(by the definition of } \tilde{p}^*) \\ &= \sum_{l=k}^{K-3} \left[v(\theta_{l+1}; \theta_l) - v(\theta_{\theta_{l+1}})\right] + \tilde{v}(\theta_{K-1}; \theta_{K-2}) - \tilde{v}(\theta_{K-1}) && \text{(by (21))} \\ &< \sum_{l=k}^{K-1} [v(\theta_{l+1}; \theta_l) - v(\theta_{l+1})] && \text{(by (22))} \\ &= v(\theta_k) - p^*_k. && \text{(by the definition of } p^*)\end{aligned}$$

Because $v(\theta_k) = \tilde{v}(\theta_k)$, the inequality implies $\tilde{p}^*_k > p^*_k$. Additionally, because $v(\theta_k; \theta_{k+1}) = \tilde{v}(\theta_k; \theta_{k+1})$, if IC_{0k} holds with p^*, i.e., $v(\theta_k; \theta_{k+1}) - p^*_k \leq 0$, then $\tilde{v}(\theta_k; \theta_{k+1}) - \tilde{p}^*_k < 0$, i.e., $\tilde{\mathrm{IC}}_{0k}$

holds. Therefore, if q is implementable, then $\tilde{q}$ is implementable.

Now we show that the optimal profit from $\tilde{q}$ is higher than that from q. For each $k \in \{1, \dots, K-2\}$, because $\tilde{p}_k^* > p_k^*$, $\tilde{p}_k^* q_k > p_k^* q_k$, i.e., the park's profit from θ_k is higher in $\tilde{q}$. Since $\tilde{p}_{K-1}^* = \tilde{v}(\theta_{K-1})$ and $p_{K-1}^* < v(\theta_{K-1})$, $\tilde{p}_{k-1}^*(q_{K-1} + q_K) = v(\theta_{K-1})q_{K-1} + v(\theta_K)q_K > p_{K-1}^* q_{K-1} + p_K^* q_K$, i.e., the firm's profit from the lowest two priority passes is higher in $\tilde{q}$. Therefore, the optimal profit from $\tilde{q}$ is higher than that from q. □

A.4.2 Proof of Proposition 7

Proof. By Proposition 3, a two-pass all-serving implementable scheme exists if and only if $\beta^h \geq \frac{v(\theta_1;\theta_2) - v(\theta_2)}{v(\theta_1) - v(\theta_2;\theta_1)}$. By p^* in Lemma 6, the revenue of the all-serving two-pass scheme is

$$N^l p_2^* + N^h p_1^* = N^l v(\theta_2) + N^h \left[v(\theta_2) + \beta^h (v(\theta_1) - v(\theta_2; \theta_1)) \right].$$

The revenue of the all-serving one-pass scheme is $N^h v(\theta_1) + N^l v(\theta_2)$. Thus, conditional on $\beta^h[v(\theta_1) - v(\theta_2;\theta_1)] \geq v(\theta_1;\theta_2) - v(\theta_2)$, the two-pass scheme is better than the one-pass scheme if and only if

$$N^l v(\theta_2) + N^h \left[v(\theta_2) + \beta^h (v(\theta_1) - v(\theta_2; \theta_1)) \right] \geq N^l v(\theta_2) + N^h v(\theta_1), \tag{23}$$

from which we have

$$\begin{aligned} \beta^h &\geq \frac{v(\theta_1) - v(\theta_2)}{v(\theta_1) - v(\theta_2;\theta_1)} && \text{(by (23))} \\ &\geq \frac{v(\theta_1;\theta_2) - v(\theta_2)}{v(\theta_1) - v(\theta_2;\theta_1)} && \text{(by Claim 1)} \\ &> 1 && \text{(by the proof of Theorem 1)} \end{aligned}$$

To conclude, set $\underline{\beta} = \frac{v(\theta_1) - v(\theta_2)}{v(\theta_1) - v(\theta_2;\theta_1)}$. □

A.4.3 Proof of Proposition 8

Proof. Assume $\tilde{v}$ creates zero externality. For each $q \in \mathcal{Q}(N,K)$ and $a \in A$ such that $\overline{q}(a) = q$, setting $p_k = \tilde{v}(\theta_k; q)$ for $k \in \{1, \dots, K\}$ implements q.

Now assume $K \geq 2$ and $\tilde{v}$ creates more downgrade externality. Given $q \in \mathcal{Q}(N,K)$, to arrive at a contradiction, assume some price vector p implements q. Let $a \in A$ be a strategy profile such that $\overline{q}(a) = q$. Let $a' \in A$ be a strategy profile where exactly one customer buying θ_2 in a switches to θ_1 and $a'' \in A$ a strategy profile where exactly one customer buying θ_1 in

a switches to θ_2. Thus, IC_{12} and IC_{21} imply

$$\tilde{v}(\theta_1; \overline{q}(a')) - \tilde{v}(\theta_2; q) \leq p_1 - p_2 \leq \tilde{v}(\theta_1; q) - \tilde{v}(\theta_2; \overline{q}(a''))$$

However, the inequality above contradicts the definition of $\tilde{v}$ creating more downgrade externality. Hence IC_{12} and IC_{21} cannot both hold, a contradiction. □

A.4.4 Proof of Proposition 9

Proof. The proof focuses on schemes with every customer buying some pass. Since $u_0 = -\infty$, for implementability, it is sufficient to consider the set of IC constraints. To construct a K-pass implementable scheme, first fix $q_1 > 0$. By p^* in Lemma 2[28], IC_{21} holds if

$$v(\theta_2; \theta_1) - v(\theta_2) \leq v(\theta_1) - v(\theta_1; \theta_2).$$

The right-hand side is strictly positive. Thus, for IC_{21} to hold, it is sufficient for the left-hand side to converge to 0 as q_2 grows. To see this, note that for every $k \in \{2, \ldots, K\}$,

$$v(\theta_k; \theta_{k-1}) - v(\theta_k) = \frac{u_{Q_{k-1}}}{q_k + 1} - \frac{v(\theta_k)}{q_k + 1},$$

where Q_{k-1} is the last position of θ_{k-1} in q. The first term, $\frac{u_{Q_{k-1}}}{q_k+1}$, converges to 0 as $q_k \to \infty$. For the second term, note that

$$\lim_{q_k \to \infty} \frac{v(\theta_k)}{q_k + 1} = \lim_{q_k \to \infty} \frac{1}{q_k + 1} \frac{\sum_{n=Q_{k-1}+1}^{Q_k} u_n}{q_k} = 0.$$

The last equality holds because $\lim_{n \to \infty} \frac{u_n}{n} = 0$. Therefore, for fixed q_1, IC_{21} holds strictly with large q_2.

Suppose that with this procedure, we have picked $q_1, \ldots, q_{k-1}$ for some $k \in \{3, \ldots, K\}$ such that IC_{lj} holds strictly for every $j, l \in \{1, \ldots, k-1\}$ such that $j < l$. Fix $j \in \{1, \ldots, k-1\}$. By (1) and (2), (p^*, q) satisfying $\text{IC}_{k-1,j}$ with a strict inequality is equivalent to

$$v(\theta_j) - v(\theta_j; \theta_{j+1}) > \sum_{l=j}^{k-2} v(\theta_{l+1}; \theta_l) - v(\theta_{l+1}). \tag{24}$$

[28] If $p_K^* < 0$, redefine $p_K^* = 0$ and $p_j^* - p_{j+1}^* = v(\theta_j) - v(\theta_{j+1}; \theta_j)$ for $j \in \{1, \ldots, K-1\}$.

Similarly, if IC_{kj} holds strictly with p^*, then

$$v(\theta_j) - v(\theta_j;\theta_{j+1}) > \sum_{l=j}^{k-1} v(\theta_{l+1};\theta_l) - v(\theta_{l+1}),$$

which, by (24), holds if the difference $v(\theta_k;\theta_{k-1}) - v(\theta_k)$ is sufficiently small. By our earlier discussion, the difference approaches zero when $q_k \to \infty$. Therefore, there exists large enough q_k such that IC_{kj} holds with p^* for every $j \in \{1,\ldots,k-1\}$. Repeat this procedure for all $k \in \{2,\ldots,K\}$, and we have constructed an implementable scheme, and let M be the total number of customers in the scheme. For every $N \geq M$, we can add to the constructed scheme the additional customers (in excess of M) to the last priority pass, and the new scheme with N customers is implementable. Therefore, there exists $M > 0$ such that if $N \geq M$ then there exists $q \in \mathcal{Q}(N,K)$ that is implementable. □

Appendix B The General Multi-Type Case

In this section, we consider the general case where there could be more than two types of base utility functions. Suppose there are T types of basic utility functions and we call this case the **multi-type** case. To be precise, assume each customer's base utility function comes from $\{u^t\}_{t=1}^T$, where t is the index for utility types. For each $t = 1,\ldots,T$, let N^t be the number of customers with base utility function u^t, and let $N = \sum_{t=1}^T N^t$ be the total number of customers. Similarly to the two-type case, we assume that for every $t \in \{1,\ldots,T-1\}$, $u_n^t > u_n^{t+1}$ for every $n \in \{1,\ldots,N\}$ and $u_n^t - u_{n+1}^t > u_n^{t+1} - u_{n+1}^{t+1}$ for every $n \in \{1,\ldots,N-1\}$. In addition, assume that the reserve utility of each customer type is zero, i.e., $u_0^t = 0$ for all type t. Let $G((N^t)_{t=1}^T, K, p, (u^t)_{t=1}^T)$ be the strategic-form game defined analogously to that in the single-type case. Given (N,K), let $\overline{\mathcal{Q}}(N,K) = \{q \in (\{0\} \cup \mathbb{N})^{K+1} : \sum_{k=0}^K q_k = N\}$. Define the set of schemes as

$$\mathcal{Q}\left(\left(N^t\right)_{t=1}^T, K\right) = \left\{q = (q^1,\ldots,q^t) \in \times_{t=1}^T \overline{\mathcal{Q}}(N^t,K) : \sum_{\tau=1}^T q_k^\tau > 0 \text{ if } 1 \leq k \leq K\right\}.$$

The restriction that $\sum_{t=1}^T q_k^t > 0$ for every $k \in \{1,\ldots,K\}$ ensures that every priority pass has at least one customer, which is analogous to the definition in the single-type case. For each customer type t, construct the pass utility function v^t from u^t. Given $j \in \{0,\ldots,K\}$ and $k \in \{1,\ldots,K\}$, IC_{jk}^t and IR_k^t are defined analogously to the single-type case with the pass-utility function changed to v^t. Analogous to the two-type case, IC_{jk}^t and IR_k^t may not be defined for every $j,k \in \{1,\ldots,K\}$. To be precise, for a scheme q to be implementable,

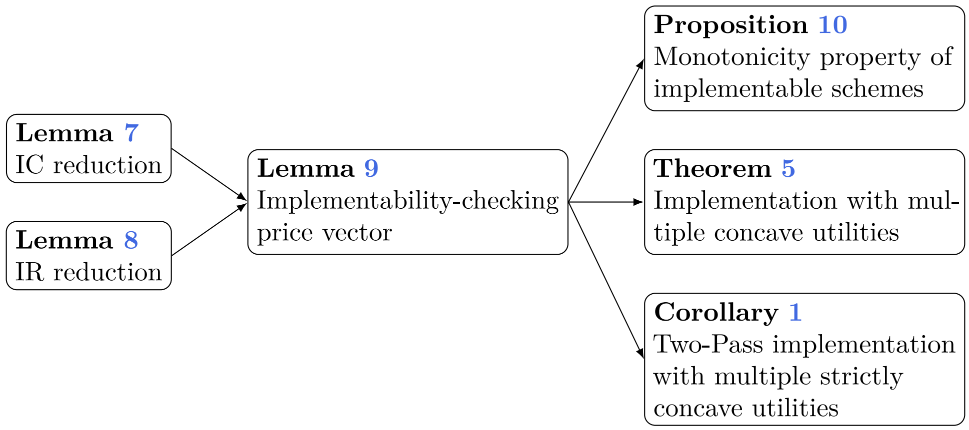

Figure 5: Road map of Appendix B.1

IC_{jk}^t is defined only if there is some customer of type t that buys θ_j, i.e., $q_j^t > 0$. Similarly, IR_k^t is defined only if $q_k^t > 0$.

Fix a scheme q. Let $\bar{t}_j = \max\{1 \le \tau \le T : q_j^\tau > 0\}$, which is the lowest customer type that buys θ_j, and $\underline{t}_j = \min\{1 \le \tau \le T : q_j^\tau > 0\}$, which is the highest customer type that buys θ_j. As in the two-type case, $\mathrm{IC}_{jk}^{\bar{t}_j}$ implies every constraint in $\{\mathrm{IC}_{jk}^\tau : q_j^\tau > 0, 1 \le \tau \le T\}$ if $0 < j < k$ and $\mathrm{IC}_{jk}^{\underline{t}_j}$ implies every constraint in $\{\mathrm{IC}_{jk}^\tau : q_j^\tau > 0, 1 \le \tau \le T\}$ if $j = 0$ or $1 \le k < j \le K$. Write $\mathrm{IC}_{jk} = \mathrm{IC}_{jk}^{\bar{t}_j}$ if $0 < j < k$ and $\mathrm{IC}_{jk} = \mathrm{IC}_{jk}^{\underline{t}_j}$ if $j = 0$ or $1 \le k < j \le K$. Similarly, $\mathrm{IR}_k^{\bar{t}_k}$ implies every constraint in $\{\mathrm{IR}_k^\tau : q_k^\tau > 0, 1 \le \tau \le T\}$, and we write $\mathrm{IR}_{jk} = \mathrm{IR}_k^{\bar{t}_k}$. Let the set of IC constraints be the collection of IC_{jk} over $j \in \{0, \ldots, K\}$ such that $\sum_{t=1}^T q_j^t > 0$ and $k \in \{1, \ldots, K\}$; let the set of IR collection of IR_k over $k \in \{1, \ldots, K\}$. Like the one-type case, implementation can be similarly defined with respect to the set of IC and IR constraints.

B.1 Lemmas for Multi-Type Case

This subsection generalizes the lemmas in the two-type case to the general multi-type case. Most results in the multi-type case use Lemma 9 that we state later in this subsection, which is a generalization of Lemma 6 in the two-type case. This lemma allows us to check the implementability of a scheme by the price vector binding IR_K and every local downward IC constraint. A road map of how lemmas in this section contribute to the results in the multi-type case is illustrated in Figure 5.

Lemma 7 (Generalization of Lemma 4)**.** *Fix $\left((N^t)_{t=1}^T, K, (u^t)_{t=1}^T\right)$ with $K \ge 2$, a scheme $q \in \mathcal{Q}((N^t)_{t=1}^T, K)$, and a price vector p. Fix $j, k \in \{1, \ldots, K\}$ such that $j < k$ and assume $IC_{l,l+1}$ holds for every $l \in \{j, \ldots, k-1\}$. If $\bar{t}_l \le \bar{t}_{l+1}$ for every $l \in \{j, \ldots, k-1\}$, then IC_{jk} holds.*

Proof. Since $\bar{t}_l \leq \bar{t}_{l+1}$, we have

$$\begin{aligned} p_l - p_{l+1} &\leq v^{\bar{t}_l}(\theta_l) - v^{\bar{t}_l}(\theta_{l+1};\theta_l) && (\text{by } \mathrm{IC}^{\bar{t}_l}_{l,l+1}) \\ &\leq v^{\bar{t}_j}(\theta_l) - v^{\bar{t}_j}(\theta_{l+1};\theta_l). && (\text{by the definition of types}) \end{aligned}$$

Therefore, combining $\mathrm{IC}^{\bar{t}_l}_{l,l+1}$ for $l \in \{j, \ldots, k-1\}$ implies

$$\begin{aligned} p_j - p_k = \sum_{l=j}^{k-1} p_l - p_{l+1} &\leq \sum_{l=l}^{k-1} v^{\bar{t}_j}(\theta_l) - v^{\bar{t}_j}(\theta_{l+1};\theta_l) && (\text{by } \mathrm{IC}^{\bar{t}_l}_{l,l+1}) \\ &= v^{\bar{t}_j}(\theta_j) - v^{\bar{t}_j}(\theta_k) - \sum_{l=j}^{k-1}\left[v^{\bar{t}_j}(\theta_{l+1};\theta_l) - v^{\bar{t}_j}(\theta_{l+1})\right] \\ &\leq v^{\bar{t}_j}(\theta_j) - v^{\bar{t}_j}(\theta_k) - \left[v^{\bar{t}_j}(\theta_k;\theta_{k-1}) - v^{\bar{t}_j}(\theta_k)\right] && (\text{by Claim 1}) \\ &= v^{\bar{t}_j}(\theta_j) - v^{\bar{t}_j}(\theta_k; j). && (\text{by Claim 1}) \end{aligned}$$

Thus, IC_{jk} holds. ☐

Lemma 8 (Generalization of Lemma 5)**.** *Given $\left((N^t)_{t=1}^T, K, (u^t)_{t=1}^T\right)$ with $K \geq 2$, fix a scheme $q \in \mathcal{Q}((N^t)_{t=1}^T, K)$ and a price vector p. Assume there exists some $j,k \in \{1,\ldots,K\}$ such that $j < k$ and $\bar{t}_j \leq \bar{t}_k$. If IC_{jk} and IR_k hold, then IR_j holds.*

Proof. Pick $t \in \{1,\ldots,T\}$ such that $q_j^t > 0$. We have

$$\begin{aligned} v^t(\theta_j) - p_j &\geq v^{\bar{t}_j}(\theta_j) - p_j && (\text{by } t \leq \bar{t}_j) \\ &\geq v^{\bar{t}_j}(\theta_k;\theta_j) - p_k && (\text{by } \mathrm{IC}^{\bar{t}_j}_{jk}) \\ &\geq v^{\bar{t}_k}(\theta_k) - p_k && (\text{by } \bar{t}_j \leq \bar{t}_k) \\ &\geq 0. && (\text{by } \mathrm{IR}^{\bar{t}_k}_k) \end{aligned}$$

Thus, IR_k holds. ☐

The following result is an extension of Lemma 6 from the two-type case.

Lemma 9 (Implementability conditions in multi-type case)**.** *Fix $\left((N^t)_{t=1}^T, K, (u^t)_{t=1}^T\right)$. Pick $q \in \mathcal{Q}((N^t)_{t=1}^T, K)$ where (i) $\bar{t}_j \leq \bar{t}_k$ for every $j,k \in \{1,\ldots,K\}$ such that $j < k$, (ii) every customer buys some priority pass, (iii) $v^T(\theta_K) \geq 0$. Let $p^* = (p_1^*,\ldots,p_K^*)$ such that $p_K^* = v^T(\theta_K)$ and $p_j^* - p_{j+1}^* = v^{\bar{t}_j}(\theta_j) - v^{\bar{t}_j}(\theta_{j+1};\theta_j)$ for every $j \in \{1,\ldots,K-1\}$. The scheme q is implementable if and only if (p^*, q) satisfies every upward IC constraint.*

Proof. Since every customer buys some priority pass and $\bar{t}_j \leq \bar{t}_k$ for every $j,k \in \{1,\ldots,K\}$ such that $j < k$, $q_K^T > 0$ and IR_K^T holds. By Lemma 7 and Lemma 8, (p^*, q) satisfies every downward IC and IR constraint. Therefore, q is implementable if (p^*, q) satisfies every upward IC constraint.

Now assume p implements q. Pick $j, k \in \{1, \ldots, K\}$ such that $j < k$. Then (p, q) satisfying IC_{kj} implies $p_j - p_k \geq v^{\underline{t}_k}(\theta_j; \theta_k) - v^{\underline{t}_k}(\theta_k)$. We have

$$\begin{aligned}
p_j - p_k &= \sum_{l=j}^{k-1} p_l - p_{l+1} \\
&\leq \sum_{l=j}^{k-1} v^{\bar{t}_l}(\theta_l) - v^{\bar{t}_l}(\theta_{l+1}; \theta_l) && \text{(by } (p, q) \text{ satisfying } \text{IC}^{\bar{t}_l}_{l,l+1}) \\
&= \sum_{l=j}^{k-1} p^*_l - p^*_{l+1} = p^*_j - p^*_k. && \text{(by the definition of } p^*)
\end{aligned}$$

Hence (p^*, q) satisfies IC_{kj}. The proof is complete. □

B.2 Implementability in the Multi-Type Case

For tractability, a recurring setup we have in this subsection is to fix $((N^t)_{t=1}^T, K, (u^t)_{t=1}^T)$ in the multi-type case and a base utility function u. For every t such that $1 \leq t \leq T$, let $u^t = \beta^t u$, where $\beta^1 > \beta^2 > \ldots > \beta^T = 1$.[29] We call this setup the **multiplicative** multi-type case. A multi-type case is called **concave** if each customer type's base utility function is concave and **linear** if each base utility function is linear.

When there are multiple customer types, implementing multi-pass schemes is possible if customers have utility functions that are sufficiently different from each other. The following result characterizes the implementability conditions with respect to customer types.

Theorem 5 (Implementation with multiple concave utilities)**.** *Consider the concave multiplicative multi-type case. Fix a scheme* $q \in \mathcal{Q}((N^t)_{t=1}^T, K)$ *such that* $\sum_{t=1}^T q_0^t = 0$ *and* $\bar{t}_k \leq \bar{t}_{k+1}$ *for* $k \in \{1, \ldots, K-1\}$*. For* $j, k \in \{1, \ldots, K\}$ *such that* $j < k$*, there exists* $b_{kj} \leq \beta^{\bar{t}_j}$ *such that* q *is implementable if and only if* $\beta^{\underline{t}_k} \leq b_{kj}$ *for each* $j, k \in \{1, \ldots, K\}$ *such that* $j < k$*.*

Proof. Assume q is implementable. Since $\bar{t}_k \leq \bar{t}_{k+1}$ for $k \in \{1, \ldots, K-1\}$, the conditions of Lemma 9 hold, and we can check the implementability of q by the price vector p^* that binds all local downward IC constraints and IR_K. Therefore, for every $j, k \in \{1, \ldots, K\}$ such that

[29] Here the superscript for each β is an index, not an exponent.

$j < k$,

$$p_j^* - p_k^* = \sum_{l=j}^{k-1} \left[v^{\bar{t}_l}(\theta_l) - v^{\bar{t}_l}(\theta_{l+1};\theta_l) \right] = \sum_{l=j}^{k-1} \beta^{\bar{t}_l} \left[v(\theta_l) - v(\theta_{l+1};\theta_l) \right].$$

Thus, (p^*, q) satisfies IC_{kj} if and only if

$$p_j^* - p_k^* = \sum_{l=j}^{k-1} \beta^{\bar{t}_l} \left[v(\theta_l) - v(\theta_{l+1};\theta_l) \right] \geq \beta^{\underline{t}_k} \left[v(\theta_j;\theta_{j+1}) - v(\theta_k) \right].$$

We can solve for $\beta^{\underline{t}_k}$ to get

$$\beta^{\underline{t}_k} \leq \sum_{l=j}^{k-1} \frac{v(\theta_l) - v(\theta_{l+1};\theta_l)}{v(\theta_j;\theta_{j+1}) - v(\theta_k)} \beta^{\bar{t}_l}. \tag{25}$$

Let b_{kj} be the right-hand side of the inequality above. Note that

$$\begin{aligned} \sum_{l=j}^{k-1} v(\theta_l) - v(\theta_{l+1};\theta_l) &= v(\theta_j) - v(\theta_k) - \sum_{l=j}^{k-1} \left[v(\theta_{l+1};\theta_l) - v(\theta_{l+1}) \right] \\ &\leq v(\theta_j) - v(\theta_k) - \left[v(\theta_k;\theta_{k-1}) - v(\theta_k) \right] && \text{(by Claim 1)} \\ &\leq v(\theta_j;\theta_{j+1}) - v(\theta_k), && \text{(by the proof of Theorem 1)} \end{aligned}$$

which implies that b_{kj} is in the affine simplex of $\beta^{\bar{t}_l}$ for $l \in \{j, \ldots, k-1\}$, hence $b_{kj} \leq \max_{j \leq l \leq k-1} \beta^{\bar{t}_l} = \beta^{\bar{t}_j}$. Therefore, if q is implementable, then $\beta^{\underline{t}_k} \leq b_{kj}$ for every $j, k \in \{1, \ldots, K\}$ such that $j < k$.

Conversely, let b_{kj} for every $j, k \in \{1, \ldots, K-1\}$ such that $j < k$ be defined to be the right-hand side of (25) and assume $\beta^{\underline{t}_k} \leq b_{kj}$. Again, because $\bar{t}_k \leq \bar{t}_{k+1}$ for $k \in \{1, \ldots, K-1\}$, the conditions of Lemma 9 hold. Since $\beta^{\underline{t}_k} \leq b_{kj}$ defined in (25) for $j, k \in \{1, \ldots, K\}$ such that $j < k$ is exactly the condition for (p^*, q) in Lemma 9 to satisfy IC_{kj}, q is implementable if $\beta^{\underline{t}_k} \leq b_{kj}$ for every $j, k \in \{1, \ldots, K\}$ such that $j < k$. □

The intuition of the theorem is similar to that of Theorem 4 in the two-type case: Customer types in different priority passes need to be sufficiently different for the scheme to be implementable. Towards a straightforward intuition, consider a special case of Theorem 5 where u is linear, $K = 3$, and the scheme in consideration has m customers in each pass. In this case, the price vector binding every downward IC constraint satisfies IC_{31} if and only if

$$\underbrace{\frac{\beta^{\bar{t}_1}}{2}(2m-1)}_{\substack{\text{Upper bound of} \\ p_1 - p_2}} + \underbrace{\frac{\beta^{\bar{t}_2}}{2}(2m-1)}_{\substack{\text{Upper bound of} \\ p_2 - p_3}} \geq \underbrace{\frac{\beta^{\underline{t}_3}}{2}(4m-1)}_{\substack{\text{Lower bound of} \\ p_1 - p_3}} \tag{26}$$

Hence, (26) does not hold if both $\beta^{\bar{t}_1}$ and $\beta^{\bar{t}_2}$ are too close to $\beta^{\underline{t}_3}$; if both slopes on the left-hand side are sufficiently larger than $\beta^{\underline{t}3}$, then IC_{31} holds. The implementability condition is for the customer types to be sufficiently different. Intuitively, a larger difference in different types raises the price difference between two priority passes, giving customers in the lower priority less incentive to upgrade.

We emphasize that the implementability condition in Theorem 5 requires that customer types in different priority passes, including customers whose priorities are close, need to be sufficiently different. For example, in the strictly concave multiplicative multi-type case, an implementable scheme implies the existence of a large enough "gap" between two adjacent customer types in the queue, as illustrated below by an immediate implication of Theorem 5.

Corollary 1 (Two-pass implementation with multiple strictly concave utilities)**.** *Consider the strictly concave multiplicative multi-type case where $K = 2$. Let $q \in \mathcal{Q}\left((N^t)_{t=1}^T, K\right)$ be such that $v^T(\theta_2) \geq 0$ and $q_0^t = 0$ for every customer type t. The scheme q is implementable if and only if $\underline{t}_2 = \bar{t}_1 + 1$ and*

$$\frac{\beta^{\bar{t}_1}}{\beta^{\underline{t}_2}} \geq \frac{v(\theta_1;\theta_2) - v(\theta_2)}{v(\theta_1) - v(\theta_2;\theta_1)}, \tag{27}$$

where v is the pass utility function constructed from u. In addition, the right-hand side of (27) is strictly greater than 1 and converges to 1 as $N^t \to \infty$ for every $t \in \{\underline{t}_2, \ldots, T\}$.

Proof. Since every customer buys some pass and each u^t is strictly concave, by Theorem 1 and Lemma 10, $\bar{t}_1 = \underline{t}_2 - 1$. Since $K = 2$ and every customer buys some priority pass, ID_{12} is necessary and sufficient for implementability, which implies

$$\frac{\beta^{\bar{t}_1}}{\beta^{\underline{t}_2}} \geq \frac{v(\theta_1;\theta_2) - v(\theta_2)}{v(\theta_1) - v(\theta_2;\theta_1)}.$$

The right-hand side is strictly larger than 1 by the proof of Theorem 1 for strictly concave base utility functions. If $N^t \to \infty$ for every $t \in \{\underline{t}_2, \ldots, T\}$, then q_2 tends to infinity. Because u is concave, both $v(\theta_2)$ and $v(\theta_2;\theta_1)$ converge to $-\infty$. lastly, as $v(\theta_2;\theta_1) = \frac{1}{q_2+1}u_{q_1} + \frac{q_2}{q_2+1}v(\theta_2)$, the right-hand side converges to 1 as q_2 tends to infinity. □

The result implies that a multi-pass scheme may not be implementable even if the range of customer types (i.e., $\beta^1 - \beta^T$) is very large but adjacent customer types are very close to each other.

Note that the conditions and the result of Theorem 5, that $\bar{t}_k \leq \bar{t}_{k+1}$ for $k \in \{1, \dots, K-1\}$, and that $\beta^{\underline{t}_k} \leq b_{kj} \leq \beta^{\bar{t}_j}$ for $j, k \in \{1, \dots, K\}$ such that $j < k$, together imply $\bar{t}_k \leq \underline{t}_{k+1}$ for $k \in \{1, \dots, K-1\}$. It turns out that this implication is necessary for implementability in the concave multi-type case, which we include in the following monotonicity result.[30]

Proposition 10 (Monotonicity with multiple concave utilities)**.** *Fix the concave multi-type case where $K \geq 2$. Assume $q \in ((N^t)_{t=1}^T, K)$ is implementable.*

(a) For every $j \in \{1, \dots, K-1\}$, $\bar{t}_j \leq \underline{t}_{j+1}$. Moreover, if additionally $j \leq K-2$, then in the inequalities $\bar{t}_j \leq \underline{t}_{j+1} \leq \bar{t}_{j+1} \leq \underline{t}_{j+2}$, at least one of the inequalities is strict.

(b) If $1 \leq \tau \leq \bar{t}_{K-1}$ and $\tau < \bar{t}_K$, then $q_0^\tau = 0$.

Proof. The two parts of the results are shown through the following two lemmas. The first lemma shows that a higher-priority customer never buys a strictly lower-priority pass than a lower-type customer does in an implementable scheme; the second lemma shows that if the customer's type is (weakly) higher than the lowest customer type that buys θ_{K-1} and strictly higher than the lowest customer type that buys θ_K, then this customer must necessarily buy some pass.

Lemma 10 (Higher-Priority for higher type)**.** *Consider the concave multi-type case and assume that $q \in \big((N^t)_{t=1}^T, K\big)$ is implementable. For every $j \in \{1, \dots, K-1\}$, $\bar{t}_j \leq \underline{t}_{j+1}$.*

Proof of lemma. Towards a contradiction, assume that there exists $q \in \mathcal{Q}((N^t)_{t=1}^T, K)$ where $\bar{t}_j > \underline{t}_k$ for some $j, k \in \{1, \dots, K\}$ such that $j < k$. With this assumption, we have

$$\begin{aligned} v^{\bar{t}_j}(\theta_j) - v^{\bar{t}_j}(\theta_j; \theta_k) &< v^{\underline{t}_k}(\theta_j) - v^{\underline{t}_k}(\theta_j; \theta_k) && \text{(by } \bar{t}_j > \bar{t}_k\text{)} \\ &\leq v^{\underline{t}_k}(\theta_k; \theta_j) - v^{\underline{t}_k}(\theta_k), && \text{(by the proof of Theorem 1)} \end{aligned}$$

However, the inequality from the two ends violates ID_{jk} and hence q is not implementable. □

Lemma 11 (Pass-Buying with concave utilities)**.** *Consider the concave multi-type case where $K \geq 2$ and assume $q \in \mathcal{Q}((N^t)_{t=1}^T, K)$ is implementable. For every customer type $t \in \{1, \dots, \bar{t}_{K-1}\}$ such that $t < \bar{t}_K$, $q_0^t = 0$.*

[30] Recall that the monotonicity property does not hold in general, as Example 3 and Example 2 in the two-type case show.

Proof of lemma. By Lemma 10, $\bar{t}_k \le \bar{t}_K$. Let p be a price vector implementing q. $\text{IC}_{K-1,K}$ implies

$$v^{\bar{t}_{K-1}}(\theta_{K-1}) - v^{\bar{t}_{K-1}}(\theta_K;\theta_{K-1}) \ge p_{K-1} - p_K.$$

Because each u^t is concave, by the proof of Theorem 1,

$$v^{\bar{t}_{K-1}}(\theta_K;\theta_{K-1}) - v^{\bar{t}_{K-1}}(\theta_K) \ge v^{\bar{t}_{K-1}}(\theta_{K-1}) - v^{\bar{t}_{K-1}}(\theta_{K-1};\theta_K).$$

Adding up the two inequalities we have in this lemma so far, we obtain

$$v^{\bar{t}_{K-1}}(\theta_{K-1};\theta_K) - p_{K-1} \ge v^{\bar{t}_{K-1}}(\theta_K) - p_K \tag{28}$$

Moreover, by Lemma 10, $\bar{t}_{k-1} \le \bar{t}_k$. We therefore have

$$\begin{aligned} v^{\bar{t}_{K-1}}(\theta_{K-1};\theta_K) - p_{K-1} &\ge v^{\bar{t}_{K-1}}(\theta_K) - p_K && \text{(by (28))} \\ &\ge v^{\bar{t}_K}(\theta_K) - p_K && (\text{by } \bar{t}_{k-1} \le \bar{t}_l) \\ &\ge 0, && (\text{by IR}_k) \end{aligned}$$

Lastly, assume there is some $t \in \{1, \dots, \bar{t}_{K-1}\}$ such that $t < \bar{t}_K$ and $q_0^t > 0$. We have

$$v^t(\theta_{K-1};\theta_K) - p_{K-1} \ge v^{\bar{t}_{K-1}}(\theta_{K-1};\theta_K) - p_{K-1} \ge v^{\bar{t}_K}(\theta_K) - p_K \ge 0.$$

At least one of the first two inequalities must be strict because $t < \bar{t}_K$, and hence IC_{0j}^t does not hold. Therefore, for q to be implementable, $q_0^t = 0$. □

Therefore, by Lemma 10, if q is implementable, then $\bar{t}_k \le \underline{t}_{k+1} \le \bar{t}_{k+1} \le \underline{t}_{k+2}$ for every $k \in \{1, \dots, K-2\}$. To see that at least one of the inequalities must be strict, note that by Theorem 1 and 2, a customer type can be found in at most two passes. Lastly, part (b) is the same as Lemma 11. □

Part (a) of Proposition 10 states that, in an implementable scheme, a higher-type customer cannot have a lower priority than does a lower-type customer. For intuition, consider a scheme in which a lower customer type buys a higher priority pass than a higher customer type. Because the lower-type's base utility function decreases more slowly with respect to positions in the queue, the price difference between the higher and the lower priority pass is not large enough for the higher-type customers' upward IC constraint from the lower priority pass, making the scheme unimplementable. An interpretation of part (b) is that if a customer's type is (weakly) higher than the lowest-type in the second-to-last priority pass, then this customer must purchase some priority pass.[31] When the park incentivizes a high-priority customer not to downgrade, the park needs to give some surplus to the customer due to the downgrade externality, and this extra surplus would incentivize every (weakly) higher-type customer not in the queue to join the queue.

Proposition 10 also sheds some indirect intuition on the constrained number of priority passes. In Theorem 5 and its Corollary 1, we have shown that customer types in different priorities, including those priorities that are close to each other, need to be sufficiently different in an implementable scheme. However, when the adjacent customer types are all close to each other, one may wonder whether the park can create "gaps" between customer types by excluding some types from the queue. For example, suppose there are five customer types in the strictly concave multi-type case, with each customer type very close to the nearest customer types. If the park would like to implement a three-pass scheme, the park may consider excluding the second and the fourth customer types from the queue so that there is enough difference between the customer types remaining in the queue. Proposition 10, however, implies that this particular exclusion is not possible in an implementable scheme, and there are restrictions to such customer exclusions. We characterize some of these restrictions in the result below, which is an immediate implication of Proposition 10.

Corollary 2 (Limit to customer exclusion)**.** *Consider the concave multi-type case. Let* $q \in \mathcal{Q}((N^t)_{t=1}^T, K)$ *be an implementable scheme.*

(a) For every $j \in \{1, \dots, K-2\}$, $\bar{t}_j = \underline{t}_{j+1}$ *or* $\bar{t}_j + 1 = \underline{t}_{j+1}$.

(b) Fix $j \in \{1, \dots, K-1\}$. *For every* $\tau \in \{\underline{t}_j + 1, \dots, \bar{t}_j - 1\}$, $q_j^\tau = N^\tau$.

Part (a) means that, between two adjacent priority passes, if there are any customer types that are completely excluded from the queue, this exclusion can only happen between the lowest customer type in the second-last priority pass and the highest customer type in the last priority pass; otherwise, customers of an excluded type would have an incentive to join the queue. Part (b) implies that, within a pass except for the lowest-priority pass, customer types in a pass must be "connected": Given an implementable scheme, if a customer's type is strictly between the highest and the lowest customer type in a pass whose priority is not the lowest, then this customer must be in that priority pass. Therefore, if the park wishes to create any "gaps", it could only do so between the two lowest priority passes.

The reader may again notice that, with the customer types fixed, (26) also holds if m is sufficiently large. In Propositions 4 and 5 for the linear multiplicative two-type case, we find that more customers help with implementability, and a scheme is not implementable if the two

[31] Note that in Example 3, the condition is not met, and the monotonicity property does not hold.

customer types are too close. The two results have extensions to the multi-type case. Before we introduce them, we make a definition that will be useful in the extended results. Given a scheme q in the multiplicative multi-type case where $K > 2$, define $\underline{R}(q) = \min_{1\leq k\leq K-2} \frac{\beta^{\bar{t}_j}}{\beta^{\underline{t}_{j+2}}}$, which gives the minimum relative difference of customer types that are two priorities apart.

Given the necessary conditions derived in Proposition 10, we often make the following recurring assumptions about schemes.

Definition 4 (Regular scheme)**.** In the multi-type case, a scheme $q \in \mathcal{Q}((N^t)_{t=1}^T, K)$ is called regular if the following two conditions hold:

(i) Every customer buys some priority pass, and IR_K holds for some price vector.

(ii) For every $j \in \{1, \ldots \leq K-1\}$, $\bar{t}_j \leq \underline{t}_{j+1}$. If additionally $j \leq K-2$, then in the inequalities $\bar{t}_j \leq \underline{t}_{j+1} \leq \bar{t}_{j+1} \leq \underline{t}_{j+2}$, at least one inequality is strict.

Condition (i) is assumed so that we could focus on the switching incentives between different priority passes. By Proposition 10, condition (ii) is necessary for implementability in the concave multi-type case. Note that, when $K > 2$, $\underline{R}(q) > 1$ for every regular scheme.

The following result formalizes the conjecture that sufficiently many customers lead to implementability.

Proposition 11 (Sufficiently many customers for implementation)**.** *Consider the linear multiplicative multi-type case where $K > 2$. Fix a regular scheme q.*

(a) For each $R > 1$, there exists $\underline{M} \geq 1$ such that if a regular scheme q has at least $\underline{M}$ customers in each priority pass and $\underline{R}(q) \geq R$, then q is implementable.

(b) If $R \geq 2$, then part (a) is true for $\underline{M} = 1$.

Proof. Since the scheme in consideration is fixed, denote $\underline{R}(q)$ by $\underline{R}$ instead. Fix $R > 1$ and assume $\underline{R} \geq R$. Since $K > 2$, by part (a) of Proposition 10, $\underline{R} > 1$. Let $\underline{m} = \min_{1\leq k\leq K} \sum_{t=1}^T q_k^t$. Fix $j \in \{1, \ldots, K-2\}$. From the proof of Theorem 5, $\text{IC}_{j+2,j}$ holds if and only if

$$\frac{\beta^{\bar{t}_j}}{2}(q_j + q_{j+1} - 1) + \frac{\beta^{\bar{t}_{j+1}}}{2}(q_{j+1} + q_{j+2} - 1) - \frac{\beta^{\underline{t}_{j+2}}}{2}(q_j + 2q_{j+1} + q_{j+2} - 1) \geq 0. \tag{29}$$

Since the left-hand side is increasing in $\beta^{\bar{t}_j}$, $\beta^{\bar{t}_{j+1}}$, q_j, q_{j+1}, and q_{j+2}, the inequality is implied by the following inequality:

$$\frac{R\beta^{\underline{t}_{j+2}}}{2}(2\underline{m} - 1) + \frac{\beta^{\underline{t}_{j+2}}}{2}(2\underline{m} - 1) - \frac{\beta^{\underline{t}_{j+2}}}{2}(4\underline{m} - 1) \geq 0, \tag{30}$$

because $\sum_{t=1}^{T} q_k^t \geq \underline{m}$ for every $k \in \{1, \ldots, K\}$ and $\beta^{\bar{t}_j}/\beta^{\underline{t}_{j+2}} \geq \underline{R} \geq R$. Inequality (30) holds if $R \geq \frac{2\underline{m}}{2\underline{m}-1}$. Now additionally assume $j > 1$ and consider $\text{IC}_{j+2,j-1}$. From $\text{IC}_{j+2,j}$ to $\text{IC}_{j+2,j-1}$ with respect to (29), the left-hand side of the inequality increases by at least $\frac{\underline{R}\beta^{\underline{t}_{j+2}}}{2}(2\underline{m}-1) - \beta^{\underline{t}_{j+2}}\underline{m}$ because $\beta^{\bar{t}_{j-1}} \geq \beta^{\bar{t}_j}$. Therefore, conditional on $\text{IC}_{j+2,j}$, $\text{IC}_{j+2,j-1}$ holds if $\underline{R} \geq \frac{2\underline{m}}{2\underline{m}-1}$. Provided that $j > 2$, continue the same reasoning, and we see that if $\underline{R} \geq \frac{2\underline{m}}{2\underline{m}-1}$, then $\text{IC}_{j+2,l}$ holds for every $l \in \{1, \ldots, j+1\}$. Since $\frac{2\underline{m}}{2\underline{m}-1}$ is decreasing in $\underline{m} \geq 1$ and tends to 1, for each fixed $R > 1$, we can find $\underline{M}$ such that $R \geq \frac{2\underline{M}}{2\underline{M}-1}$. Therefore, if $\underline{R}(q) \geq R$ and $\underline{m} \geq \underline{M}$, q is implementable.

Lastly, note that $\frac{2\underline{m}}{2\underline{m}-1} \leq 2$ for $m \geq 1$. Therefore, if $R \geq 2$, then every scheme q such that $\underline{R}(q) \geq R$ is implementable. □

The result's condition on the relative difference between types ensures enough price difference between priority passes to eliminate upgrade incentives. When $\underline{R}$ is bounded below strictly above 1, provided that IR_K still holds, having sufficiently many customers raises the price difference between a high and low-priority pass, making the IC constraints between the two passes hold.

To see how a sufficiently large $\underline{R}(q)$ makes a scheme implementable, consider the linear multiplicative multi-type case where $K = 4$. Consider a regular scheme where each priority pass has exactly m customers. We can get IC_{42} by changing the pass indices in (26):

$$\underbrace{\frac{\beta^{\bar{t}_2}}{2}(2m-1)}_{\substack{\text{Upper bound} \\ \text{of } p_2 - p_3}} + \underbrace{\frac{\beta^{\bar{t}_3}}{2}(2m-1)}_{\substack{\text{Upper bound} \\ \text{of } p_3 - p_4}} \geq \underbrace{\frac{\beta^{\underline{t}_4}}{2}(4m-1)}_{\substack{\text{Lower bound} \\ \text{of } p_2 - p_4}},$$

which holds if $\frac{\beta^{\bar{t}_2}}{\beta^{\underline{t}_4}} \geq \frac{2m}{2m-1}$ by the proof of Proposition 11. For linear utility functions, binding the downward local IC constraints satisfies IC_{41} if and only if

$$\underbrace{\frac{\beta^{\bar{t}_1}}{2}(2m-1)}_{\substack{\text{Upper bound} \\ \text{of } p_1 - p_2}} + \frac{\beta^{\bar{t}_2}}{2}(2m-1) + \frac{\beta^{\bar{t}_3}}{2}(2m-1) \geq \underbrace{\beta^{\underline{t}_4} m + \frac{\beta^{\underline{t}_4}}{2}(4m-1)}_{\text{Lower bound of } p_1 - p_4}.$$

Comparing the two inequalities shows that, from IC_{42} to IC_{41}, the left-hand side increases by $\frac{\beta^{\bar{t}_1}}{2}(2m-1)$ and the right-hand side increases by $\beta^{\underline{t}_4} m$. Thus, conditional on IC_{42}, IC_{41} holds if $\frac{\beta^{\bar{t}_1}}{\beta^{\underline{t}_4}} \geq \frac{2m}{2m-1}$, which the proof shows holds when $\underline{R}(q) \geq 2$ even when there is only one customer in each priority pass, i.e., $m = 1$. For a customer buying θ_4, the inequality $\beta^{\bar{t}_1}(2m-1) \geq 2\beta^{\underline{t}_4} m$ means that the price difference $p_1 - p_4$ is more than the customer's utility increase between switching to θ_1, eliminating the customer's incentive to upgrade to θ_1 conditional on IC_{42}.

Note that $\underline{R}(q) \geq 2$ implies an exponential growth in the difference between customer types. For example, when $\underline{R}(q) = 2$, a scheme with six priority passes implies that the slope of the lowest customer type in the highest-priority pass is at least eight times larger than that of the highest customer type in the lowest-priority pass.

Proposition 11 shows that having sufficiently many customers in each priority pass leads to implementability, but it does not provide clarification on implementability when the number of customers in each pass is small. The following result shows that a scheme is not implementable if some customer types are close and there are not many customers in each pass.

Proposition 12 (Not implementable when customer types are too close)**.** *Consider the linear multiplicative multi-type case where $K > 2$. For each $\overline{m} \geq 1$, there exists $\delta \in (0, 1)$ such that a regular scheme q with at most $\overline{m}$ customers in each priority pass is not implementable if $\underline{R}(q) < 1 + \delta$.*

Proof. Since the scheme in consideration is fixed, denote $\underline{R}(q)$ by $\underline{R}$ instead. Let $\overline{m} = \max_{1 \leq k \leq K} \sum_{t=1}^{T} q_k^t$. Pick $j \in \{1, \ldots, K-2\}$ such that $\frac{\beta^{\bar{t}_j}}{\beta^{\underline{t}_{j+2}}} = \underline{R}$. From the proof of Theorem 5, $\mathrm{IC}_{j+2,j}$ does not hold if and only if

$$\frac{\beta^{\bar{t}_j}}{2}(q_j + q_{j+1} - 1) + \frac{\beta^{\bar{t}_{j+1}}}{2}(q_{j+1} + q_{j+2} - 1) - \frac{\beta^{\underline{t}_{j+2}}}{2}(q_j + 2q_{j+1} + q_{j+2} - 1) < 0.$$

The left-hand side is increasing in $\beta^{\bar{t}_j}$, $\beta^{\bar{t}_{j+1}}$, q_j, q_{j+1},and q_{j+2}, and hence the above inequality is implied by:

$$\frac{\underline{R}\beta^{\bar{t}_{j+2}}}{2}(2\overline{m} - 1) + \frac{\underline{R}\beta^{\bar{t}_{j+2}}}{2}(2\overline{m} - 1) - \frac{\beta^{\bar{t}_{j+2}}}{2}(4\overline{m} - 1) < 0,$$

because $\sum_{t=1}^{T} q_k^T \leq \overline{m}$ for every $k \in \{1, \ldots, K\}$ and $\beta^{\bar{t}_{j+1}} \leq \beta^{\bar{t}_j} = \underline{R}\beta^{\underline{t}_{j+2}}$. The inequality implies that q is not implementable if $\underline{R} < \frac{4\overline{m}-1}{4\overline{m}-2}$. The proof is complete if we let $\delta = \frac{4\overline{m}-1}{4\overline{m}-2} - 1 > 0$. □

This result further shows that as the difference between customer types gets closer to zero, the lowest possible $\underline{M}$ that one can take in the statement of Proposition 11, gets higher and is unbounded as the difference approaches zero. When the customer types are not significantly different for two non-consecutive passes and the number of customers in each pass (including the passes between them) is small, the price difference between the passes is not sufficiently large to resolve the upgrade and downgrade incentives between the two passes.

Now, we wish to explicitly analyze how the required number of customers in each priority pass would vary when the number of customer types and passes grow at the same rate, with adjacent customer types getting closer and closer. For a clear picture of this relationship, we consider schemes in which each priority pass has the same number of customers and customer types in adjacent priorities are equally distanced. The following result shows that, when the number of passes grows, the number of customers in each priority that is sufficient and necessary for implementability grows towards infinity.

Proposition 13 (Type-separating schemes)**.** *Consider the linear multiplicative multi-type case, where $K = T$, $N^1 = N^2 = \cdots = N^T = m$ for some m, and $(\beta^t)_{t=1}^T$ decreases uniformly from c to 1 for some $c > 1$. Consider the scheme $q \in \mathcal{Q}((N^t)_{t=1}^T, K)$ where $q_t^t = m$ for every $t \in \{1, \ldots, T\}$, i.e., every t-th type customer is in the t-th priority pass. Let $M(T) = \frac{c}{6(c-1)}(T-1) + \frac{1}{6}$. The scheme q is implementable if and only if $m \geq M(T)$.*

Proof. Let $\Delta = \beta^1 - \beta^2 = \cdots = \beta^{K-1} - \beta^K = \frac{c-1}{T-1}$. Pick $j \in \{1, \ldots, K-2\}$. By the proof of Theorem 2 and choice of the scheme q, $\text{IC}_{j+2,j}$ holds if and only if

$$\frac{\beta^{j+2} + 2\Delta}{2}(2m-1) + \frac{\beta^{j+2} + \Delta}{2}(2m-1) - \frac{\beta^{j+2}}{2}(4m-1) \geq 0, \tag{31}$$

which holds if and only if $m \geq \frac{\beta^{j+2}}{6\Delta} + \frac{1}{2}$. Provided that $j \geq 2$, from $\text{IC}_{j+2,j}$ to $\text{IC}_{j+2,j-1}$, the left-hand side of (31) increases by $\frac{\beta^{j+2}+3\Delta}{2}(2m-1) - \beta^{j+2}m$. Therefore, conditional on $\text{IC}_{j+2,j-1}$, $\text{IC}_{j+2,j}$ holds if $\frac{\beta^{j+2}+3\Delta}{2}(2m-1) - \beta^{j+2}m \geq 0$, which is equivalent to $m \geq \frac{\beta^{j+2}}{6\Delta} + \frac{1}{2}$. Continuing this reasoning, we see that if $m \geq \frac{\beta^{j+2}}{6\Delta} + \frac{1}{2}$, $\text{IC}_{j+2,k}$ holds for every $k \in \{1, \ldots, j+1\}$. Since β^{j+2} is maximized at $j = 1$ for β^3, we see that the scheme q is implementable if and only if $m \geq \frac{\beta^3}{6\Delta} + \frac{1}{2} = \frac{c}{6(c-1)}(T-1) + \frac{1}{6}$. To complete the proof, set $M(T) = \frac{c}{6(c-1)}(T-1) + \frac{1}{6}$. □

For linear base utility functions, having equally distanced customer types and a uniform number of customers in each priority pass allows us to track all IC constraints. Specifically, the proof shows that, with the assumptions in Proposition 13, the scheme q is implementable if and only if p^* in Lemma 9, which binds IR_K and every local downward IC constraint, satisfies IC_{31}. By the linearity of $M(T)$, we see that as the customer types get closer and the number of priority passes gets larger, the required number of customers for implementability grows towards infinity.

Note that in Proposition 13, a larger value of c, which means a larger range for customer types, helps with implementability by lowering $M(T)$. However, there is a limit to how much this range parameter can help: Because $M(T)$ is bounded below by $T/6$, in Proposition 13, for fixed m and $c > 1$, the scheme q is not implementable if $T > 6m$. This observation highlights the role that the difference between "close-by" customer types plays in the scheme's implementability: Even if the customer types have a very wide range, if the (relative) differences between "close-by" customer types are not large enough, then the scheme is

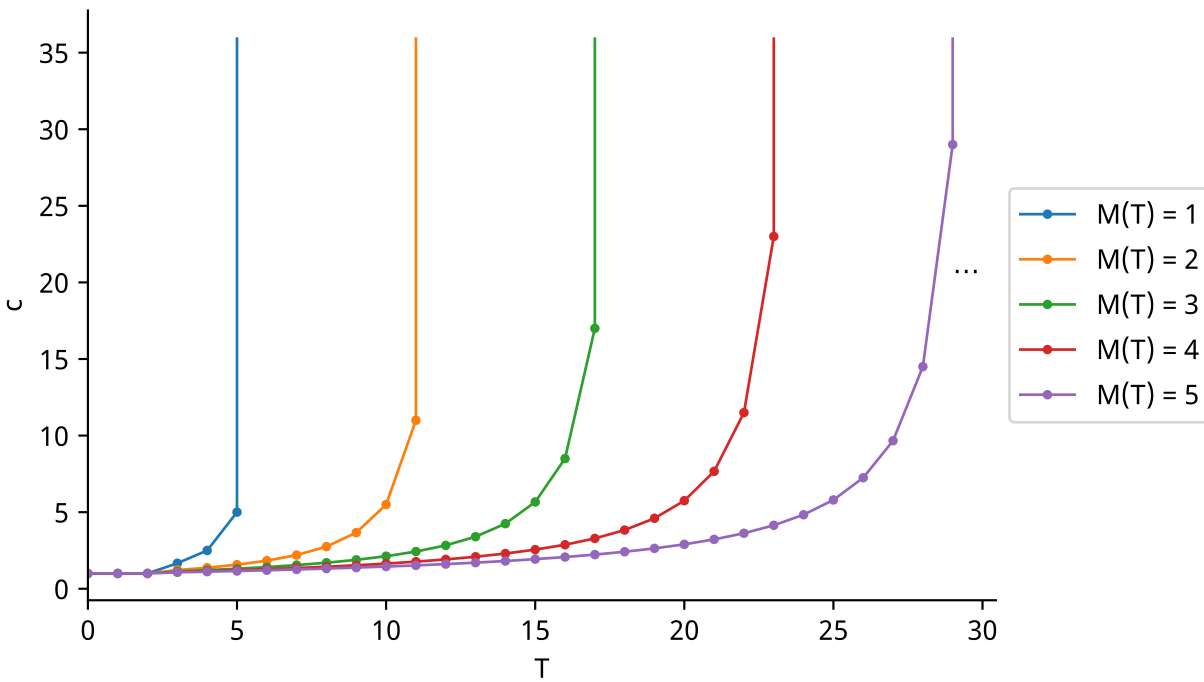

Figure 6: Level curves of $M(T)$ with respect to T and c

still not implementable without sufficiently many customers in each priority pass. Figure 6 illustrates this intuition. The curves are integer-valued level curves of $M(T)$. For a level curve with value M, every point (c', T') on the left of the curve means that, when $c = c'$ and $T = T'$, the scheme is implementable if every pass has M customers. In contrast, if the point is on the right, the scheme is not implementable if every pass has M customers. Given a level curve, we see that whenever the curve becomes vertical on a point's left, a larger c no longer helps with implementability, illustrating the limited role the range parameter can play in the scheme's implementability.

In summary, when there are multiple types of utility functions, the issue with resolving the upgrade and downgrade incentives is abated yet could persist. We have shown that to implement multi-pass schemes that are not implementable under the single-type case, there need to be large enough gaps between different customer types.

References

Adiri, Igal and Ury Yechiali (1974). "Optimal priority-purchasing and pricing decisions in non-monopoly and monopoly queues". In: *Operations Research* 22.5, pp. 1051–1066.

Alperstein, Hanna (1988). "Note—optimal pricing policy for the service facility offering a set of priority prices". In: *Management Science* 34.5, pp. 666–671.

Balachandran, K. R. (1972). "Purchasing priorities in queues". In: *Management Science* 18.5-part-1, pp. 319–326.

Bergstrom, Theodore, Lawrence Blume, and Hal Varian (1986). "On the private provision of public goods". In: *Journal of Public Economics* 29.1, pp. 25–49.

Bloch, Francis and Armando Gomes (2006). "Contracting with externalities and outside options". In: *Journal of Economic Theory* 127.1, pp. 172–201.

Bolton, Patrick and Mathias Dewatripont (2005). *Contract theory*. MIT press.

Carroll, Gabriel (2012). "When are local incentive constraints sufficient?" In: *Econometrica* 80.2, pp. 661–686.

Csorba, Gergely (2008). "Screening contracts in the presence of positive network effects". In: *International Journal of Industrial Organization* 26.1, pp. 213–226.

Gomes, Armando (2005). "Multilateral contracting with externalities". In: *Econometrica* 73.4, pp. 1329–1350.

Guesnerie, Roger and Jean-Jacques Laffont (1984). "A complete solution to a class of principal-agent problems with an application to the control of a self-managed firm". In: *Journal of Public Economics* 25.3, pp. 329–369.

Hassin, Refael and Moshe Haviv (1997). "Equilibrium threshold strategies: The case of queues with priorities". In: *Operations Research* 45.6, pp. 966–973.

Kamada, Yuichiro and Aniko Öry (2020). "Contracting with word-of-mouth management". In: *Management Science*.

Katz, Michael L. and Carl Shapiro (1986). "Technology adoption in the presence of network externalities". In: *Journal of Political Economy* 94.4, pp. 822–841.

Kuksov, Dmitri and J Miguel Villas-Boas (2010). "When more alternatives lead to less choice". In: *Marketing Science* 29.3, pp. 507–524.

Maskin, Eric and John Riley (1984). "Monopoly with incomplete information". In: *The RAND Journal of Economics* 15.2, pp. 171–196.

Park, Jeong-Yeol and SooCheong Shawn Jang (2013). "Confused by too many choices? Choice overload in tourism". In: *Tourism Management* 35, pp. 1–12.

Rasmusen, Eric B., J Mark Ramseyer, and John S Wiley Jr (1991). "Naked exclusion". In: *The American Economic Review*, pp. 1137–1145.

Segal, Ilya (1999). "Contracting with externalities". In: *The Quarterly Journal of Economics* 114.2, pp. 337–388.

Segal, Ilya and Michael D. Whinston (2003). "Robust predictions for bilateral contracting with externalities". In: *Econometrica* 71.3, pp. 757–791.

Shi, Zijun, Kaifu Zhang, and Kannan Srinivasan (2019). "Freemium as an optimal strategy for market dominant firms". In: *Marketing Science* 38.1, pp. 150–169.