Flash Pass*

Yuichiro Kamada†  Zihao Zhou‡

June 1, 2019

Preliminary Draft

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 number of customers buying a higher-or-equal priority pass and hence other customers’ purchase decisions have an externality on the customer’s valuation. This externality differentiates our model from the standard screening model and we discuss implementability of selling multiple passes for different utility patterns. In our model, the externality makes the implementation of multi-pass schemes difficult, an issue that persists even when there are multiple types of customer types.

---

*We are grateful to Ganesh Iyer, 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 seminar participants at the 40th Annual ISMS Marketing Science Conference for their feedbacks. All errors remain with us.

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

‡Zihao Zhou: Haas School of Business, University of California, Berkeley, Email: zihao_zhou@haas.berkeley.edu
# Contents

1 Introduction .................................................. 3

2 Model .......................................................... 5

3 Preliminaries .................................................... 6

4 Implementability .................................................. 10

5 Extension to Heterogeneous Utilities ......................... 13
   5.1 Two Utility Types ........................................ 13
   5.2 General Multiple Utility Types ......................... 16

6 Discussions ..................................................... 18
   6.1 Implementation and Externality ........................ 18
   6.2 Profits ...................................................... 20
   6.3 Implementation in Large Queues ....................... 20

7 Conclusions ..................................................... 22

A Auxiliary Results with Proofs .............................. 23

B Proofs of Results in Main Text ............................. 27
   B.1 Proof of Lemma 2 ........................................ 27
   B.2 Proof of Proposition 1 .................................... 28
   B.3 Proof of Proposition 2 .................................... 29
   B.4 Proof of Lemma 4 ........................................ 29
   B.5 Proof of Proposition 3 .................................... 30
   B.6 Proof of Theorem 1 ....................................... 30
   B.7 Proof of Proposition 4 .................................... 30
   B.8 Proof of Theorem 2 ....................................... 31
   B.9 Proof of Proposition 5 .................................... 31
   B.10 Proof of Proposition 6 ................................... 32
   B.11 Proof of Theorem 3 ...................................... 32
   B.12 Proof of Proposition 7 ................................... 33
   B.13 Proof of Proposition 10 .................................. 34
   B.14 Proof of Proposition 11 .................................. 34
   B.15 Proof of Theorem 4 ...................................... 35
1 Introduction

When access to a service facility is congested, it is common for service providers to implement multiple priority queues. Applications of such queuing management abound, such as amusement parks where customers queue for entry, computing clusters for which different workstations queue, and CUP power for which different programs within a computer queue. Incentives for selling multiple priority queues include social welfare consideration such as congestion management and profit consideration such as price discrimination. To give a concrete context, consider an amusement park whose customers queue in front of a single entrance. The park can sell a flash pass and a regular pass so that a customer with a flash pass has a high priority and always enters the park earlier than a customer holding a regular pass in the same queue. One key feature of multiple priority queues to a single entry is that, given that a customer’s utility varies with respect to her position in the queue, other customers’ purchase decisions affect her valuation of each pass. For example, in the two-pass case, if everyone else is buying the regular pass, then buying the flash pass guarantees her the front position of the queue. In this case, her valuation of the flash pass is high. In contrast, if many customers choose to buy the VIP pass, then her valuation of the flash pass is relatively low because of the congestion in the flash-pass queue.

In practice, we observe that parks usually sell only a small number of passes. Motivated by this observation, our focus is on the implementability of multi-pass schemes, i.e., we analyze whether a park is able to offer a large number of priority passes and price them in a way such that each pass has at least one customer. For example, we show that selling a different pass to each customer, when customers’ utility functions are homogeneous, is not implementable. When there are many utility types, we show that implementation of a multi-pass scheme depends on whether we can find large enough “gaps” between some “adjacent” types.

In the queuing literature from operations research (see e.g., Balachandran (1972), Adiri and Yechiali (1974), Hassin and Haviv (1997), and Alperstein (1988)), the pass-selling problem under our context is considered as selling multiple priority queues that can be ranked by their relative priorities. In their models, customers arrive sequentially. Each customer observes the state of the queues, makes a forecast about her waiting time before entrance conditional on her purchase decision, and chooses a pass to maximize her expected utility or chooses the outside option. In particular, Alperstein (1988) discusses the optimal pricing and the number of passes to sell. The author finds that the optimal number of passes to sell equals the number of customers, i.e., each customer is in her own pass. Given that the number of customers usually greatly exceeds the number of passes, a model that explains this disparity would be useful. Our modeling approach

---

1Not all parks implement a preemptive priority queue, i.e., whether a high-priority pass holder always enter before a low-priority pass holder, regardless of how long a low-priority pass holder has already waited. Whether preemptive or non-preemptive priority queue is better is out of the scope of this paper.
is static in the sense that we assume customers arrive simultaneously and analyze whether a pass-selling scheme can be implemented. In the aforementioned queuing literature, implementation does not pose a big difficulty: a pass without any customer gets more and more attractive as other passes get congested. For profit consideration, Alperstein (1988) implies that a large number of pass types should be sold. Therefore, in order to obtain a small number of passes as part of our prediction, it is necessary to change the modeling assumptions. Our approach can be viewed as changing the assumption about each customer’s information when she makes a purchase decision. Specifically, instead of assuming that each customer knows where exactly she is in the queue conditional on her purchase decision, she only knows the range of positions her pass is to cover.

We use the standard mechanism design approach to analyze the pass-selling problem, but our model differs from the standard screening model in the following sense. In the standard screening model such as Guesnerie and Laffont (1984) and Maskin and Riley (1984) about the pricing problem for a monopolist with incomplete information, for a finite number of utility types, the seller can separate different types of customers by manipulating the combination of price and quantity or “quality”. For the pass-selling problem we analyze, the park does not have such leeway in manipulating “quality” directly: each customer’s valuation of a pass depends on the purchase decisions of other customers rather than some “quality” that can be freely adjusted by the park. The existence of externality among customers is quintessential to queue management and makes our model different from the standard screening model. Because of this difference, implementability in our model does not extend from the aforementioned literature. We show that the existence of this externality makes implementation harder, i.e., the set of implementable schemes when there is externality is a subset of the same set when there is no externality (??).

To qualify our model, we do not claim that our model itself sufficiently explains the constraint on the number of passes. Indeed, the reasons for the constraint can be multi-fold in practice. For example, there could be considerable logistic costs involved with selling additional priority queues, and in light of this cost, the service provider may choose a small number of passes in practice. In addition, since holders of high priority passes can cut in a line, results of Allon and Hanany (2012) suggest that social norms in queues can also act as a constraint on the number of passes. One objective of this paper is to show that the externality can be helpful in explaining the constraint on the number of priority queues. We do not offer any normative argument about whether selling multiple priority queues is the optimal way to manage queues.

In Section 2 and Section 3, we introduce the model and present implementation results with respect to one utility type in Section 4. In Section 5, we extend our model to cases with multiple types of utility functions. We give a more detailed discussion of our results in Section 6 and conclude our paper in Section 7.
2 Model

There are \( N \geq 1 \) customers lining up outside an amusement for entrance. The possible positions in a queue range from 1 to \( N \). A customer’s valuation of visiting an amusement park depends solely on her position in the queue and the price she pays. A base utility function \( u : \mathbb{N} \rightarrow \mathbb{R} \) assigns a utility to each position in the queue, where the interpretation is that \( u(n) \) denotes the utility from being at the \( n \)-th position in the queue\(^2\). The function \( u \) is assumed to be strictly decreasing on \( \mathbb{N} \) in order to model the disutility of queuing. For convenience, we use \( u_n \) in place of \( u(n) \)\(^2\).

The park chooses the number \( K \in \mathbb{N} \cup \{0\} \) of different passes. For each \( k = 1, \ldots, K \), denote the \( k \)-th pass by \( \theta_k \). The relative priority of a pass varies inversely with respect to the pass index, i.e., if \( j < k \), then \( \theta_j \) has a higher priority than does \( \theta_k \). Define a strict order \( < \) on \( \{\theta_k : 1 \leq k \leq K\} \) such that \( \theta_j < \theta_k \) if and only if \( j < k \). The park sets \( p = (p_1, \ldots, p_K) \in \mathbb{R}^K_+ \) where \( p_k \) is the price of \( \theta_k \). Having observed \( p \), customers make purchase decisions simultaneously. Specifically, each customer buys one of the passes, or leaves the park to obtain a fixed utility of outside option denoted by \( u_0 \). Unless otherwise specified, we set \( u_0 = 0 \). Let \( \theta_0 \) denote the option of leaving the park and let \( p_0 = 0 \)\(^3\).

We model the above setup as a strategic-form game. Given \( (N, K, p, u) \), we define a strategic-form game \( G(N, K, p, u) = (N, A, (\pi_i)_{i=1}^N) \), where \( A = \times_{i=1}^N A_i \) with \( A_i = \{\theta_k\}_{k=0}^K \) being the action set for customer \( i \), and \( \pi_i : A \rightarrow \mathbb{R} \) is each customer’s (common) payoff function assumed to have a quasi-linear form: for every \( a \in A \), \( \pi_i(a) = \mu(a_i) - p(a_i) \), where \( \mu : A \rightarrow \mathbb{R} \) is specified below and \( p(a_i) \) is the price paid by \( i \), i.e., \( p(a_i) = p_k \) if \( a_i = \theta_k \).

For an action profile \( a \in A \), define \( \bar{q}(a) = (\bar{q}_k(a))_{k=1}^K \), where \( \bar{q}_k(a) = |\{a_i : a_i = \theta_k\}| \), which gives the number of customers in each pass resulting from \( a \). If a customer buys \( \theta_k \) with \( k \neq 0 \), she is guaranteed to be ahead of every customer in \( \theta_j \) if \( \theta_k < \theta_j \) and behind every customer in \( \theta_j \) if \( \theta_j < \theta_k \). As for each customer’s position relative to other customers buying the same pass, her position is assumed to be uniformly distributed on the positions of all customers in the same pass. These assumptions imply that \( \mu \) can be written as

\[
\mu(a) = \begin{cases} 
\frac{\sum_{n=Q_{k-1}(\bar{q}(a))}^{Q_k(\bar{q}(a))} u_n}{\bar{q}_k(a)} & \text{if } a_i \neq \theta_0 \\
u_0 & \text{if } a_i = \theta_0,
\end{cases}
\]

where for each \( k \) with \( 1 \leq k \leq K \) and \( q \in \bar{q}(A) \), we let \( Q_k(q) = \sum_{n=1}^k q_n \), which denotes the

\(^2\)We use \( \mathbb{N} \) to denote the set of strictly positive integers.

\(^3\)This specification features only one base utility function. Section 5 considers the case with multiple utility types.

\(^4\)We deliberately do not define \( < \) for \( \theta_0 \), because \( \theta_0 \) does not have relative priority to other passes.
<table>
<thead>
<tr>
<th>Customer</th>
<th>Pass Bought</th>
<th>Price Paid</th>
<th>Expected Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$\theta_1$</td>
<td>$p_1$</td>
<td>$\frac{u_1 + u_2}{2} - p_1$</td>
</tr>
<tr>
<td>B</td>
<td>$\theta_1$</td>
<td>$p_1$</td>
<td>$\frac{u_1 + u_2}{2} - p_1$</td>
</tr>
<tr>
<td>C</td>
<td>$\theta_2$</td>
<td>$p_2$</td>
<td>$\frac{u_1 + u_2 + u_5}{3} - p_2$</td>
</tr>
<tr>
<td>D</td>
<td>$\theta_2$</td>
<td>$p_2$</td>
<td>$\frac{u_1 + u_2 + u_5}{3} - p_2$</td>
</tr>
<tr>
<td>E</td>
<td>$\theta_2$</td>
<td>$p_2$</td>
<td>$\frac{u_1 + u_2 + u_5}{3} - p_2$</td>
</tr>
<tr>
<td>F</td>
<td>$\theta_3$</td>
<td>$p_3$</td>
<td>$u_6 - p_3$</td>
</tr>
<tr>
<td>G</td>
<td>$\theta_0$</td>
<td>$p_0 (= 0)$</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 1: An example of a scheme: $N = 7$, $K = 3$, the scheme in consideration is $(2, 3, 1)$.

end position of $\theta_k$, and $Q_0(q) = 0$. Note that the utility of each customer’s purchase decision also depends on the purchase decisions of other customers, because a customer’s position in a queue depends on other customers’ decisions. Given $N$ and $K$, we define $Q(N, K) = \{ q \in \mathbb{N}^K : \sum_{k=1}^{K} q_k \leq N \}$ as the set of schemes for $(N, K)$\footnote{In words, we allow customers to be excluded from a scheme but we do require that a scheme has at least one customer for each pass.}. See Figure 1 for an example that illustrates various parts of the model.

3 Preliminaries

With the specification on payoff functions, we define implementability. In short, a scheme is implementable if it results from each customer’s optimal purchase decision that takes into account the other customers’ decisions.
**Definition 1 (Implementation).** Fix \((N, K, u)\).

(a) A price vector \(p\) implements a scheme \(q \in Q(N, K)\) if \(G(N, K, p, u)\) has a (pure-strategy) Nash equilibrium \(a^*\) such that \(q(a^*) = q\). A scheme \(q\) is implementable for \((N, K, u)\) if there exists a price vector \(p\) that implements \(q\).

(b) A tuple \((N, K, u)\) is implementable if there exists a scheme that is implementable for \((N, K, u)\).

When \((N, K, u)\) is without ambiguity, we only write “\(q\) is implementable” without including \((N, K, u)\).

To give an intuitive understanding, we provide an equivalent formulation of implementation based on incentive constraints. To do so, we first define the utility from deviating from a strategy profile. Given \(q \in Q(N, K)\), define \(v : \{\theta_k\}_{k=1}^K \times \{\theta_j\}_{j=0}^K \rightarrow \mathbb{R}\) such that

\[
v(\theta_k; \theta_j) := \begin{cases} 
\frac{\sum_{n=Q_k(q)}^{Q_k(q+1)} u_n}{q_k + 1} & \text{if } \theta_j < \theta_k, \\
\frac{\sum_{n=Q_k(q+1)}^{Q_k(q+2)} u_n}{q_k + 1} & \text{if } \theta_j = \theta_k, \\
\frac{\sum_{n=Q_k(q+2)}^{Q_k(q+3)} u_n}{q_k + 1} & \text{if } \theta_j > \theta_k \text{ or } \theta_j = \theta_0
\end{cases}
\]

which gives the utility of switching: if a customer switches from pass \(\theta_j\) to pass \(\theta_k\), then her utility would be \(v(\theta_k; \theta_j)\). For completeness, define \(v(\theta_0; \theta) = u_0\) for each \(\theta \in \{\theta_k\}_{k=0}^K\). Abuse notation to write \(v(\theta) := v(\theta; \theta)\) for each \(\theta \in \{\theta_k\}_{k=0}^K\). If a function \(v\) is specified in this manner using \(u\), we say \(v\) is associated with \(u\).

With a fixed scheme \(q\), a few properties of \(v\) are immediately clear. Firstly, with fixed \(\theta_j\) and \(\theta_k\) such that \(\theta_j < \theta_k\), \(v(\theta_k; \theta_j) = v(\theta_k; \theta_{k-1}) > v(\theta_k)\) and \(v(\theta_j; \theta_k) = v(\theta_j; \theta_{j+1}) < v(\theta_j)\), i.e., the utility of switching to a different pass only depends on whether the old pass has a higher or lower priority, but not on “how much” higher or lower. Secondly, for every \(\theta_j\) and \(\theta_k \neq \theta_0\), \(v(\theta_k; \theta_j) > v(\theta_{k+1}; \theta_j)\), i.e., switching to a higher priority pass is always preferred to switching to a lower priority pass if the prices for those passes are the same. Thirdly, for \(\theta_j, \theta_k \neq \theta_0\), \(v(\theta_k; \theta_j) \geq v(\theta_k; \theta_{j+1})\), i.e., switching from a higher priority pass is weakly better than switching to the same pass from a lower priority pass. Lastly, if a customer in \(\theta_j\) with \(j \neq 0\) switches to \(\theta_k\) such that \(\theta_j < \theta_k\), she improves the utility in \(\theta_k\) since the front position of \(\theta_k\) is one position ahead after the switching. In contrast, if \(\theta_k < \theta_j\), the utility in \(\theta_k\) decreases since now the last position of this pass is one position behind.
Now we formulate implementability from the perspective of incentive constraints. Fix \((N, K, u)\) and \(i \in \{1, \ldots, N\}\). Consider an action profile \(a^* \in A\) such that \(q := q(a^*) \in Q(N, K)\). Let \(p\) be a price vector. By the definition of the action set, there exists \(j \in \{0, \ldots, K\}\) such that \(\theta_j = a^*_i\). Firstly, for \(j \neq 0\), we say that \((p, q)\) satisfies \(j\)'s individual-rationality constraint (henceforth IR\(_j\)) if \(i\) would be weakly better off buying \(\theta_j\) than leaving the park, i.e.,

\[
v(\theta_j) - p_j \geq u_0 = 0. \tag{IR\(_j\)}
\]

Secondly, for \(j \in \{0, \ldots, K\}\) such that \(\theta_j = a^*_i\) and \(k \in \{1, \ldots, K\}\), we say that \((p, q)\) satisfies \(i\)'s incentive-compatibility constraint with respect to \(\theta_k\) (henceforth IC\(_{jk}\)) if \(i\) has no incentive to switch to \(\theta_k\), i.e.,

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

By the definition of the incentive constraints above, the following lemma is immediate.

**Lemma 1.** Fix \((N, K, u)\). A scheme \(q \in Q(N, K)\) is implementable if and only if there exists a price vector \(p\) such that \((p, q)\) satisfy IR\(_k\) and IC\(_{jk}\) for every \(j \in \{0, \ldots, K\}\) and \(k \in \{1, \ldots, K\}\).

When every customer buys some pass, for \(K = 1\), there is only one possible scheme regardless of implementability: all customers are in the same pass. Provided that \(v(\theta_1) \geq 0\), the park just needs to set \(p_1 \leq v(\theta_1)\) to implement the scheme. Since implementability of a single-pass scheme is easy to check, we focus on schemes with two or more passes.

Fix an IC constraint IC\(_{jk}\) where \(j\) and \(k\) are from \(\{1, \ldots, K\}\). We say IC\(_{jk}\) is a downward IC constraint if \(j < k\); a local downward IC constraint if \(k = j + 1\); a upward IC constraint if \(j > k\); a local upward IC constraint if \(j = k + 1\). In the standard screening model, e.g., Guesnerie and Laffont (1984) and Maskin and Riley (1984), the set of constraints can be reduced so that at the solution of the seller’s optimization problem, only the IR constraint of the lowest-priority pass and the local downward IC constraints are binding. Similarly, in our model, we show that at the profit-maximizing price vector of a fixed implementable scheme, only IR\(_k\) and all local downward IC constraints are binding. We also show that in our model, IC\(_{jk}\) holds for every \(j\) and \(k\) with \(1 \leq j < k \leq K\) if IC\(_{j,j+1}\) holds for every \(1 \leq j < K\), but it is not true that if IC\(_{j,j-1}\) holds for every \(1 < j \leq K\) then IC\(_{kj}\) holds for every \(j\) and \(k\) with \(1 \leq j < k \leq K\). We summarize these properties of the IR and IC constraints in the following lemma, as well as providing a condition that is useful for proving some negative results about implementation.

**Lemma 2** (Constraint reduction). Fix \(q \in Q(N, K)\) and a price vector \(p \in \mathbb{R}^K_+\).

(a) Assume \(K \geq 2\). Given \(\theta_j < \theta_k\), if \((p, q)\) satisfies local downward IC constraint, then \((p, q)\) satisfies IC\(_{jk}\), i.e., if all the local downward IC constraints between two different passes hold, then the downward IC constraint for the two passes holds.
(b) Assume $K \geq 2$. Given $\theta_j < \theta_k$, if $(p, q)$ satisfies $IR_k$ and $IC_{jk}$, then $(p, q)$ satisfies $IR_j$.

(c) For every $\theta_j < \theta_k$, if $(p, q)$ satisfies both $IC_{jk}$ and $IC_{kj}$, then

$$\nu(\theta_j; \theta_k) - \nu(\theta_k) \leq \nu(\theta_j) - \nu(\theta_k; \theta_j).$$

(ID$_{jk}$)

We say that $q$ satisfies ID$_{jk}$ if the inequality above holds and call the set of ID$_{jk}$ for all $\theta_j < \theta_k$ the increasing difference (ID) conditions.

**Lemma 3** (Necessity of ID conditions). If $q \in Q(N, K)$ is implementable, then $q$ satisfies ID$_{jk}$ for each pair $\theta_j$ and $\theta_k$ with $j, k \geq 1$ and $\theta_j < \theta_k$.

To see why this is true, consider $\theta_j$ and $\theta_k$ with $\theta_j < \theta_k$. IC$_{jk}$ implies an upper bound on $p_j - p_k$, while IC$_{kj}$ implies a lower bound on $p_j - p_k$. Implementability requires that the upper bound is always above the lower bound, and hence we obtain ID$_{jk}$. We illustrate this argument in the following simple example.

**Example 1** (ID condition). Consider a case with $N = 3$, $K = 2$, $q = (1, 2)$. Assume there exists a price vector $(p_1, p_2)$ that implements $q$. IC$_{12}$ implies

$$p_1 - p_2 \leq \nu(\theta_1) - \nu(\theta_2; \theta_1) = u_1 - \frac{u_1 + u_2 + u_3}{3} = \frac{2u_1 - u_2 - u_3}{3},$$

while IC$_{21}$ implies

$$p_1 - p_2 \geq \nu(\theta_1; \theta_2) - \nu(\theta_2) = \frac{u_1 + u_2}{2} - \frac{u_2 + u_3}{2} = \frac{u_1 - u_3}{2}.$$

Combining, we obtain:

$$\frac{u_1 - u_3}{2} = \nu(\theta_1; \theta_2) - \nu(\theta_2) \leq p_1 - p_2 \leq \nu(\theta_1) - \nu(\theta_2; \theta_1) = \frac{2u_1 - u_2 - u_3}{3}. \quad (1)$$

□

The ID conditions are commonly assumed in the mechanism design literature and are necessary and sufficient for local downward and upward constraints to hold. In other words, ID conditions are necessary and sufficient for implementation if (1) every upward IC constraint holds whenever every local upward IC constraint holds; and (2) every downward IC constraint holds whenever every local downward IC constraint holds.\footnote{See Chapter 2 of Bolton and Dewatripont (2005) for a summary of these results.} In our model, the ID conditions are necessary for implementation as Lemma 3 shows, yet the ID conditions are not sufficient. The only
special case where the ID conditions are necessary and sufficient for implementability is when $K = 2$ and every customer buys some pass\footnote{We consider the case where every customer buys some pass so that we do not need to consider IC$_{0j}$ for checking implementability.} where the ID conditions are necessary and sufficient for the existence of prices that make all the constraints hold. Given the necessity of the ID conditions, violating any ID condition would imply that a scheme is not implementable.

4 Implementability

In this section, given $N$ and $K$, we present our results about implementability of a scheme with respect to different types of base utility functions. For the base utility function $u$, we mainly consider the following three types of functions:

- Concave base utility function ($u_n - u_{n+1} \leq u_{n+1} - u_{n+2}$ for each $n$): The concave case applies when the customers incur an opportunity cost for the queuing time. Specifically, assume that a customer’s utility from a park depends on her time spent in the park. If she spent $x$ units of time in the park, where $0 \leq x \leq T$ for some $T > 0$, then her utility would be $y(x)$, where $y' > 0$ and $y'' < 0$, i.e., the customer enjoys spending time in the park but faces diminishing marginal utility with respect to time. Supposing being at $n$-th position means that she will wait for $n$ units of time before going into the park, her utility is $u_n = y(T - n)$, which is decreasing and concave 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 from operations research, e.g., Balachandran (1972) and Adiri and Yechiali (1974).

- Convex base utility function ($u_n - u_{n+1} \geq u_{n+1} - u_{n+2}$ for each $n$): The convexity assumption can be justified if queuing is inherently unpleasant and a customer becomes less sensitive about an incremental increase in the waiting time when the waiting time is longer. In addition, the convexity assumption also holds when the customer obtains an instantaneous utility upon entrance and this utility is exponentially discounted with respect to waiting time.

**Proposition 1** (Implementation with non-linear concave utility). Fix $(N, K, u)$ where $K > 1$, $N > 2$, and $u$ is concave and non-linear. If $q \in Q(N, K)$, then $q$ is not implementable.

That is, no scheme with two or more passes such that more than one customer exists in at least one pass is implementable under a non-linear concave base utility function.
The proof of the above proposition shows that when the utility is non-linearly concave, some ID condition is violated. A special case of non-linear concavity is strict concavity: for every \(1 \leq n \leq N - 2\), \(u_n - u_{n+1} < u_{n+1} - u_{n+2}\). For intuition, consider the following example.

**Example 2** (Example 1 continued). For the scheme \(q\) to be implementable, ID\(_{12}\) in Equation (1) requires that
\[
\frac{u_1 - u_3}{2} \leq \frac{2u_1 - u_2 - u_3}{3},
\]
and a simple manipulation shows that this inequality implies \(u_1 - u_2 \geq u_2 - u_3\). This, however, contradicts the strict concavity of \(u\) and hence ID\(_{12}\) is violated.

The key intuition is that each customer has some weight. In other words, when a customer switches to a different pass, she creates some externality for customers in the new pass. Indeed, in a hypothetical situation in which each customer creates no externality, i.e., \(v(\theta_k; \theta_j) = v(\theta_k)\) for every \(j\) and \(k\), then IC\(_{12}\) for the aforementioned example implies \(p_1 - p_2 \leq u_1 - \frac{u_2 + u_3}{2}\) and IC\(_{21}\) implies \(p_1 - p_2 \geq u_1 - \frac{u_2 + u_3}{2}\), and the scheme would become implementable as in the standard model.

With a simple modification of the proof of Proposition 1, we can show that when \(u\) is convex, i.e., \(u_i - u_{i+1} \geq u_{i+1} - u_{i+2}\), all ID conditions are satisfied. Since the ID conditions can be shown to be sufficient for implementation with \(K = 2\), we see that for convex \(u\), every scheme is implementable, as follows.

**Proposition 2** (Two-pass implementation with convex utility). Fix \(N\) and a convex \(u\). For every scheme \((q_1, q_2) \in Q(N, 2)\) such that \(v(\theta_2) \geq u_0\), we have \((q_1, q_2)\) is implementable for \((N, 2, u)\).

Our next result concerns the implementability when \(K = N\), which we show to be negative. Before we present the result, we first introduce an intermediate result that is useful for later results. The lemma is about the necessary and sufficient conditions for implementation which are the counterpart of Rochet (1987, Theorem 1).

**Lemma 4** (Implementation condition). Fix \((N, K, u)\) where \(N \geq K \geq 2\). Let \(q \in Q(N, K)\) be such that every customer buys some pass. Let \(p\) be a 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 \(k = 1, \ldots, K - 1\). The scheme \(q\) is implementable if and only if \((p^*, q)\) satisfies every upward IC constraint and \(v(\theta_K) \geq 0\).

We relate the above lemma to Rochet (1987, Theorem 1). By Rochet (1987, Theorem 1), the necessary and sufficient condition for a scheme \(q\) to be implementable is that given any finite finite cycle \(k_0, k_1, \ldots, k_m, k_{m+1} = k_0\) in \(\{0, \ldots, K\}\), if we bind IC\(_{k_l; k_{l+1}}\) for \(l = 0, \ldots, m - 1\) and

\(^8\)A few results later assume strict concavity instead of non-linear concavity.
combine them to get a price difference $p_{k_0} - p_{k_1}$, then this price difference must satisfy $IC_{k_0k_1}$. In our model, Lemma 2 shows that if we bind $IR_k$ and every local downward IC constraint as we define $p^*$ in Lemma 4, all downward IC constraints and $IR_j$ for every $j$ with $1 \leq j \leq K$ hold. Hence in the above lemma, it only remains to check whether given the scheme $q$, $(p^*, q)$ satisfies every upward IC constraint.

In the standard screening model, with a fixed scheme $q$, $(p^*, q)$ would satisfy every local downward IC constraint, which makes every downward and local upward IC constraint hold thanks to the ID conditions. In addition, since in the standard screening model, every upward IC constraint holds when every local upward IC constraint holds, we would have $(p^*, q)$ satisfies all the incentive constraints and hence implement the scheme in the standard screening model. What makes our model different is that it is not true that every upward IC constraint holds whenever every local upward IC constraint holds, hence the result of Lemma 4. Alperstein (1988) finds that the optimal number of passes equals the number of customers. In the following result, we show that such a scheme is not implementable in our model.

**Proposition 3** (Implementation with $N = K$). Fix $(N, K, u)$. If $N = K > 2$, then the unique scheme where each type of pass has exactly one customer is not implementable.

We give some intuition for this result. 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 the second pass, since she pays less but still has the chance of being at the same position after switching. To incentivize the first customer against switching, $p_1 - p_2$ needs to be small. Similarly, $p_2 - p_3$ needs to be small so that the second customer does not want to switch to the third pass. For the third customer, however, upgrading to the first pass can be tempting, since by switching can the third customer strictly improve her position in the queue, albeit at a higher price. Hence $p_1 - p_3$ needs to be large enough so that the third customer does not want to upgrade to the first pass. We show in the proof that the upper bound on $p_1 - p_3$ implied by those for $p_1 - p_2$ and $p_2 - p_3$ is strictly less than the lower bound on $p_1 - p_3$, which is a contradiction.

Another special case of the base utility function $u$ is the linear utility function. Since the linear case is weakly convex, we know from Proposition 2 that every scheme in $K = 2$ is implementable. For $K > 2$, we show that $(N, K, u)$ is not implementable for any linear $u$.

**Theorem 1** (Implementation with linear utility). Fix $(N, K, u)$ where $u$ is linear. Assume there is $q \in Q(N, K)$ such that $v(\theta_K) \geq u_0$. $(N, K, u)$ has an implementable scheme if and only if $K \leq 2$.

Results we have derived so far hinge upon the externality customers create on each other. We have shown that the presence of this externality creates a conflict between the upgrade and downgrade incentives, the feasibility of whose resolution depends on the underlying base utility.
function. In the next section where we discuss cases with multiple types of base utility functions, we show that this conflict between the upgrade and downgrade incentives can persist.

5 Extension to Heterogeneous Utilities

We have analyzed the model that has only one base utility function for the customers. Call this case the single-type case. We now consider the cases where customers have heterogeneous utilities which we call the multi-type case. To be precise, assume each customer’s base utility function comes from \{u^t\}_{t=1}^T, where \(1 \leq T \leq N\) and \(t\) is the index for utility type. For each \(t = 1, \ldots, T\), let \(N^t\) be the number of customers with base utility function \(u^t\), where the outside option of each type is assumed to be 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 analogously defined strategic-form game. The definition of implementability can be straightforwardly extended to the multi-type case.

Define the set of schemes as

\[ Q\left((N^t)_{t=1}^T, K\right) = \left\{(q^t)_{t=1}^T : q^t \in (\{0\} \cup \mathbb{N})^K, \sum_{t=1}^T q^t_k > 0 \text{ for every } k \text{ with } 1 \leq k \leq K \right\}. \]

The restriction that \(\sum_{t=1}^T q^t_k > 0\) for every \(k\) with \(1 \leq k \leq K\) ensures every pass has at least one customer, which is analogous to the restriction imposed in the definition of the set of schemes in the single-type case. For \(\theta_j\) and \(\theta_k\), IC\(_{jk}\), the IC constraint for type \(t\), and IR\(_j\), the IR constraint of this type, can both be analogously defined for the multi-type case. Different from the IC constraints in the single-type case, IC\(_{jk}\) may not need to hold for implementation. To be precise, for a scheme \(q\) to be implementable, IC\(_{jk}\) needs to hold only if there is some customer of type \(t\) in \(\theta_j\), i.e., \(q_j^t > 0\). Given a scheme \(q\), for two passes \(\theta_j\) and \(\theta_k\), we say IC\(_{jk}\) or IR\(_j\) is active if and only if the constraint needs to hold for the scheme to be implementable.

Given \(\{u^t\}_{t=1}^T\), assume for every \(\tau_1\) and \(\tau_2\) with \(1 \leq \tau_1 \leq \tau_2 \leq T\) implies at every position \(n\), \(u_n^{\tau_1} > u_n^{\tau_2}\) and \(u_n^{\tau_1} - u_{n+1}^{\tau_1} \geq u_n^{\tau_2} - u_{n+1}^{\tau_2}\). We show later that some results we have in the single-type case can be extended to the multi-type case under these assumptions.

5.1 Two Utility Types

For now, we consider the case where there are two types of utility functions, which we call the high and low types. Denote the high type’s utility function by \(u^h\) and that of the low type’s by \(u^l\). As before, at every position \(n\) in a scheme \(q\), we assume \(u^h_n > u^l_n\) and \(u^h_n - u^l_{n+1} \geq u^h_n - u^l_{n+1}\). Given \(N^l\), \(N^h\) and \(K\), and a scheme \(q \in Q((N^h, N^l), K)\), when both IC\(_{jk}^h\) and IC\(_{jk}^l\) are active, one of them is sufficient for the other to hold.
For \( j \in \{0, \ldots, K\} \) and \( k \in \{1, \ldots, K\} \), define \( IC_{jk} \) such that
\[
IC_{jk} = \begin{cases} 
IC^h_{jk} & \text{if } IC^h_{jk} \text{ implies } IC^l_{jk} \\
IC^l_{jk} & \text{if } IC^l_{jk} \text{ implies } IC^h_{jk} 
\end{cases}
\]

With this notation, we can reuse and extend the definition of implementability with respect to incentive constraints in the single-type case to the two-type case. For \( \theta_j < \theta_k \), let \( t_j \) be the type such that \( IC^l_{jk} = IC_{jk} \) and \( t^j \) be the type such that \( IC^l_{kj} = IC_{kj} \). It is immediately clear that \( IR^l_j = IR_j \).

The addition of this heterogeneity introduces some complications for constraint reduction. In the single-type case, \[\text{Lemma 2} \] and \[\text{Lemma 4} \] help us reduce the set of downward IC constraints to the set of local downward IC constraints and \( \{IR_j : 1 \leq j \leq K\} \) to only \( IR_K \). Generally, neither types of constraint reductions hold in the two-type case. In \[\text{Appendix A} \], we show with two lemmas that under some conditions, the reductions of IR and downward IC constraints hold. With these two lemmas, we derive a result similar to \[\text{Lemma 4} \] in \[\text{Appendix A} \]. We impose an intuitive restriction that allows us to use the two lemmas for constraint reduction. The restriction that \( q^j_t > 0 \) implies \( q^l_k > 0 \) for all \( j \) and \( k \) with \( 1 \leq j \leq k \leq K \) eliminates schemes where a lower-priority pass has only high-type customer and a higher-priority pass has at least one lower-type customer.

Now we focus on the linear utility, a special case of concave utility functions. For \( n \geq 1 \), let \( u^h_n = \alpha^h - \beta^h nd \) and \( u^l_n = \alpha^l - nd \), where \( \alpha^h \geq \alpha^l \), \( d > 0 \) and \( \beta^h \geq 1 \). Given \( N^l \) and \( N^h \), we keep the assumption that \( u^h_n \geq u^l_n \) for \( 1 \leq n \leq N \). Since \( q \) is implementable, from the proof of \[\text{Theorem 1} \] we have \( K \leq 4n^P \). The next result presents the conditions under which \( ((N^h, N^l), K, (u^h, u^l)) \) is implementable for \( K = 4 \). The intuition of the condition is that the slopes of \( u^h \) and \( u^l \) need to be sufficiently different from each other.

**Theorem 2** (Implementation with Two Linear Utilities). Consider \( ((N^h, N^l), K, (u^h, u^l)) \) where \( K = 4 \). Suppose \( u^l_n = \alpha^l - nd \) and \( u^h_n = \alpha^h - \beta^h nd \), where \( d > 0 \), \( u^h_{N^h+N^l} - u^l_{N^h+N^l} > 0 \) and \( \beta^h > 1 \). Consider \( q \in Q \) \( ((N^h, N^l), K) \) such that \( \sum_{k=1}^{K} q^k_t \geq N^t \) for \( t = l, h \). Assume \( v^l(\theta^h_k) \geq 0 \). For each such scheme \( q \), there exists \( b \geq 0 \) such that \( q \) is implementable if and only if the following two conditions hold.

(a) \( q^h_1 + q^h_2 = N^h \) and \( q^l_1 + q^l_2 = N^l \).

(b) \( \beta^h \geq b \).

Part (a) of \[\text{Theorem 2} \] is a special case for the restriction that \( q^j_t > 0 \) implies \( q^l_k > 0 \), which we have imposed for constraint reduction. For two concave utilities, the restriction that \( q^j_t > 0 \)
implies \( q_k^j > 0 \) is necessary yet not sufficient for implementation. The following proposition shows that implementation with two concave base utility functions implies a condition stronger than the restriction for constraint reduction.

**Proposition 4** (Implementable schemes with two concave utilities). Consider \(((N^h, N^l), K, (u^h, u^l))\) where both \( u^h \) and \( u^l \) are concave and non-negative. Assume \( q \) is implementable. If \( u^h \) or \( u^l \) is strictly concave, or if \( u^h_n - u^l_n \) is strictly decreasing in \( n \) for \( 1 \leq n \leq N \), then \( q_j^j > 0 \) implies \( q_k^j = 0 \) for all \( j \) and \( k \) with \( 1 \leq j < k \leq K \).

In the standard screening problem, the following monotonicity property holds: If some type buys some pass, then all buyers from higher types buy the same or a higher-priority pass. This property does not hold in our model. There are two different intuitions on why monotonicity does not hold. We illustrate them with the following two examples.

**Example 3** (Non-monotonicity 1). Let \(((N^h, N^l), K, (u^h, u^l))\) be such that \( N^h = N^l = 1 \) and \( K = 1 \). Let \( u^h = (4, 0) \) and \( u^l = (5, 1) \) be the utility of the two types on the first two positions of a queue. Consider the scheme \( q \) in which there is only the low type customer. It is straightforward to verify that \( p_1 = 4 \) implements \( q \) and the high type customer buys no pass. In this example, monotonicity fails because given the price, the second position is bad enough even for the high type such that a high type would not want to join the queue.

**Example 4** (Non-monotonicity 2). Consider \(((N^h, N^l), K, (u^h, u^l))\) where \( N^h = 2, N^l = 1 \), and \( K = 2 \). Let \( u^l = (6, 2, 1, 0) \) and \( u^h = (6.3, 2.2, 1.1, 0.05) \) be the base utility functions of the two types on the first four position in the queue. Consider the scheme \( q = (q_4^1, q_1^h) \) such that \( q_1^h = (0, 2) \) and \( q_1^l = (1, 0) \). It is straightforward to check that \( p_1 = 4.65 \) and \( p_2 = 1.65 \) implement \( q \), and in this scheme, the high type customers buy the second pass while the low type customer buys the higher priority pass. In this example, monotonicity fails because when a high type attempts to buy a first pass, she would create additional congestion to this pass, and the decrease in utility of the first pass can be larger than the surplus formerly enjoyed by the first-pass holders when the base utility functions are convex and a high type customer is not very different from a low type.

In Proposition 12 in the Appendix, we show that the monotonicity property holds for multi-type with concave base utility functions.

The result below shows that the introduction of utility heterogeneity makes some multi-pass scheme that is not implementable in the single-type case implementable in the two-type case.

**Proposition 5** (Two-pass implementation with two strictly concave utilities). Let \(((N^h, N^l), K, (u^h, u^l))\) where \( K = 2 \). Assume \( u^h = u \) and \( u^l = \beta^l u^l \), where \( u \) is a strictly concave base utility function and \( \beta^l \in (0, 1) \). Let \( q \in Q((N^h, N^l), K) \) such that \( \sum_{k=1}^K q_k^t = N^t \) for \( t = l, h \) and \( c^l(\theta_2) \geq 0 \). \( q \) is
implementable if and only if $\beta^l \leq \frac{c(\theta_{1})-c(\theta_{2};\theta_{1})}{c(\theta_{1};\theta_{2})-c(\theta_{2})}$ where $v$ is associated with $u$. Furthermore, if $q$ is implementable, then $q^h_1 = N^h$ and $q^l_2 = N^l$.

Again, the condition for implementability is about the two types being sufficiently different from each other. We will see that this condition is still important as we extend to the general multi-type case.

5.2 General Multiple Utility Types

Results we have proven so far in this section can be extended to the case with an arbitrary number of utility types. Suppose that there are $T$ types of basic utility functions. Similar to the two-type case, we use superscripts to distinguish variables for different utility types and define other variables similarly. Again for every pass $\theta_j$ and $\theta_k$, define $t_j$, $^k$, IC$_{jk}$, IC$_{kj}$, and IR$_j$ similarly to the multi-type case. We extend the lemmas for the two-type case to the general multi-type case in Appendix A. We have an extension of [Lemma 10] with which we can check the implementability of a scheme satisfying the restriction that $t_j \leq t_k$ for every $j$ and $k$ with $1 \leq j < k \leq K$. Similar to the restriction introduced in the two-type case, the restriction for the multi-type case implies that every customer in a lower-priority pass cannot have a strictly higher type than does any customer in a higher-priority pass. The following proposition shows that implementation with multiple concave base utility functions implies a condition stronger than the imposed restriction for constraint reduction.

**Proposition 6** (Implementable schemes with multiple concave utilities). Fix $((N^t)_{t=1}^T, K, (u^t)_{t=1}^T)$. Assume either $u^t$ is strictly concave for all $1 \leq t \leq T$ or $u^{\tau_1} - u^{\tau_2}$ is strictly decreasing for all $\tau_1$ and $\tau_2$ with $1 \leq \tau_1 < \tau_2 \leq T$. If $q$ is implementable, then $t_j \leq ^k$ for every $j$ and $k$ with $1 \leq j < k \leq K$.

An immediate implication of **Proposition 6** with multiple concave base utility functions is that $^k \leq t_k \leq ^{k+1} \leq t_{k+1}$ for all $1 \leq k < K$. We now provide a general implementability result for linear utilities. The intuition of the result is that when the slopes are sufficiently different, the price-difference upper bounds implied by all the local downward IC constraints become sufficiently large to satisfy all the upward IC constraints, making a scheme implementable.

**Theorem 3** (Implementation with multiple linear utilities). Fix $((N^t)_{t=1}^T, K, (u^t)_{t=1}^T)$. For every $t$ with $1 \leq t \leq T$, assume $u^t_n = \alpha^t - \beta^t d$, where $d > 0$. Assume $\alpha^1 \geq \alpha^2 \geq \cdots \geq \alpha^T > 0$ and $\beta^1 > \beta^2 > \cdots > \beta^T = 1$. Let $q = Q((N^t)_{t=1}^T, K)$ be such that $\sum_{k=1}^K q^t_k = N^t$ for every type $t$ and $v^T(\theta_k) \geq 0$. Then for each $k = 2, \ldots, K$, there exists a mapping $f_k$ that assigns a nonnegative real number in $[0, \infty)$ to each $(\beta^1, \ldots, \beta^{t-1}; q)$, such that $q$ is implementable if and only if the following two conditions hold:

16
(a) \( t_j \leq t^k \) for every \( j \) and \( k \) with \( 1 \leq j < k \leq K \)

(b) For every \( k \) with \( 2 \leq k \leq K \), \( \beta^t_k \leq f_k(\beta^t_1, \ldots, \beta^{t_{k-1}}; q) \).

For linear utilities, if \( t_j > t^l \) for some \( j \) and \( l \) with \( 1 \leq j < l \leq K \), then IC\(_{ij} \) does not hold, hence part (a) of the theorem. Part (b) of the theorem, the inequality in the statement, is essentially a requirement that the slopes need to be different enough. Similar intuitions can be applied to the non-linear concave case. For the general non-linear concave case, we focus on the case of \( K = 2 \), which is not implementable for the single-type case. When \( K = 2 \), implementation is possible for \( t^1 < t^2 \). Specifically, given a scheme \( q \) with \( K = 2 \) and non-linear concave utility functions, it is necessary that \( t^1 = t^2 - 1 \). Given this necessary condition, the necessary and sufficient condition for the implementation of \( q \) is

\[
\nu^{t_1}(\theta_1) - \nu^{t_1}(\theta_2; \theta_1) \geq \nu^{t_2}(\theta_1; \theta_2) - \nu^{t_2}(\theta_2),
\]

which again holds if the two types are “sufficiently different.” Another intuition from this result is that for two-pass implementation, only the incentive constraints of two specific adjacent types matter. In other words, to implement a two-pass scheme, we only need to look for “gaps” between adjacent types.

For the family of strictly concave utility functions, the above notion of “gap” is not straightforward to illustrate. To simplify the intuition, we consider a special functional form where every pair of two functions from the family are an affine transformation of the other, as in the following proposition.

**Proposition 7** (Two-pass implementation with multiple strictly concave utilities). Let \( u \) be a strictly concave base utility function. Pick \( 0 \leq \alpha^T \leq \cdots \leq \alpha^1 \) and \( \beta^1 > \cdots > \beta^T = 1 \). Consider \(((N^t)_{t=1}^T, K = 2, (u^t)_{t=1}^T)\) where \( u^t = \alpha^t + \beta^t u \). Let \( q \in Q \left( (N^t)_{t=1}^T, 2 \right) \) such that every customer buys some pass and \( \nu^T(\theta_2) \geq 0 \). \( q \) is implementable if and only if

\[
\frac{\beta^{t_1}}{\beta^{t_2}} \geq \frac{\nu(\theta_1; \theta_2) - \nu(\theta_2)}{\nu(\theta_1) - \nu(\theta_2; \theta_1)},
\]

where \( \nu(\cdot; \cdot) \) is associated with \( u \). In addition, the right-hand side of (2) is strictly greater than 1, and converges to 1 as \( N^t \to \infty \) for any \( t \geq t_2 \).

The intuition of this result is that when two adjacent types are sufficiently different, the upper bound of \( p_1 - p_2 \) implied by the downward IC constraint is above the lower bound implied by the upward IC constraint, making the scheme implementable.

In summary, we have shown that when there are multiple types of utility functions, the issue with resolving the upgrade and downgrade incentives is abated yet the issue could still persist.
Specifically for the family of utility functions where each type is a positive affine transformation of the other, we show that implementing multi-pass schemes that are not implementable in the single-type case can depend on how different (with respect to the slope) each pair of adjacent types are. In other words, the ‘gaps’ between two adjacent types need to large enough for implementation.

6 Discussions

In the single-type case, the one-pass scheme maximizes the profit. In this section, we show that a multi-pass scheme can be optimal in the multi-type case. Particularly, we look at the two-type case in Section 6.2. In Section 6.3, we revisit an earlier discussion about the size of externality and whether customers actually take the externality into account for their decision making. In this part, we look at the convex case where externality is allowed to decrease to 0 as $N$ tends to infinity. In addition, we also analyze implementation with $\epsilon$-implementation, where customers do not deviate from a pass unless the deviation can improve her payoff by at least $\epsilon$.

6.1 Implementation and Externality

We now discuss in more detail the difference between the standard screening model and our model. In the standard model, every upward IC constraint holds if every local upward IC constraint holds, and every downward IC constraint hold if every local downward IC constraint holds. This constraint reduction does not hold in our model. In the standard model, the crucial step to achieve this simplification for $1 < j < K$ would be that $v(\theta_{j-1}; \theta_j) - v(\theta_j) \geq v(\theta_{j-1}; \theta_{j+1}) - v(\theta_j; \theta_{j+1})$, which does not hold in our model as $v(\theta_{j-1}; \theta_j) = v(\theta_{j-1}; \theta_{j+1})$, i.e., the utility from switching only depends on whether the new pass has a higher or lower priority.

The lack of reduction of the upward IC constraints by the local upward IC constraints makes implementation of multi-pass schemes non-trivial. In the standard model, the ID conditions imply that binding every local downward IC constraint satisfies every downward IC constraint, and the local IC constraints together imply all the IC constraints. Hence a scheme that satisfies the ID conditions in a standard setup is always implementable. However, in our model, as it is not true that every upward IC constraint holds whenever every local upward IC constraint holds, implementation of a scheme is not guaranteed with the ID conditions.

The IC constraints in our model differ from standard screening models in the existence of externality each customer’s decision imposes on others: when one customer switches to another pass, she creates congestion if the new pass has a higher priority and improves the waiting time of the new pass if the new pass has a lower priority. We can show that the existence of this type
of externality makes implementation harder than a model in which customers do not take into account this externality. We first formalize the definition of externality and specify the externality with which we discuss the relative implementability.

**Definition 2.** Fix \((N, K)\) and \(q \in Q(N, K)\). A utility function \(v : \{\theta_k\}_{k=1}^K \times \{\theta_j\}_{j=0}^K \to \mathbb{R}\)

(a) creates more downgrading externality if for every \(j\) and \(k\) with \(1 \leq j < k \leq K\), we have 
\[0 < v(\theta_j) - v(\theta_j; \theta_k) < v(\theta_k; \theta_j) - v(\theta_k),\]

(b) creates zero externality if \(v(\theta_j) = v(\theta_j; \theta_k)\) for every two passes \(\theta_j\) and \(\theta_k\).

Every \(v\) associated with strictly concave base utility function in the main text creates more downgrading externality on every scheme \(q\). In the next result, we show that a utility function that creates zero externality makes implementation easy.

**Proposition 8 (Implementation with zero externality).** Fix \((N, K)\) and \(q \in Q(N, K)\). Assume \(v\) creates zero externality. If \(v(\theta_j) \geq 0\) for every \(1 \leq j \leq K\), then the price vector \(p^*\) that \(p^*_j = v(\theta_j)\) for every \(1 \leq j \leq K\) implements \(q\).

In words, if each customer’s valuation of a pass does not change when the number of customers in that pass changes, then the park is able to implement a scheme by using the price vector that extracts the entire customer surplus. In our model, lack of implementability comes from the existence of externality. One immediate consequence of Proposition 8 is that when the utility function creates externality, the set of implementable schemes becomes weakly smaller.

**Corollary 1.** Fix \((N, K)\) and \(q \in Q(N, K)\). Let \(v\) be a utility function that creates externality and \(\phi\) a utility function that creates zero externality and \(\phi(\theta_j) = v(\theta_j)\) for every \(1 \leq j \leq K\). If \(q\) is implementable with respect to \(v\), then \(q\) is implementable with respect to \(\phi\).

In the following proposition, we show that when a utility function creates more downgrading externality, only single-pass schemes are implementable. The proposition is an immediate consequence of Lemma 3

**Proposition 9 (Implementation with more downgrading externality).** Fix \((N, K)\) and \(q \in Q(N, K)\). Assume the utility function \(v\) creates more downgrading externality and \(v(\theta_j) \geq 0\) for every \(1 \leq j \leq K\). We have \(q\) is implementable if and only if \(K = 1\).

Essentially, in the two results above, we show the existence of externality “shrinks” the set of prices that can implement a queue, making the set of implementable schemes with externality a proper subset of implementable schemes in the case without this externality.
6.2 Profits

Another difference from the standard model concerns the profit of a scheme. First note that the \( p^* \) defined in [Lemma 4] and its extensions in the multi-type case implies the optimal price of an implementable scheme. In the standard model, provided that all customers are served, more passes weakly improve the profit because the principal can extract even more surplus from those with a higher valuation, whereas in our model with only one utility type, having more passes actually weakly hurts the profit. To see this, note in the single-type case, having only one pass allows the park to extract the entire customer surplus while implementing a multi-pass scheme means that the park needs to give away surplus to customers with higher priority passes, making multi-pass scheme suboptimal with regard to profits. The key intuition for the observation that a single-pass scheme can yield a higher profit than a multi-pass scheme is that in the standard model, a single-pass scheme would not change the valuation of customers in the last pass, while in our model, having all customers in one single pass actually increases the utility of those who used to be in the lowest-priority pass since now they have an opportunity to be at the front of queue.

For the multi-type case, however, the single-pass scheme does not always dominate a multi-pass one. Indeed, for the two-type strictly concave utility case we just discussed, we can show that for some values of \( \beta^l \), the two-pass scheme gives a higher profit than does a single-pass scheme, as we show in the following proposition about the two-type case. We focus on all-serving schemes, where every customer of every type buys some pass.

**Proposition 10** (Profitability of two-pass schemes). Consider the two-type case and fix \( N^h \) and \( N^l \). Let \( u^h = u \), where \( u \) is strictly concave, and \( u^l = \beta^l u \), where \( \beta^l \in (0, 1) \). There exists \( \bar{\beta} \in (0, 1) \) such that for \( \beta^l \leq \bar{\beta} \), \( K = 2 \) is implementable, and the profit from the optimal all-serving two-pass scheme is higher than that of the optimal one-pass all-serving scheme.

6.3 Implementation in Large Queues

The implementation results we have depend on the externality customers impose on each other. A natural question is whether the difficulty with implementing a multi-pass scheme can be reduced when the magnitude of externality gets small. In the next result, we allow the number of customers to grow and show that when \( u \) is convex, under some conditions, implementing many passes is possible.

**Proposition 11** (Implementation with convex utility, fixed \( K \) and large \( N \)). Fix \( K \) and a strictly decreasing sequence \( (u_n)_{n=1}^{\infty} \). If \( u \) is convex with \( \lim_{n \to \infty} \frac{u_n}{n} = 0 \), then there exist \( M \) and \( u : \mathbb{N} \to \mathbb{R} \) such that for every \( N \geq M \), \( (N, K, u) \) is implementable with \( u_0 = u(N) \), i.e., there exists an
implementable scheme for \((N, K, u)\) when the utility of outside option is \(u(N)\) which can vary with \(N\).

The condition \(\lim_n \frac{u_n}{n} = 0\) does allow an unbounded utility function. For example, \(u_n = -\log n\) is unbounded and satisfies the condition. The condition on \(u\) does imply a slow rate of decrease, excluding the linear and concave base utility functions.

Even without convexity, we can still show that the effect of externality can become negligible when there are many customers. We formalize this idea by using the concept of \(\varepsilon\)-implementation, where each constraint of the park’s optimization problem is allowed to be relaxed by arbitrary and fixed \(\varepsilon > 0\). Specifically, \(\text{IR}_k\) in \(\varepsilon\)-implementation becomes

\[
v(\theta_k) - p_k + \varepsilon \geq u_0, \quad (\varepsilon \text{IR}_k)
\]

and \(\text{IC}_{jk}\) in \(\varepsilon\)-implementation becomes

\[
v(\theta_j) - p_j + \varepsilon \geq v(\theta_k; \theta_j) - p_k, \quad (\varepsilon \text{IC}_{jk})
\]

and the definition of implementation is defined similarly to the exact-implementation case.

**Definition 3** (\(\varepsilon\)-implementation). Fix \(N \geq K\).

(a) A scheme \(q \in Q(N, K)\) is \(\varepsilon\)-implementable if for each \(\varepsilon > 0\), there exists some price vector 
\(p = (p_1, \ldots, p_K)\) such that \(\varepsilon \text{IR}_k\) and \(\varepsilon \text{IC}_{jk}\) hold for every two passes \(\theta_j\) and \(\theta_k\).

(b) \((N, K, u)\) is \(\varepsilon\)-implementable if there exists an \(\varepsilon\)-implementable scheme with \(N\) customers and \(K\) passes.

Here \(u_0\) may be non-zero because when we are dealing with serving a large customer pool, we may need the utility from the outside option to be low enough to satisfy the IR constraints. For \(\varepsilon > 0\), we have an asymptotic result, i.e., \(N\) is allowed to vary while \(K\) is fixed. We show that that the perfect extract pricing, ie, \(p_k = v(\theta_k)\) for every \(k\), is \(\varepsilon\)-implementable for some schemes when \(N\) is allowed to grow.

**Theorem 4** (\(\varepsilon\)-implementation). Fix \(K, \varepsilon > 0,\) and a strictly decreasing sequence \((u_n)_{n=1}^{\infty}\). If 
\[\lim_{n \to \infty} \frac{u_n}{n} = 0,\]
then there exist \(M\) and \(u : \mathbb{N} \to \mathbb{R}\) such that \(N > M\) implies \((N, K, u)\) is \(\varepsilon\)-implementable with \(u_0 = u(N)\). Specifically, an implementable scheme is the scheme in which customers in each pass have zero surplus, i.e., 
\(p_j = v(\theta_k) - u_0\) for every pass \(\theta_j\).

The proof for the existence of \(M\) in the above statement is constructive: we need \(M\) to be large enough such that the change in utility when a customer in \(\theta_j\) switches to \(\theta_k\) for \(\theta_j < \theta_k\) is smaller.
than $\epsilon$. The cutoff $M$ we construct in the proof is sufficient for $\epsilon$-implementation, but it is not necessary. Particularly, given the assumptions, we can have however many customers in the first pass, as long as we make sure that there are much more customers in later passes.

7 Conclusions

To summarize, we show the difficulty with 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 directly from the conflict between incentivizing customers from upgrading and from downgrading. We show that when utility function is strictly concave, which is commonly assumed in the mechanism design literature, implementing a multi-pass scheme cannot be incentive compatible if there is only one type. We show that such incentive conflicts in our model can persist even when there are multiple types of base utility functions.

References


A Auxiliary Results with Proofs

Lemma 5 (Sufficient condition for implementation). Given $(N, K, u)$ and a scheme $q = (q_1, \ldots, q_K) \in Q(N, K)$ such that $\sum_{k=1}^{K} q_k = N$, assume $\nu(\theta_K) \geq 0$ and all ID conditions hold. If for every $l \leq K - 2$, $\nu(\theta_l; \theta_{l+1}) - \nu(\theta_{l+1}; \theta_{l+2}) \leq \nu(\theta_l) - \nu(\theta_{l+1}; \theta_l)$, then the scheme is implementable.

Proof. From Lemma 4, we only need to check whether setting $p_j - p_{j+1} = \nu(\theta_j; \theta_j) - \nu(\theta_{j+1}; \theta_j)$ for every $j$ will respect all the upward constraints. For $j < k$, IC$_{kj}$ implies $p_j - p_k \geq \nu(\theta_j; \theta_k) - \nu(\theta_k)$. Hence it is sufficient to show

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

which is equivalent to

$$\sum_{l=j}^{k-1} \nu(\theta_l; \theta_k) - \nu(\theta_{l+1}; \theta_k) \leq \sum_{l=j}^{k-1} \nu(\theta_l) - \nu(\theta_{l+1}; \theta_l).$$

Since $\nu(\theta_{k-1}; \theta_k) - \nu(\theta_k) \leq \nu(\theta_{k-1}) - \nu(\theta_k; \theta_{k-1})$ holds because of the ID conditions, it is sufficient to check that

$$\sum_{l=j}^{k-2} \nu(\theta_l; \theta_k) - \nu(\theta_{l+1}; \theta_k) \leq \sum_{l=j}^{k-2} \nu(\theta_l) - \nu(\theta_{l+1}; \theta_l).$$

Since $\nu(\theta_l; \theta_k) = \nu(\theta_l; \theta_{l+1})$ for every $l < k$, the above inequality is equivalent to

$$\sum_{l=j}^{k-2} \nu(\theta_l; \theta_{l+1}) - \nu(\theta_{l+1}; \theta_{l+2}) \leq \sum_{l=j}^{k-2} \nu(\theta_l; \theta_l) - \nu(\theta_{l+1}; \theta_l),$$

which holds if $\nu(\theta_l; \theta_{l+1}) - \nu(\theta_{l+1}; \theta_{l+2}) \leq \nu(\theta_l; \theta_l) - \nu(\theta_{l+1}; \theta_l)$ holds for every $l$ with $1 \leq l \leq K - 2$. \hfill $\square$

Lemma 6. If $u_n^b - u_{n+1}^b \geq u_n^l - u_{n+1}^l$ for every $n$, then given $j < k$ and $(m, r) \in \{(j, j), (k, k), (j, k)\}$, we have $u^b(\theta_j; \theta_m) - u^b(\theta_k; \theta_r) \geq u^l(\theta_j; \theta_m) - u^l(\theta_k; \theta_r)$.

Proof. To economize on space, we omit the argument of $Q_k(q)$ when there is no ambiguity.\footnote{We use the same omission in later proofs as well.}

When $m = r = j$, the inequality is equivalent to

$$\frac{\sum_{n=Q_j+1}^{Q_k} u_n^b}{q_j} \geq \frac{\sum_{n=Q_k}^{Q_k} u_n^h}{q_k + 1} \geq \frac{\sum_{n=Q_j+1}^{Q_k} u_n^l}{q_j} \geq \frac{\sum_{n=Q_k}^{Q_k} u_n^l}{q_k + 1},$$

$$\frac{\sum_{n=Q_j+1}^{Q_k} u_n^b}{q_j} \geq \frac{\sum_{n=Q_k}^{Q_k} u_n^h}{q_k + 1} \geq \frac{\sum_{n=Q_j+1}^{Q_k} u_n^l}{q_j} \geq \frac{\sum_{n=Q_k}^{Q_k} u_n^l}{q_k + 1},$$
which is equivalent to
\[ \frac{\sum_{n=Q_j-1+1}^{Q_j} u_n^h - u_n^l}{q_j} \geq \frac{\sum_{n=Q_k-1}^{Q_k} u_n^h - u_n^l}{q_k+1}, \]
which holds if \( u_n^h - u_n^l \) is non-increasing in \( n \). Similarly, for \( m = r = k \), the inequality is equivalent to
\[ \frac{\sum_{n=Q_j-1+1}^{Q_j+1} u_n^h - u_n^l}{q_j+1} \geq \frac{\sum_{n=Q_k-1+1}^{Q_k} u_n^h - u_n^l}{q_k}, \]
which holds for non-increasing \( u^h - u^l \). When \( m = j \) and \( r = k \), the inequality is equivalent to
\[ \frac{\sum_{Q_j-1}^{Q_j} u_n^h - u_n^l}{q_j} \geq \frac{\sum_{Q_k}^{Q_k} u_n^h - u_n^l}{q_k}, \]
which again holds if \( u_n^h - u_n^l \) is non-increasing in \( n \).

\( \square \)

**Lemma 7** (Effective IC constraint). For two passes \( \theta_j \) and \( \theta_k \), if \( \theta_j < \theta_k \), then \( I^l_{jk} \) implies \( I^h_{jk} \); if \( \theta_k < \theta_j \) then \( I^h_{jk} \) implies \( I^l_{jk} \).

**Proof.** We need Lemma 6 in Appendix A. Let \( k > j \). Then \( I^l_{jk} \) implies \( I^h_{jk} \) if and only if \( v^h(\theta_j) - v^h(\theta_k; \theta_j) \geq v^l(\theta_j) - v^l(\theta_k; \theta_j) \), which holds from Lemma 6. The proof for \( k < j \) is similar. \( \square \)

**Lemma 8** (IC Reduction with two utilities). Fix \( ((N^h, N^l), K, (u^h, u^l)) \) with \( K \geq 2 \). Given a scheme \( q \) with a price vector \( p \). Pick \( j \) and \( m \) with \( 1 \leq j < m \leq K \) and assume \( (p, q) \) satisfies \( I^l_{j,k+1} \) for every \( k \) with \( j \leq k < m \). If \( q^l_j = 0 \) or \( q^l_k > 0 \) for every \( j < k < m \), then \( (p, q) \) satisfies \( I^l_{j,m} \).

**Proof.** Assume \( q^l_j = 0 \), then \( I^l_{j,j+1} = I^h_{j,j+1} \). Assume the induction hypothesis that \( (p, q) \) satisfies \( I^l_{jk} \) holds for some \( j < k < m \). It suffices to show that \( (p, q) \) satisfies \( I^l_{j,k+1} \), i.e., we need show \( p_j - p_{k+1} \leq v^h(\theta_j) - v^h(\theta_{k+1}; \theta_j) \). Since \( (p, q) \) satisfies all local IC constraints between \( \theta_j \) and \( \theta_m \), we have \( p_r - p_{r+1} \leq v^r(\theta_r) - v^r(\theta_{r+1}; \theta_r) \), where \( j \leq r < m \). We thus have
\[
\begin{align*}
p_j - p_{k+1} & = \sum_{r=j}^{k} p_r - p_{r+1} \leq \sum_{r=j}^{k} v^r(\theta_r) - v^r(\theta_{r+1}; \theta_r) \\
& \leq \sum_{r=j}^{k} v^h(\theta_r) - v^h(\theta_{r+1}; \theta_r) \\
& \leq \sum_{r=j}^{k-1} v^h(\theta_r) - v^h(\theta_{r+1}) + \sum_{r=j}^{k} v(\theta_k) - v(\theta_{k+1}; \theta_k) \\
& = v^h(\theta_j) - v^h(\theta_{k+1}; \theta_k) = v^h(\theta_j) - v^h(\theta_{k+1}; \theta_j),
\end{align*}
\]
where the second line comes from [Lemma 6]. Hence \((p, q)\) satisfies IC\(_{jm}\). Now assume instead \(q^l_r > 0\) for all \(r\) with \(j < r < m\) and hence IC\(_{rs}\) = IC\(_{rs}^l\) for every \(r\) with \(j < r < m\). Assume the induction hypothesis that \((p, q)\) satisfies IC\(_{j,k}\) for some \(j < k < m\). To show \((p, q)\) satisfies IC\(_{j,k+1}\), note that we have

\[
\nu^j(\theta_j) - p_j \geq \nu^j(\theta_k; \theta_j) - p_k \\
\geq \nu^j(\theta_k; \theta_j) - p_k \\
\geq \nu^j(\theta_k) - p_k \\
\geq \nu^j(\theta_{k+1}; \theta_k) = \nu^j(\theta_{k+1}; \theta_k) - p_{k+1},
\]

and hence \((p, q)\) satisfies IC\(_{j,k+1}\). \(\square\)

**Lemma 9** (IR Reduction with two utilities). Fix \(((N^h, N^l), K, (u^h, u^l))\) with \(K \geq 2, q \in Q((N^h, N^l), K)\) and a price vector \(p\). Assume for every \(j\) and \(k\) with \(1 \leq j < k \leq K\), \((q, p)\) satisfies IC\(_{jk}\) and IR\(_k\). If \(q^l_j = 0\) or \(q^l_k > 0\), then \((q, p)\) satisfies IR\(_j\).

**Proof.** If \(q^l_j = 0\), then IR\(_j\) = IR\(_j^h\) and IC\(_{jk}\) = IC\(_{jk}^h\). Let \(t \in \{l, h\}\) such that IR\(_k\) = IR\(_k^t\). Then we have

\[
\nu^h(\theta_j) - p_j \geq \nu^h(\theta_k; \theta_j) - p_k \geq \nu^h(\theta_k) - p_k \geq 0,
\]

and thus IR\(_j\) holds. If \(q^l_k > 0\), then IR\(_k\) = IR\(_k^l\). IC\(_{jk}\) implies

\[
\nu^l(\theta_j) - p_j \geq \nu^l(\theta_k; \theta_j) - p_k \geq \nu^l(\theta_k; \theta_j) - p_k \geq 0,
\]

and hence IR\(_j\) holds. \(\square\)

**Lemma 10** (Two-type implementation conditions). Fix \(((N^h, N^l), K, (u^h, u^l))\). Let \(q \in Q((N^h, N^l), K)\) in which every customer buys some pass. Assume for every \(\theta_1 \leq \theta_j < \theta_K\), \(q^l_j > 0\) implies \(q^l_{j+1} > 0\). Let \(p^* = (p^*_1, \ldots, p^*_K)\) such that \(p^*_K = \nu^l(\theta_K)\) and \(p^*_j - p^*_j = \nu^j(\theta_j) - \nu^j(\theta_{j+1}; \theta_j)\). \(q\) is implementable if and only if \((p^*, q)\) satisfies every upward IC constraint and \(\nu^l(\theta_K) \geq 0\).

**Proof.** Since \(N^l > 0\) and \(q^l_k > 0\) implies \(q^l_{k+1} > 0\) for every \(k\), we have \(q^l_k > 0\) and hence IR\(_K\) = IR\(_K^l\). Regardless of the implementability of \(q\), \((p^*, q)\) satisfies every local downward IC constraint. Pick \(j\) and \(k\) with \(1 \leq j < k \leq K\). If \(q^l_j = 0\), then [Lemma 8] implies IC\(_{jk}\) holds. If \(q^l_j > 0\), then \(q^l_r > 0\) for every \(r\) with \(j \leq r < k\) by assumption, and hence again by [Lemma 8] IC\(_{jk}\) holds. Thus \((p^*, q)\) satisfies every downward IC constraint. Since in addition IR\(_K^l\) holds, by [Lemma 9] all the IR constraints also hold. Thanks to the notations for the two-type case, the rest of the proof is exactly the same with that of [Lemma 4]. We have shown that \(q\) is implementable if and only if \((p^*, q)\) satisfies every upward IC constraint. \(\square\)
Lemma 11 (Generalization of Lemma 7). Fix \( ((N^i)_{i=1}^T, K, (u^i)_{i=1}^T) \). Fix a scheme \( q = (q_1, \ldots, q_K) \) with a price vector \( p \), and \( j \) and \( k \) with \( 1 \leq j \leq K \). Let \( \tau = \{ t \in \mathbb{N} : 1 \leq t \leq T, q^j_t > 0 \} \). For \( \theta_j < \theta_k \), we have \( IC_{jk} = IC_{jk}^{\max, \tau} \), for \( \theta_k < \theta_j \), we have \( IC_{jk} = IC_{jk}^{\min, \tau} \).

Proof. The proof is similar to that of Lemma 7 and we omit the proof.

Lemma 12 (Generalization of Lemma 8). Fix \( ((N^i)_{i=1}^T, K, (u^i)_{i=1}^T) \) and a price vector \( p \). Given \( \theta_j < \theta_t \), assume \( IC_{k,k+1} \) holds for every \( \theta_j \leq \theta_k < \theta_t \). If \( t_j \leq t_k \) for every \( \theta_j \leq \theta_k < \theta_t \), then \( IC_{jl} \) holds.

Proof. We need to show \( p_j - p_l \leq v^j_i(\theta_j) - v^j_i(\theta_l; \theta_j) \). From Lemma 6 for every \( j \) and \( k \) with \( 1 \leq j \leq k \leq K \), we have \( v^i_j(\theta_k) - v^i_j(\theta_{k+1}; \theta_k) \geq v^i_{k}(\theta_k) - v^i_{k}(\theta_{k+1}; \theta_k) \). From the set of local downward IC constraints, we have

\[
p_j - p_l = \sum_{k=j}^{l-1} p_k - p_{k+1} \leq \sum_{k=j}^{l-1} v^i_{k}(\theta_k) - v^i_{k}(\theta_{k+1}; \theta_k) \leq \sum_{k=j}^{l-1} v^i_{j}(\theta_k) - v^i_{j}(\theta_{k+1}; \theta_k) \leq \sum_{k=j}^{l-2} v^i_{j}(\theta_k) - v^i_{j}(\theta_{k+1}) + v^i_{j}(\theta_{l-1}) - v^i_{j}(\theta_{l}; \theta_{l-1}) = v^i_{j}(\theta_j) - v^i_{j}(\theta_l; \theta_j),
\]

and hence \( IC_{jl} \) holds.

Lemma 13 (Generalization of Lemma 9). Fix \( ((N^i)_{i=1}^T, K, (u^i)_{i=1}^T) \) with \( K \geq 2 \). Fix \( q \in Q((N^i)_{i=1}^T, K) \) and a price vector \( p \). Fix \( j \) and \( k \) in \( \{1, \ldots, K\} \) and assume \( t_j \leq t_k \). If \( IC_{jk} \) holds and \( IR_k \) holds, then \( IR_j \) holds.

Proof. The proof is similar to that of Lemma 9 and we omit the proof.

Lemma 14 (Implementation conditions in multi-type case). Fix \( ((N^i)_{i=1}^T, K, (u^i)_{i=1}^T) \). Let \( q \in Q((N^i)_{i=1}^T, K) \) be a scheme such that \( t_j \leq t_k \) for every \( j \) and \( k \) with \( 1 \leq j < k \leq K \). Assume every customer buys some pass. Let \( p^* = (p^*_1, \ldots, p^*_K) \) such that \( p^*_k = v^T(\theta_K) \) and \( p^*_j - p^*_{j+1} = v^i_j(\theta_j) - v^i_j(\theta_{j+1}; \theta_j) \). \( q \) is implementable if and only if \( (p^*, q) \) satisfies every upward IC constraint and \( v^T(\theta_K) \geq 0 \).

Proof. By construction, \( (p^*, q) \) satisfies every local downward IC constraint. By assumptions, conditions for Lemma 12 and Lemma 13 are met and hence \( (p^*, q) \) satisfies every downward IC constraint and \( IR_j \) for every \( j \) with \( 1 \leq j \leq K \). Hence if \( (p^*, q) \) satisfies every upward IC constraint and \( v^T(\theta_K) \geq 0 \), \( q \) is implementable.
Now assume $q$ is implementable and let $p$ implement $q$. Since every customer in $q$ buys, $t^K = T$. Implementability of $q$ therefore implies $\nu^T(\theta_K) \geq 0$. Pick $j$ and $k$ with $1 \leq j < k \leq K$. Since $p$ implements $q$, $(p, q)$ satisfies $\text{IC}_{k,j}$ and hence

$$\nu^{t_k}(\theta_j; \theta_k) - \nu^{t_k}(\theta_k) \leq p_j - p_k.$$ 

In addition, for $j \leq l < k$, we have

$$\sum_{l=j}^{k-1} p_l - p_{l+1} \leq \sum_{l=j}^{k-1} \nu^{l_j}(\theta_l) - \nu^{l+1_j}(\theta_l) = \sum_{l=j}^{k-1} p_l^* - p_{l+1}^*.$$ 

Combining these inequalities, we get

$$p_j^* - p_k^* = \sum_{l=j}^{k-1} p_l^* - p_{l+1}^* \geq \sum_{l=j}^{k-1} p_l - p_{l+1} = p_j - p_k \geq \nu^{t_k}(\theta_j; \theta_k) - \nu^{t_k}(\theta_k),$$

and hence $(p^*, q)$ satisfies every upward IC constraint. The proof is complete.

**Proposition 12.** Consider $((N^t)_{t=1}^T, K, (u^t)_{t=1}^T)$ where $T \geq 2$, $K \geq 2$, and either $u^t$ is linear for all $t$ or $u^t$ is strictly concave for all $t$. Let $q \in Q((N^t)_{t=1}^T, K)$ be implementable. If there exists some type $\tau < T$ such that $q^\tau_j > 0$ for some $\theta_j < \theta_K$, then $\sum_{k=1}^K q^t_k = N^t$.

**Proof.** Since $t_j = \max\{t : q^t_j > 0\}$, $\tau \leq t_j$, $\nu^\tau(\theta_j) - p_j \geq \nu^{t_j}(\theta_j) - p_j$. In addition, since $\tau < T$, we have $\nu^\tau(\theta_K) > \nu^T(\theta_K)$. If $u^t$ is linear or strictly concave, we have

$$\nu^\tau(\theta_j; \theta_K) - p_j^* \geq \nu^{t_j}(\theta_j; \theta_K) - p_j^* = \nu^{t_j}(\theta_j) - p_j^* - \left[\nu^{t_j}(\theta_j) - \nu^{t_j}(\theta_j; \theta_K)\right] \geq \nu^{t_j}(\theta_K; \theta_j) - p_k^* - \left[\nu^{t_j}(\theta_j) - \nu^{t_j}(\theta_j; \theta_K)\right] = \nu^{t_j}(\theta_K; \theta_j) - \nu^T(\theta_K) - \left[\nu^{t_j}(\theta_j) - \nu^{t_j}(\theta_j; \theta_K)\right] \geq \nu^{t_j}(\theta_K; \theta_j) - \nu^{t_j}(\theta_K) - \left[\nu^{t_j}(\theta_j) - \nu^{t_j}(\theta_j; \theta_K)\right] \geq 0.$$

If $t_j < T$, then the second-to-last inequality would be strict. If $t_j = T$, then $\tau < t_j$ and the first inequality would be strict. Hence if $\sum_{k=1}^K q^t_k < N^t$, $\text{IC}_{0j}^t$ would not hold, a contradiction. \[\square\]

**B Proofs of Results in Main Text**

**B.1 Proof of Lemma 2**

**Proof.** To prove (a), for $j$ and $k$ with $1 \leq j < k \leq K$, assume $(p, q)$ satisfies $\text{IC}_{l,l+1}$ for every $l$ with $j \leq l < K$. If $k = j + 1$, the statement is trivial. Assume $j + 1 < k$ instead. We want to show
that $IC_{j,j+n}$ for every $n \in \mathbb{N}$ such that $j + n \leq k$. To show this, we use strong induction on $n$ and assume that $(p, q)$ satisfies $IC_{j,j+n_i}$ for every $n_i$ with $1 \leq n_i \leq n - 1$. We then have

$$v(\theta_j) - p_j \geq v(\theta_{j+n-1}; \theta_j) - p_{j+n-1}$$

$$\geq v(\theta_{j+n-1}) - p_{j+n-1}$$

$$\geq v(\theta_{j+n}; \theta_{j+n-1}) - p_{j+n} = v(\theta_{j+n}; \theta_j) - p_{j+n},$$

where the first inequality comes from the induction hypothesis, the second inequality comes from the assumption, the second inequality and the the last equality come from the property of $v(\cdot, \cdot)$. Hence the proof for (a) is complete.

To prove (b), assume $(p, q)$ satisfies $IC_{jk}$ and $IR_k$. From $IC_{jk}$ and $IR_k$, we have

$$v(\theta_j) - p_j \geq v(\theta_j; \theta_k) - p_k \geq v(\theta_k) \geq 0,$$

and hence $(p, q)$ satisfies $IR_j$ and the proof for (b) is complete. (c) is immediate clear when the inequalities from $IC_{jk}$ and $IC_{kj}$ are combined. \hfill \Box

### B.2 Proof of Proposition 1

**Proof.** We are to prove the result by contradiction. Since $N > 2$ is arbitrary and $q$ is implementable for $(N, K, u)$ implies the same scheme is implementable for $(\sum_{k=1}^{K} q_k, K, u)$, it is without loss of generality to assume $\sum_{k=1}^{K} q_k = N$. Conditional on this simplification, we further argue that it is necessary that $q_k > 1$ for some pass $k$. If not, then we can use the same reasoning for proving Proposition 3 to show that the scheme would not be implementable.

We show that when $u$ is concave and non-linear, the ID conditions would be violated. To arrive at a contradiction, assume $q_k > 1$ for some pass $k$ and $q$ is implementable. Since $u$ is concave and non-linear, we can find some $n$ such that $u_n - u_{n+1} < u_{n+1} - u_{n+2}$. Find $j$ such that $\theta_j$ and $\theta_{j+1}$ include a non-linear part of the utility function. For $n = 0, \ldots, q_j$, define $x_n = u_{Q_j-n+1}$. Similarly, for $n = 0, \ldots, q_{j+1}$, define $y_n = u_{Q_{j+1}+n}$. By the construction, we have $y_{q_{k+1}+1} < \cdots < y_0 \leq x_0 < x_1 < \cdots < x_{q_j}$. By the concavity of $u$, we have $x_1 - x_0 \leq y_0 - y_1$. ID$_{jk}$ is equivalent to

$$\frac{\sum_{n=1}^{q_j} x_n}{q_j} - \frac{\sum_{n=0}^{q_j+1} y_n}{q_j+1} \geq \frac{\sum_{n=0}^{q_j} x_n}{q_j+1} = \frac{\sum_{n=1}^{q_j+1} y_n}{q_j(q_j+1)} \geq \frac{\sum_{n=1}^{q_{j+1}} (y_0 - y_n)}{q_{j+1}(q_{j+1}+1)}.$$

By the assumption of $u$, $\sum_{n=1}^{q_j} (x_n - x_0) \leq \frac{q_j(q_j+1)(x_1-x_0)}{2}$. Similarly, $\sum_{n=1}^{q_{j+1}} (y_0 - y_n) \geq \frac{q_{j+1}(q_{j+1}+1)(y_0-y_1)}{2}$. At least one of the two inequalities is strict, since the two passes have at least one non-linear part,
from which we have
\[
\sum_{n=1}^{q_j} (x_n - x_0) \over q_j(q_j + 1) < \sum_{n=1}^{q_k} (y_n - y_0) \over q_k(q_k + 1),
\]
violating \(ID_{i,j+1}\). Hence the pass is not implementable. \(\square\)

### B.3 Proof of Proposition 2

**Proof.** Let \(q \in Q(N, K)\). Since the base utility function is convex and there are only two passes, both \(IC_{12}\) and \(IC_{21}\) hold. If \(q_1 + q_2 = N\), then the scheme is implementable.

Now assume \(q_1 + q_2 < N\). Consider \(p_2 = v(\theta_2)\) and \(p_1 = p_2 + v(\theta_1) - v(\theta_2; \theta_1)\). We show that \((p_1, p_2)\) implements \(q\) for \((N, K, u)\). It is straightforward to check that with \((p_1, p_2), IC_{12}, IC_{21}, IR_1\) and \(IR_2\) all hold. Since \(p_2 = v(\theta_2)\) and \(v(\theta_2) > v(\theta_2; \theta_0)\), we see \(IC_{02}\) holds. It remains to show that \(IC_{01}\) holds. We have
\[
v(\theta_1; \theta_2) - p_1 = v(\theta_1; \theta_2) - v(\theta_2) - v(\theta_1) + v(\theta_2; \theta_1).
\]
Since \(u\) is convex, we have \(v(\theta_2; \theta_1) - v(\theta_2) \geq v(\theta_1) - v(\theta_1; \theta_2)\), and hence \(v(\theta_1; \theta_2) - p_1 \leq 0\) and \(IC_{01}\) indeed holds. Therefore, \(q\) is implementable. \(\square\)

### B.4 Proof of Lemma 4

**Proof.** By construction, \((p^*, q)\) satisfies every local downward IC constraint. From [Lemma 2], \(p^*\) satisfies every downward IC constraint. Since \((p^*, q)\) satisfies \(IR_K\) and every downward IC constraint, from [Lemma 2], \(IR_k\) holds for every \(k\) with \(1 \leq k \leq K\). It remains to show \((p^*, q)\) satisfies every upward IC constraint. Hence if \((p^*, q)\) satisfies every upward IC constraint for every \(j\) and \(k\) with \(1 \leq j < k \leq K\), \(p^*\) implements \(q\) and \(q\) is implementable.

Now assume \(q\) is implementable and let \(p\) be a price vector that implements \(q\). \(v(\theta_k) \geq 0\) since \(v(\theta_k) \geq v(\theta_k) - p_K \geq 0\) by \(IR_K\). Pick \(\theta_j < \theta_k\) and we consider \(IC_{kj}\), which implies \(p_j - p_k \geq v(\theta_j; \theta_k) - v(\theta_k)\). The local downward IC constraints between \(\theta_j\) and \(\theta_k\) implies \(p_l - p_{l+1} \leq v(\theta_l) - v(\theta_{l+1}; \theta_l)\) for \(j \leq l < k\). Adding these inequalities together to get
\[
v(\theta_j; \theta_k) - v(\theta_k) \leq p_j - p_k = \sum_{l=j}^{k-1} p_l - p_{l+1} \leq \sum_{l=j}^{k-1} v(\theta_l) - v(\theta_{l+1}; \theta_l) = p^*_j - p^*_k,
\]
and hence \((p^*, q)\) satisfies \(IC_{kj}\) and hence all the upward IC constraints. \(\square\)
B.5 Proof of Proposition 3

Proof. Since we require each pass to have at least one pass, the $Q(N, K)$ is a singleton when $N = K$. We show that some ID conditions would be violated. Assume instead that all ID conditions hold. Set $p = p^*$. By Lemma 4, we have $p_2 - p_3 = v(\theta_2) - v(\theta_3; \theta_2) = \frac{u_2 - u_3}{2}$ and $p_1 - p_2 = v(\theta_1) - v(\theta_2; \theta_1) = \frac{u_1 - u_2}{2}$. If the scheme is implementable, then IC$_{31}$ needs to hold, which is equivalent to

$$p_1 - p_3 \geq v(\theta_1; \theta_3) - v(\theta_3) \iff u_2 \leq u_3,$$

which contradicts the strict monotonicity of $u$. Hence the scheme is not implementable. \qed

B.6 Proof of Theorem 1

Proof. If $K \leq 2$ and there exists $q \in Q(N, K)$ such that $v(\theta_K) \geq u_0$, then Proposition 2 implies $(N, K, u)$ is implementable since $u$ is weakly convex.

Consider an arbitrary scheme $(q_1, \ldots, q_K) \in Q(N, K)$ for some $K \geq 3$. It suffices to prove the case for $\sum_{k=1}^{K} q_k = N$. Set $p = p^*$. From Lemma 4 we have $p_1 - p_3 = v(\theta_1) - v(\theta_2; \theta_1) + v(\theta_2) - v(\theta_3; \theta_2)$. From IC$_{31}$, we have $p_1 - p_3 \geq v(\theta_1; \theta_3) - v(\theta_3)$. Hence for the scheme to be implementable, it is necessary that

$$v(\theta_1) - v(\theta_2; \theta_1) + v(\theta_2) - v(\theta_3; \theta_2) \geq v(\theta_1; \theta_3) - v(\theta_3).$$

However, we show that when $u$ is linear, the above inequality does not hold. Define $d = u_1 - u_2$. Then we have

$$v(\theta_1) - v(\theta_1; \theta_3) = \frac{\sum_{n=1}^{Q_1} u_n}{q_1} - \frac{\sum_{n=1}^{Q_1+1} u_n}{q_1 + 1} = \frac{\sum_{n=1}^{Q_1} (u_n - u_{Q_1+1})}{q_1(q_1 + 1)} = \frac{d}{2}.$$ 

Similarly, we can get $v(\theta_2; \theta_1) - v(\theta_2) = v(\theta_3; \theta_2) - v(\theta_3) = \frac{d}{2}$, and thus

$$v(\theta_1) - v(\theta_1; \theta_3) - [v(\theta_2; \theta_1) - v(\theta_2)] - [v(\theta_3; \theta_2) - v(\theta_3)] = -\frac{d}{2} < 0,$$

which violates the the inequality. Hence no scheme is implementable. \qed

B.7 Proof of Proposition 4

Proof. Suppose for some $j$ and $k$ with $1 \leq j < k \leq K$ we have $q^j_1 > 0$ and $q^j_k > 0$. From Lemma 7 we have IC$_{jk} = IC^j_{jk}$ and IC$_{kj} = IC^h_{kj}$. Note that we have

$$v^j(\theta_j) - v^j(\theta_k; \theta_j) \leq v^j(\theta_j; \theta_k) - v^j(\theta_k) \leq v^h(\theta_j; \theta_k) - v^h(\theta_k),$$

30
where the first inequality comes from the proof of Proposition 1 on the concavity of \( v^l \) and the second inequality comes from Lemma 6. If \( v^l \) is strictly concave, then the first inequality is strict; if \( v^h - v^l \) is strictly decreasing, the second inequality is strict. Another set of inequalities we have derived are

\[
v^l(\theta_j) - v^l(\theta_k; \theta_j) \leq v^h(\theta_j) - v^h(\theta_k; \theta_j) \leq v^h(\theta_j; \theta_k) - v^h(\theta_k),
\]

where the first inequality comes from Lemma 6 and the second inequality comes from the proof of Proposition 1 on the concavity of \( v^h \). If \( v^h \) is strictly concave, the second inequality is strict. Hence if \( v^h \) or \( v^l \) is strictly concave or \( v^h - v^l \) is strictly decreasing, \( v^l(\theta_j) - v^l(\theta_k; \theta_j) < v^h(\theta_j; \theta_k) - v^h(\theta_k) \), a contradiction. Hence \( q^j_l > 0 \) implies \( q^h = 0 \).

**B.8 Proof of Theorem 2**

**Proof.** From Proposition 4 and Theorem 1, when \( K = 4 \), it is necessary that \( q^l_1 = q^l_2 = 0 \) and \( q^h_3 = q^h_4 = 0 \). From Lemma 10, it suffices to check whether \( p^* \) implements \( q \). Since \( v^l(\theta_K) \geq 0 \), IR^l \( K \) holds. It remains to show that \( (p^*, q) \) satisfies every upward IC constraint. Begin with IC41, which implies

\[
p_1 - p_4 \geq v^l(\theta_1) - v^l(\theta_4) - \frac{d}{2}.
\]

From \( p^* \), we have

\[
p^*_1 - p^*_4 = v^h(\theta_1) - v^h(\theta_2) + v^h(\theta_2) - v^h(\theta_3) + v^l(\theta_3) - v^l(\theta_4) - \beta d - \frac{d}{2},
\]

and hence \( (p^*, q) \) satisfies IC41 if and only if

\[
v^h(\theta_1) - v^l(\theta_1) - \left[ v^h(\theta_3) - v^l(\theta_3) \right] - \beta d \geq 0,
\]

from which we get

\[
\beta^h \geq \frac{q_1 + 2q_2 + q_3}{q_1 + 2q_2 + q_3 - 2} > 1.
\]

Similarly, for \( (p^*, q) \) to satisfy IC42, we need \( \beta^h \geq \frac{q_1 + q_4}{q_1 + q_3 - 1} > 1 \). Lastly, for IC31, we have \( \beta^h \geq 1 \). Hence there exists \( b \) such that \( p^* \) implements \( q \) if and only if \( \beta^h \geq b \). □

**B.9 Proof of Proposition 5**

**Proof.** To see the uniqueness of \( q \), note that \( q \) is the only scheme in which each type is in one pass. For every other different pass, at least one type has customers in both passes, meaning that both the upward and downward IC constraints are active for this type. Since \( u \) is strictly concave, by Proposition 1 such a pass is not implementable.
By Lemma 7, \( q \) is implementable if and only if
\[
\nu(\theta_1) - \nu(\theta_2; \theta_1) \geq \beta^1 [\nu(\theta_1; \theta_2) - \nu(\theta_2)],
\]
which completes the proof. \( \square \)

**B.10 Proof of Proposition 6**

*Proof.* Assume \( t_j > t_k \) for some \( j \) and \( k \) with \( 1 \leq j < k \leq K \), i.e., there exist some \( \tau_1 \) and \( \tau_2 \) with \( 1 \leq \tau_1 < \tau_2 \leq T \) such that \( q^\tau_2 > 0 \) and \( q^\tau_1 > 0 \). From Lemma 7, we have \( t_j \geq \tau_2 > \tau_1 \) and \( t_k \leq \tau_1 < \tau_2 \), implying \( t_k < t_j \). For an implementing price vector to exist, we need
\[
\nu^{j}(\theta_j) - \nu^{j}(\theta_k; \theta_j) \leq \nu^{k}(\theta_j; \theta_k) - \nu^{k}(\theta_k),
\]
where the first inequality comes from the proof of Proposition 1 and the second inequality comes from Lemma 6. If all utility functions are strictly concave, the first inequality is strict; if \( u^{\tau_1} - u^{\tau_2} \) is strictly decreasing for every \( \tau_1 \) and \( \tau_2 \) with \( 1 \leq \tau_1 < \tau_2 \leq T \), then the second inequality is strict, a contradiction. Hence \( t_j \leq t_k \) for every \( j \) and \( k \) with \( 1 \leq j < k \leq K \). \( \square \)

**B.11 Proof of Theorem 3**

*Proof.* From Proposition 6, Condition (a) is necessary for the implementability of \( q \). From Lemma 14, it suffices to check that \( p^* \) as defined in the invoked proposition implements \( q \). Since \( \nu^{T}(\theta_K) \geq 0 \), \( IR^T_K \) holds. It remains to show that for for every \( j \) and \( l \) with \( 1 \leq j < l \leq K \), \( (p^*, q) \) satisfies IC\(_{lj}\). Set \( p = p^* \). First IC\(_{lj}\) implies
\[
p_j - p_l \geq \nu^{l}(\theta_j; \theta_l) - \nu^{l}(\theta_l) = \nu^{l}(\theta_j) - \nu^{l}(\theta_l) - \beta^{l} \frac{d}{2}.
\]

From \( p^* \), we have
\[
p_j^* - p_l^* = \sum_{k=j}^{l-1} p_k^* - p_{k+1}^* = \sum_{k=j}^{l-1} \nu^{k}(\theta_k) - \nu^{k}(\theta_{k+1}) - \beta^k \frac{d}{2},
\]
and hence \((p^*, q)\) satisfies IC\(_{ij}\) if and only if

\[
\frac{1}{2} \sum_{k=j}^{l-1} v^{t_k}(\theta_k) - v^{t_k}(\theta_{k+1}) - \beta^{t_k} d \geq v^{t_i}(\theta_j) - v^{t_i}(\theta_l) - \frac{\beta^{t_i} d}{2}
\]

\[
\iff \left[ v^{t_i}(\theta_j) - v^{t_i}(\theta_l) \right] - \left[ v^{t_{l-1}}(\theta_l) - v^{t_l}(\theta_l) \right] - \left[ \sum_{k=j}^{l-2} v^{t_k}(\theta_{k+1}) - v^{t_{k+1}}(\theta_{k+1}) \right] + \frac{d}{2} \left[ \beta^{t_i} - \sum_{k=j}^{l-1} \beta^{t_k} \right] \geq 0,
\]

plugging in the utility functions \(u^n_i = \alpha^i - \beta^n i d\), we get

\[
\left[ \alpha^{t_j} - \alpha^{t_i} - (\beta^{t_j} - \beta^{t_i}) Q_{j-1} d - \frac{(\beta^{t_j} - \beta^{t_i})(1 + q_j)}{2} d \right]
\]

\[
- \left[ \alpha^{t_{l-1}} - \alpha^{t_i} - (\beta^{t_{l-1}} - \beta^{t_i}) Q_{l-1} d - \frac{(\beta^{t_{l-1}} - \beta^{t_i})(1 + q_l)}{2} d \right]
\]

\[
- \left[ \sum_{k=j}^{l-2} \alpha^{t_k} - \alpha^{t_{k+1}} - (\beta^{t_k} - \beta^{t_{k+1}}) Q_k d - \frac{(\beta^{t_k} - \beta^{t_{k+1}})(1 + q_{k+1})}{2} d \right] + \frac{d}{2} \left[ \beta^{t_i} - \sum_{k=j}^{l-1} \beta^{t_k} \right] \geq 0,
\]

from which we solve for \(\beta^{t_i}\) to get

\[
\beta^{t_i} \leq \frac{-1 + q_k + q_{k+1}}{-1 + q_j + q_l + 2 \sum_{m=j+1}^{l-1} q_m} \beta^{t_k} := \beta_j^{t_i},
\]

which is independent of \(d\). We get \(\beta_j^{t_i}\) for every \(\theta_j < \theta_l\) and set \(f_i(\beta^{t_i}, \ldots, \beta^{t_{l-1}}) = \min_{j<k} \beta_j^{t_i}\) and the proof is complete. \(\square\)

**B.12 Proof of Proposition 7**

**Proof.** From Lemma 7, Proposition 6 and Proposition 1, we have \(t_1 = t^2 - 1\). So the relevant IC constraints here only some pair of adjacent types. Here, the necessary and sufficient condition for implementability is

\[
\frac{\beta^1}{\beta^2} \geq \frac{\nu(\theta_1; \theta_2) - \nu(\theta_2)}{\nu(\theta_1) - \nu(\theta_2; \theta_1)},
\]

where \(\nu\) is associated with \(u\). As \(u\) is strictly concave, the right-side of the inequality is strictly above 1, and hence the inequality is not trivial. Lastly, if \(N^t\) tends to infinity for any \(t \geq t_2\), both \(\nu(\theta_2)\) and \((\theta_2; \theta_1)\) converge to \(-\infty\), and the right-side converges to 1. \(\square\)
B.13 Proof of Proposition 10

Proof. From Lemma 10, there is an implementable scheme with two passes if and only if

\[
\beta^l \leq \frac{\nu(\theta_1) - \nu(\theta_2; \theta_1)}{\nu(\theta_1; \theta_2) - \nu(\theta_2)}.
\]

The revenue of having only one pass is \(\beta^l (N^h \nu(\theta_1) + N^h \nu(\theta_2))\). When the above inequality holds, from Lemma 10, the revenue from the optimal two-pass scheme is

\[
N^l p^*_2 + N^h p^*_1 = N^l \beta^l \nu(\theta_2) + N^h \left[ \beta^l \nu(\theta_2) + \beta^l (\nu(\theta_1; \theta_2) - \nu(\theta_2)) \right].
\]

Conditional on that \(\nu(\theta_1) - \nu(\theta_2; \theta_1) \geq \beta [\nu(\theta_1; \theta_2) - \nu(\theta_2)]\), the two-pass scheme is better than the one-pass scheme if and only if

\[
N^l \beta^l \nu(\theta_2) + N^h \left[ \beta^l \nu(\theta_2) + \nu(\theta_1; \theta_2) \right] \geq \beta^l \left( N^l \nu(\theta_2) + N^h \nu(\theta_1) \right),
\]

from which we get

\[
\beta^l \leq \frac{\nu(\theta_1) - \nu(\theta_2; \theta_1)}{\nu(\theta_1) - \nu(\theta_2)} \leq \frac{\nu(\theta_1) - \nu(\theta_2; \theta_1)}{\nu(\theta_1; \theta_2) - \nu(\theta_2)} \in (0, 1),
\]

which shows that there exists \(\beta^l\) such that there exists an implementable two-pass scheme whose profit is higher than the one-pass scheme. \(\square\)

B.14 Proof of Proposition 11

Proof. If for every \(N\), we can find some scheme \(q\) such that there exists some \((p, q)\) that satisfies the IC constraints, then we can find \(u(N) < u_N\) such that \(q\) is implementable with \(u_0 = u(N)\).

If \(K \leq 2\), then the scheme is clearly implementable. Hence assume \(K > 2\). We introduce Lemma 5 in Appendix A. The lemma gives us a sufficient condition for implementation, which we use for this proof.

Consider a scheme \(q = (q_1, \ldots, q_K)\) with every customer buying some pass. The convexity of \(u\) implies that all ID conditions hold. From Lemma 5, it suffices to have \(\nu(\theta_l) - \nu(\theta_{l+1}; \theta_l) \geq \nu(\theta_l; \theta_{l+1}) - \nu(\theta_{l+1}; \theta_{l+2})\) for every \(l\) with \(1 \leq l \leq K - 2\). The inequality is equivalent to

\[
\frac{\sum_{n=Q_{l-1}+1}^{Q_l} u_n}{Q_l} - \frac{\sum_{n=Q_l}^{Q_{l+1}} u_n}{q_l + 1} \geq \frac{\sum_{n=Q_{l+1}}^{Q_{l+1}+1} u_n}{Q_{l+1}} - \frac{\sum_{n=Q_{l+1}+1}^{Q_{l+1}+1} u_n}{Q_{l+1} + 1},
\]
from which we get
\[
\frac{\sum_{n=Q_{t-1}+1}^{Q_t} (u_n - u_{Q_{t+1}})}{q_t(q_t + 1)} \geq \frac{u_{Q_t} - u_{Q_{t+1}}}{q_{t+1} + 1},
\]  
(3)
whose left-side is fixed and strictly positive. To make the inequality hold, it suffices to have the right-side converge to 0 when \(q_{t+1}\) approaches infinity. The right converges to 0 if \(\lim_{n \to \infty} u_n = 0\), proving Equation (3) Assume \(\lim_{n \to \infty} u_n \neq 0\). Since
\[
\frac{u_{Q_t} - u_{Q_{t+1}}}{q_{t+1} + 1} = \frac{u_{Q_t} - u_{Q_{t+1}}}{u_{Q_{t+1}} + 1} \times \frac{Q_{t+1} + 1}{q_{t+1} + 1} \times \frac{u_{Q_{t+1}}}{Q_{t+1} + 1},
\]
the first part of the multiplicative term converges to some finite real, the second part converges to 1, and the last term converges to 0 according the assumption that \(\lim_{n \to \infty} \frac{u_n}{n} = 0\). Hence Equation (3) holds for large enough \(q_{t+1}\). Therefore, given \(q_1\) and \(q_2\), we can find \(q_3\) large enough such that the sufficient conditions in Lemma 5 for those that involve only \(\theta_1, \theta_2\) or \(\theta_3\) hold. Given \(q_1, \ldots, q_k\) for \(k < K\), we can find \(q_{k+1}\) large enough such that the sufficient conditions involving only \(\theta_1, \ldots, \theta_k\) hold. This procedure is finite since \(K\) is fixed and finite. Hence for some \(M\) large enough, for all \(N \geq M\), we can find an implementable scheme for \(K\). \(\square\)

**B.15 Proof of Theorem 4**

*Proof*. For every \(N\), if there exists some scheme \(q\) such that there is some \((p, q)\) that satisfies the IC constraints, then setting \(u(N) < u_N\) will make \(q\) implementable with \(u_0 = u(N)\). Given \(N\) and \(K\), and a scheme \(q = (q_1, \ldots, q_K)\) where every customer buys some pass, we consider the pricing where \(p_k = \nu(\theta_k)\) for every \(k\). With this pricing, \(\epsilon \text{IC}_{jk}\) holds for every \(j\) and \(k\) with \(1 \leq j < k \leq K\). For \(\epsilon \text{IC}_{jk}\) where \(j < k\) to hold, we need
\[
\nu(\theta_k; \theta_j) - p_k - \left[\nu(\theta_j; \theta_j) - p_j\right] - \epsilon = \frac{u_{Q_{k-1}}}{q_k + 1} \nu(\theta_k) - \frac{\nu(\theta_k)}{q_k + 1} - \epsilon < 0.
\]

We have \(\lim_{q_k \to \infty} \frac{u_{Q_{k-1}}}{q_k + 1} = 0\). It remains to show that \(\lim_{q_k \to \infty} \frac{\nu(\theta_k)}{q_k + 1} = 0\), which we claim is implied by \(\lim_{n \to \infty} \frac{u_n}{n} = 0\). To see this, note that
\[
\frac{\nu(\theta_k)}{q_k + 1} = \frac{1}{q_k + 1} \sum_{n=Q_{k-1}+1}^{Q_k} \frac{u_n}{q_k},
\]
which would converge to 0 if \(\lim_{n \to \infty} \frac{u_n}{n} = 0\). Hence \(\epsilon \text{IC}_{jk}\) holds. Therefore, given \(q_1 > 0\), we can find \(q_2\) large enough such that \(\text{IC}_{12}\) holds with \(p_k = \nu(\theta_k)\) for every \(k\). Then given \(q_1, \ldots, q_{k-1}\) for some \(k \leq K\), for each previous pass, we can find \(q_k\) large enough such that \(\epsilon \text{IC}_{jk}\) holds for every \(j\) with \(1 \leq j \leq k - 1\). The procedure terminates after a finite number of times since \(K\) is fixed. Thus
there exists $M$ such that for all $N \geq M$, there is a scheme $(q_1, \ldots, q_K)$ that can be implemented by $p_k = v(\theta_k)$ for every $k$. □