C.Ö. Sayin

# Unobservable and Observable Make-to-Stock Queueing Systems with One or Two Products

Master Thesis

January 12th, 2026

Thesis supervisors: dr. O. Kanavetas
E. Kosarevskaia

Leiden University
Mathematical Institute

## Abstract

In this thesis we consider make-to-stock queueing systems with strategic customers, describing customer's optimal behavior and seeking optimal decision variables with respect to producer's profit as well as overall social welfare of producer and customers. These decision variables are the product price and the target inventory level. We make distinctions between one-product and two-product systems, as well as unobservable and observable systems. A system being unobservable or observable represents the level of information granted to arriving customers, which influences their behavior and thus the corresponding decisions of the producer or social planner. Additionally, we present a numerical analysis for the observable one-product system in which results relating to (the optimization of) producer's profit and social welfare are visualized, as well as their sensitivities to the modification of key system parameters.

## Contents

## 1 Introduction

The mathematical theory of queueing systems knows many applications. Queueing theory's ability to simulate a broad scope of systems in which there exists some continuous demand for a finite resource has proven useful in multiple areas such as computing, manufacturing and commerce. One practical use-case for queueing theory is the simulation of a system in which fulfillment of demand incurs some reward while waiting for fulfillment carries a penalty. Such characterics can be found, for example, in a system consisting of a facility which serves incoming customers, like a food vendor. When customers receive service (or analogously, a product) they receive a reward, while they pay some price for joining the system and spending time waiting in the queue.

One might assume that the customers in this situation have some agency over their participation in the system, deciding whether or not to join the system based on their perceived benefit. This invariably involves a trade-off between reward and costs. The customer's decision has to be based on information about the state of the system, for which several levels of observability may be assumed.

To encourage customers to join the system, a producer may choose to maintain product inventory. To that end, the producer continuously sustains production until a fixed target inventory level is reached. A practical consideration is the presence of so-called holding costs associated with the storage of inventory.

Several metrics to evaluate performance of such a system can be considered, for example the producer's profit as well as the social welfare which takes into account the net benefit of both the producer as well as the customers. In order to maximize the desired objective, key system variables like the price and target inventory level can be adjusted. This results in an optimization problem which involves a trade-off between producer's revenue, producer's holding costs, customer's reward and customer's waiting and product costs.

In this thesis, we evaluate unobservable and observable make-to-stock queueing systems with strategic customers, for the one-product as well as the two-product case. For these systems we seek to answer the following question.

*Which (joining) strategy is optimal for individual customers arriving in our unobservable/observable one-product/two-product make-to-stock queueing system and how can the decision variables be chosen such that the producer's profit/social welfare is optimal?*

Section 2 presents an overview of prerequisite theory regarding queueing systems, make-to-stock systems and strategic customers. In Section 3 the one-product model will be introduced. A previous analysis for the unobservable case of this model will be reviewed in Section 4. Section 5 covers an analysis of the observable one-product model. We will discuss the extension of the one-product model to the two-product model in Section 6. Section 7 then offers a review of work done on the unobservable two-product model, while Section 8 briefly goes over key insights for the observable two-product model. We conclude with a discussion about the results and potential future work in Section 9. Appendix A contains the Python code used for the numerical analysis.

# 2 Preliminary Theory

The queueing systems analyzed in this thesis build upon the standard M/M/1 queue and additionally assume strategic customers as well as a make-to-stock production approach. This section provides a brief overview of the prerequisite theory.

## 2.1 Queueing Systems and the M/M/1 Queue

We initially consider a setting in which one server, or producer, produces one type of resource, or product, for which there exists some demand. According to this demand customers arrive at a service facility in order to receive one unit of product. To meet demand, the server persists production as long as at least one customer is present in the system. Since the production of one product takes a non-zero amount of time, a queue of waiting customers is able to form, leading to a so-called *queueing system*.

The following assumptions on the queueing system will form the basis of our analysis:

- Customers arrive according to a Poisson process with parameter $\lambda > 0$, meaning that the periods of time between consecutive customer arrivals, or inter-arrival times, are mutually independent exponential stochastic variables with parameter $\lambda$. From this follows that the mean inter-arrival time equals $\lambda^{-1}$;
- The production time of one product is an exponential stochastic variable with parameter $\mu > 0$, from which follows that the mean production time equals $\mu^{-1}$;
- We have $\lambda < \mu$, meaning that (on average) customer arrivals are less frequent than production completions;
- Production is handled by one singular server on a first-come, first-served (FCFS) basis; upon production completion the first customer in the queue receives the product and leaves the system immediately.

These assumptions together give rise to a M/M/1 queue in Kendall's notation.[4] Associated to this M/M/1 queue is a stochastic process known as a birth-and-death process. If we denote the possible states of the system by the amount of waiting customers, that is, the states are denoted $i$ for $i \geq 0$, then the system transitions from state $i$ to $i + 1$ with rate $\lambda$ (corresponding to the arrival of a new customer) and from state $i \neq 0$ to $i - 1$ with rate $\mu$ (corresponding to a production completion and the departure of the customer receiving the completed product). Since in each state the transition probabilities are independent from the states the system was previously in, the process is a Markov chain as illustrated by Figure 1. By the assumption $\lambda < \mu$, the system tends (as time increases) to a unique *stationary distribution* $\boldsymbol{\pi}$ over the possible states $i \geq 0$; if for all states $i$, the probability of the system being in state $i$ is described by $\pi_i$, then this probability does not change over time, meaning that the distribution $\boldsymbol{\pi}$ is stationary. It describes the situation

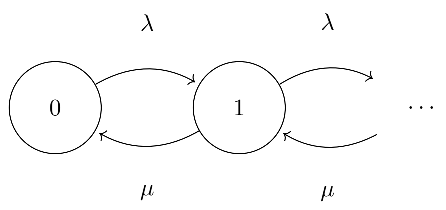


Figure 1: Markov chain corresponding to the M/M/1 queue.

in which the system is in equilibrium: for obtaining $\boldsymbol{\pi}$, we use the local balance equations

$$\lambda\pi_i = \mu\pi_{i+1} \text{ for } i \geq 0.$$

Writing $\rho = \lambda/\mu$, the balance equations yield the stationary distribution probabilities $\pi_i = (1-\rho)\rho^i$ for $i \geq 0$.

## 2.2 Make-to-Stock

In the M/M/1 queue, the server follows a *make-to-order* approach: initiation (or continuation) of production is dependent on the arrival of a customer, which represents an order. No production occurs when no customers are present in the queue.

In the following, however, the server follows a *make-to-stock* approach. In this setting a target inventory level $S$ is set by the server beforehand. Production halts when inventory level reaches the target level $S$. In general this approach obviously decreases the average waiting time: if a customer joins the system and finds a product present in inventory, they take it and leave immediately.

## 2.3 Strategic Customers

We assumed that arriving customers always join the system. In general, this need not be the case. A more realistic analysis takes into account the ability of arriving customers to decide whether or not to join the system, leading to a distinction between arrival rate and *effective* arrival rate (the rate at which customers actually join the system). If the utility an arriving customer gains from joining the queue is indicated by $U$, then they *join* the system if $U > 0$ and *balk* (leave) if $U < 0$. We arbitrarily assume that in the case that $U = 0$, an arriving customer *joins* the system.

A customer's utility evaluation is based on the *information level* they are granted. We distinguish two cases:

1. The system is *unobservable*, meaning that an arriving customer has no knowledge about the state of the system. That is, neither the amount of customers

waiting in the queue nor the amount of products in inventory are known. In this case, customers calculate their *expected* utility.

2. The system is *observable*, meaning that an arriving customer has full knowledge about the state of the system. That is, to an arriving customer the amount of customers waiting in the queue as well as the amount of products in inventory are known. In this case, customers are able to calculate their utility based on this information.

## 3 Description of the One-Product Model

Now we come to our first model of interest. We consider a one-product, make-to-stock queueing system with strategic customers. Customers arrive according to a Poisson process with rate $\lambda > 0$. The one singular server at the service facility handles production with a production rate of $\mu > 0$ and FCFS order. Additionally we assume that $\lambda < \mu$, or equivalently $\rho = \lambda/\mu < 1$.

The server employs a make-to-stock production approach with an initially fixed target inventory level of $S$ products. When the inventory level is short of $S$, the producer sustains production (at rate $\mu$) until an inventory level of $S$ products is reached. Each product in inventory incurs a holding cost of $h$ per unit time to be payed by the server. Furthermore, the server charges to each joining customer a price of $p$ to be payed at the instant of joining the system.

If an arriving customer decides to join the system, they receive a reward of $R > p$ upon service completion (receipt of the product). If the product is in stock they pay the price $p$, receive the product and leave immediately. Note that this then triggers a replenishment job for the server as the inventory level decreases by one. If, on the other hand, the product is *not* in stock, they pay price $p$ and join the queue, additionally paying a waiting cost of $c$ per in-queue unit time, before receipt of the product and departure.

The producer is able to set the target inventory level $S$ based on the anticipated behavior of the customers. This behavior greatly depends on the information level provided to them. We will specifically make the distinction between an *unobservable* and an *observable* system as described in Section 2.3, indicating whether the current state of the system, that is, the amount of waiting customers as well as the amount of products in inventory, is known to an arriving customer. We assume that the *target* inventory level $S$ is always known to the customers.

## 4 Unobservable One-Product Model

Öz and Karaesmen[6] carried out an analysis for the case in which the system is *unobservable* to customers, meaning that an arriving customer knows neither the amount of customers waiting in the queue nor the amount of products in inventory. The target inventory level $S$ *is* known. This section provides a brief review of this work. Since the target inventory level $S$ set by the producer depends on the customer's behavior, we begin by solving the so-called customer's problem.

### 4.1 Customer's Problem

As we assume that the current queue length as well as the current inventory level are unobservable to an arriving customer, the decision whether to join or balk is described by a joining probability $q \in [0, 1]$. By the unobservability of the system, a customer's joining decision is based on the *expected* utility gained from joining the system. Let $U(q)$ be this expected utility for a (fixed) arriving customer when *all* other customers join with probability $q$. If we denote by $E[W(\bar{\lambda}, S)]$ the expected waiting time for given *effective* arrival rate $\bar{\lambda}$ (which in our case equals $\lambda q$ as $q$ describes joining probability) and target inventory level $S$, which Buzacott and Shantikumar [1] showed to be

$$E[W(\bar{\lambda}, S)] = \frac{\bar{\lambda}^S}{\mu^S(\mu - \bar{\lambda})},$$

then our expected utility function can be defined as

$$\begin{aligned} U(q) &= R - p - cE[W(\lambda q, S)] \\ &= R - p - c\frac{(\lambda q)^S}{\mu^S(\mu - \lambda q)}. \end{aligned}$$

Indeed, the joining customer is expected to receive reward $R$ and pay the product price $p$ as well as the expected waiting cost $cE[W(\lambda q, S)]$.

Note that by symmetry between customers the joining probability $q$ will be the same for all of them. We seek to find this joining probability given the system parameters. Recall that the target inventory level $S$ is known to the customers. We distinguish the following three cases for the state in which an arriving customer encounters the system.

1. Suppose $S = 0$ and $R - p < c/\mu$. Since then $cE[W(\lambda q, S)] = c/(\mu - \lambda q) > c/\mu$, the rightmost expression being the minimum waiting cost corresponding to an empty queue and immediate service, we have $U(q) < 0$ so that the arriving customer does not join the system. Note that if $R - p < c/\mu$ but $S > 0$, the arriving customer may in fact join, taking into account the possibility of receiving on-hand inventory.

2. Suppose $R - p > cE[W(\lambda, S)]$. This corresponds to the situation in which the expected utility is positive even if all other customers join (i.e. $q = 1$). Since then it also holds that $R - p > cE[W(\lambda q, S)]$, it follows that $U(q) > 0$ so that the arriving customer does join the system.

3. If neither of the above two cases hold, the customer joins the system with probability $q$. There exists a unique equilibrium $q^*$ such that $U(q^*) = 0$; if either $U(q^*) > 0$ or $U(q^*) < 0$ then more customers will join or leave the system, respectively, over time until $U(q^*) = 0$ .

It follows that the customers' *equilibrium* joining probability $q_S$ (as a function of the target inventory level $S$) is given by

$$q_S = \begin{cases} 0 & \text{if } S = 0 \text{ and } R - p < c/\mu, \\ 1 & \text{if } R - p > c\frac{\lambda^S}{\mu^S(\mu-\lambda)}, \\ q^* & \text{otherwise, where } U(q^*) = 0. \end{cases}$$

Furthermore, $q_S$ is non-decreasing in $S$. This is to be expected since a higher target inventory level generally results in a lower average waiting time as some joining customers receive on-hand inventory, leaving the system immediately.

## 4.2 Producer's Profit Optimization

Since the producer is aware of the customer's equilibrium joining probability $q_S$, a target inventory level $S$ can be set accordingly in order to maximize the expected *producer's profit* (per unit time). This expected profit $Z(S)$ is given by the difference between expected revenue and expected holding costs, that is,

$$Z(S) = p\lambda q_S - hE[I(S)],$$

where $E[I(S)]$ is the expected amount of products in inventory. Buzacott and Shantikumar[1] showed that

$$E[I(S)] = S - \frac{\lambda q_S}{\mu - \lambda q_S}\left(1 - \left(\frac{\lambda q_S}{\mu}\right)^S\right),$$

from which follows that

$$Z(S) = p\lambda q_S - h\left(S - \frac{\lambda q_S}{\mu - \lambda q_S}\left(1 - \left(\frac{\lambda q_S}{\mu}\right)^S\right)\right).$$

There exists a target inventory level $\bar{S}$ such that $q_S = 1$ (meaning that customers always join) and $Z(S) \leq Z(\bar{S})$ for $S \geq \bar{S}$. Hence, a profit-maximizing target inventory level $S^*$ can be found by Algorithm 1.

## 4.3 Social Welfare Optimization

Also of interest is the maximization of the so-called *social welfare*. This quantity considers the total utility the system grants to both the producer and the customers.

**Algorithm 1** Profit-maximizing target inventory level $S^*$

```
$k \leftarrow 0$
while $q_k \neq 1$ do
    $k \leftarrow k + 1$
end while
$S^* \leftarrow \arg\max\{\Pi(s) : 0 \leq s \leq k\}$
```

Since payment of the product price is nothing more than a transfer of wealth from customer to producer, we disregard the price $p$ for this analysis. Hence, the social welfare takes into account the producer's holding cost, the customers' reward and the customers' waiting cost. We assume that a social optimizer is able to control the target inventory level $S$ and the customer joining (i.e. effective arrival) rate $\bar{\lambda}$. The expected social welfare $T(\bar{\lambda}, S)$ is then given by

$$\begin{aligned} T(\bar{\lambda}, S) &= \bar{\lambda}R - hE[I(\bar{\lambda}, S)] - c\bar{\lambda}E[W(\bar{\lambda}, S)] \\ &= \bar{\lambda}R - h\left(S - \frac{\bar{\lambda}}{\mu - \bar{\lambda}}\left(1 - \left(\frac{\bar{\lambda}}{\mu}\right)^S\right)\right) - c\frac{\bar{\lambda}^{S+1}}{(\mu - \bar{\lambda})\mu^S}. \end{aligned}$$

In order to find optimal $(\bar{\lambda}, S)$ a sequential approach can be carried out as follows. First, find the optimal target inventory level $S^*_{\bar{\lambda}} = \arg\max_S T(\bar{\lambda}, S)$, which Veatch and Wein[7] characterized as

$$S^*_{\bar{\lambda}} = \left\lfloor \frac{\ln\left(\frac{h}{h+c}\right)}{\ln\left(\frac{\bar{\lambda}}{\mu}\right)} \right\rfloor.$$

Second, use $S^*_{\bar{\lambda}}$ to find the optimal customer joining rate $\bar{\lambda}^* = \arg\max_{\bar{\lambda}} T(\bar{\lambda}, S^*_{\bar{\lambda}})$. The resulting pair $(\bar{\lambda}^*, S^*_{\bar{\lambda}})$ maximizes the expected social welfare $T(\bar{\lambda}, S)$.

# 5 Observable One-Product Model

## 5.1 Model Description

In this section we again analyze the one-product, make-to-stock queueing system with strategic customers, now with an important alteration: we take the system to be *observable* to the customers. That is, to an arriving customer both the amount of customers waiting in the queue as well as the current amount of products in inventory is known, in addition to the target inventory level $S$. We suppose that the server acts on the customer's behavior by setting either the target inventory level $S$ or the product price $p$.

This analysis will be an extension of Naor's work[5], as described by Hassin and Haviv[2], which dealt with a similar observable one-product queueing system with strategic customers in which a make-to-order instead of make-to-stock approach is taken.

Again we begin by solving the customer's problem. It will prove useful to consider the Markov chain our system gives rise to. In Figure 2 this chain is depicted. The states $-S, \dots, -1$ indicate an inventory level and the states $0, 1, \dots$ indicate an amount of waiting customers. This description of the Markov chain is well-defined: since a joining customer immediately receives a product from inventory, it cannot be the case that both the queue and the inventory are non-empty.

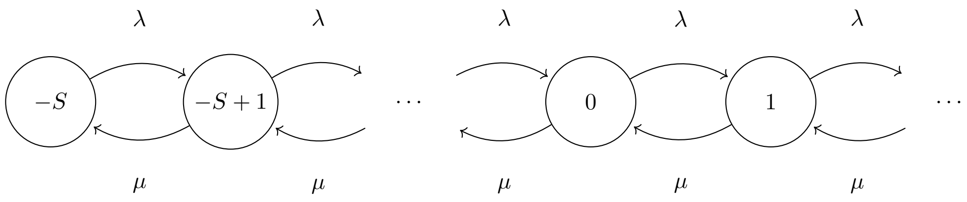


Figure 2: Markov chain corresponding to the observable one-product model.

## 5.2 Customer's Problem

By our observability assumption an arriving customer is aware of the current state $i \geq -S$ of the system. This information dictates the decision to join or balk, in contrast to the joining probability in the unobservable system considered in Section 4.

Remark that if an arriving customer finds the system to be in state $-S \leq i \leq -1$, they join to receive a product out of inventory immediately, incurring no waiting time, as we assumed that the reward $R$ of receiving a product exceeds the product price $p$.

If, however, an arriving customer encounters state $i \geq 0$, then their expected sojourn time will be $\frac{i+1}{\mu}$ since they are served only after the $i$ other customers in the system are served and service times are distributed exponentially with parameter $\mu$. Hence, the expected utility of a customer joining the system being in state, i.e. having queue length, $i \geq 0$ is given by

$$U(i) = R - p - \frac{c(i+1)}{\mu}.$$

Recall that an arriving customer joins the system if $U(i) \geq 0$ and balks if $U(i) < 0$. This gives rise to an additional assumption: if $R < c/\mu$ then $U(i) < R - c/\mu < 0$, meaning no customer ever joins (given that there is no inventory present). Hence, we assume that $R\mu/c \geq 1$. By $U(i)$ being obviously linear and decreasing in $i$, there exists an $i^*$ such that $U(i) \geq 0$ for $i \leq i^*$, and $U(i) < 0$ for $i > i^*$. Solving $U(i) = 0$ yields that these inequalities hold if and only if $i^* = (R-p)\mu/c - 1$. As states are described by *integers*, we may define the so-called joining threshold to be

$$n = \left\lfloor \frac{(R-p)\mu}{c} \right\rfloor. \tag{1}$$

Now the customers can be said to follow a *threshold strategy*: if the queue length is $n$ or higher, then $U(i) < 0$ and an arriving customer *balks*, but if the queue length is less than $n$, then $U(i) \geq 0$ and an arriving customer *joins*.

## 5.3 Producer's Profit Optimization

Based on the customer's decision strategy, which we found to be the threshold strategy with joining threshold $n$, the producer accordingly sets either of his controllable variables, namely the target inventory level $S$ or the product price $p$. The objective function is the expected *producer's profit* (per unit time) $Z(p, S)$ given by

$$Z(p, S) = p\bar{\lambda} - hE[I], \tag{2}$$

where $\bar{\lambda}$ is the effective arrival rate and $E[I]$ is the expected amount of products in inventory.

To compute $\bar{\lambda}$ we first note that the customers' threshold strategy reduces the Markov chain from Section 5.1 as follows. As we have seen, $n$ is the maximum amount of customers in the system: no arriving customer will join upon encountering this amount of customers in the queue. Hence, the possible states for our Markov chain become $-S, \dots, n$ as illustrated in Figure 3. The corresponding balance equations for this system are

$$\lambda\pi_j = \mu\pi_{j+1} \text{ where } j = -S, \dots, n-1.$$

Writing $\rho = \frac{\lambda}{\mu}$, we find the stationary distribution $\boldsymbol{\pi}$ given by

$$\pi_j = \frac{\rho^{j+S}(1-\rho)}{1-\rho^{n+S+1}} \text{ where } j = -S, \dots, n.$$

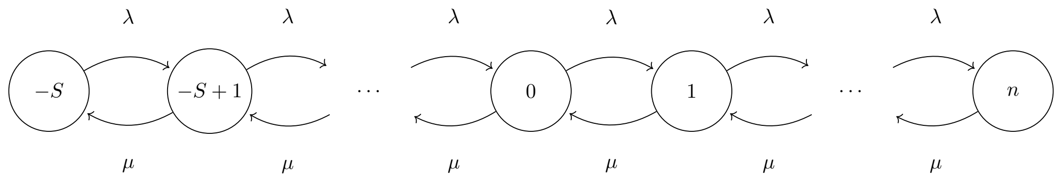


Figure 3: Markov chain corresponding to the observable one-product model taking into account joining threshold $n$.

Note that $\pi_n$ is the equilibrium probability of finding the maximum of $n$ customers in the system, that is, $\pi_n$ is the equilibrium *balking* probability. Consequently, $1-\pi_n$ is the equilibrium *joining* probability, from which we conclude that the effective joining rate equals

$$\bar{\lambda} = \lambda(1-\pi_n) = \lambda\frac{1-\rho^{n+S}}{1-\rho^{n+S+1}}. \tag{3}$$

We additionally note that the producer, seeking to maximize profit, sets the maximum price $p$ which is still in accordance with $n$ as described by (1). That is to say that we set

$$p = p(n) = R - \frac{nc}{\mu}. \tag{4}$$

Finally, since in state $-j$ we have $j$ products in inventory, we find that the expected amount of products in inventory is equal to

$$E[I] = \sum_{j=1}^{S} j\pi_{-j} = \frac{S(1-\rho)-\rho(1-\rho^S)}{(1-\rho)(1-\rho^{n+S+1})}. \tag{5}$$

Substituting these expressions for $\bar{\lambda}$, $p$ and $E[I]$ into (2), we find that in equilibrium the server's profit is

$$\begin{aligned} Z(n,S) &= \lambda\frac{1-\rho^{n+S}}{1-\rho^{n+S+1}}\left(R-\frac{nc}{\mu}\right) - h\frac{S(1-\rho)-\rho(1-\rho^S)}{(1-\rho)(1-\rho^{n+S+1})} \\ &= \frac{1}{1-\rho^{n+S+1}}\left(\lambda(1-\rho^{n+S})\left(R-\frac{nc}{\mu}\right) - h\frac{S(1-\rho)-\rho(1-\rho^S)}{1-\rho}\right). \end{aligned} \tag{6}$$

### 5.3.1 Producer Controls Target Inventory Level

In this section we take the producer's controllable variable to be the target inventory level $S$. That is to say that the producer aims to maximize the profit by varying $S$.

In order to ease into the analysis we first suppose that there are no holding costs, i.e. $h = 0$. Then the expected profit becomes

$$Z(S) = \lambda\frac{1-\rho^{n+S}}{1-\rho^{n+S+1}}\left(R-\frac{nc}{\mu}\right).$$

Since it then holds for all $n \geq 0$ and $S \geq 0$ that

$$Z(S+1) - Z(S) = \frac{\lambda(1-\rho)^2\rho^{n+S}(R - \frac{nc}{\mu})}{(1-\rho^{n+S+1})(1-\rho^{n+S+2})} > 0,$$

it follows that $Z$ is increasing in $S$; the higher the target inventory level, the higher the producer's profit. Indeed, given that there are no holding costs the producer profits from increasing inventory as this delays saturation of the system. This in turn results in more customers joining per unit time, netting the producer higher revenue hence higher profit.

Now we suppose that the holding costs are non-negative, so that $Z(S)$ is given by (6). In the absence of a rigorous proof for $Z(S)$ having a unique maximum $S^*$, we note that in practice there is an upper limit $S_{max}$ for the feasible target inventory levels. Indeed, realistically the producer does not have access to infinite storage space. Hence, it suffices to find

$$S^* = \arg\max\{Z(S) : 0 \leq S \leq S_{max}\},$$

which can be done in polynomial time.

### 5.3.2 Producer Controls Price

In this section we take the producer's controllable variable to be the product price $p$. Since the highest possible price $p = p(n)$ given a joining threshold $n$ can be found by (4), we carry out our analysis in terms of $n$.

The expression for $Z(n)$ as given by (6) can be rewritten as

$$Z(n) = A(n)\left(B(n)\frac{\nu - n}{\nu} - H\right),$$

where we define the substitutions

$$A(n) = \frac{1}{1-\rho^{n+S+1}},$$

$$B(n) = \lambda R(1-\rho^{n+S}),$$

$$\nu = \frac{R\mu}{c},$$

$$H = h\frac{S(1-\rho) - \rho(1-\rho^S)}{1-\rho}.$$

We are interested in a joining threshold $n^*$ maximizing $Z(n)$. Since $Z(n)$ is a *discrete* function, we require $n^*$ to satisfy the inequalities

$$\begin{cases} Z(n^*) > Z(n^*-1), \\ Z(n^*) \geq Z(n^*+1). \end{cases} \tag{7}$$

In order to simplify the written-out versions of these inequalities, we introduce for $\cdot \in \{A, B\}$ the following notation:

$$\begin{cases} \cdot & = \cdot(n^*), \\ \cdot' & = \cdot(n^* - 1), \\ \cdot^\dagger & = \cdot(n^* + 1). \end{cases}$$

Now the inequalities in (7) become

$$\begin{cases} A(B(\nu - n^*) - H\nu) > A'(B'(\nu - n^* + 1) - H\nu), \\ A(B(\nu - n^*) - H\nu) \geq A^\dagger(B^\dagger(\nu - n^* - 1) - H\nu). \end{cases}$$

Note that the term

$$A(n)B(n) = \frac{\lambda R(1 - \rho^{n+S})}{1 - \rho^{n+S+1}}$$

is increasing in $n$, since the factor $1 - \rho^{n+S}$ in the numerator decreases more slowly than the factor $1 - \rho^{n+S+1}$ in the denominator. From this follows $A'B' < AB < A^\dagger B^\dagger$. Hence, the inequalities can be summarized as

$$\frac{A'B' + (AB - A'B')n^*}{AB - A'B' - (A - A')H} < \nu \leq \frac{A^\dagger B^\dagger + (A^\dagger B^\dagger - AB)n^*}{A^\dagger B^\dagger - AB - (A^\dagger - A)H}.$$

Rewriting further, we find the first inequality in (7) to be

$$\nu > n^* + \frac{(A - A')Hn^* + A'B'}{A(B - H) - A'(B' - H)}$$

and the second inequality in (7) to be

$$\begin{aligned} \nu &\leq n^* + \frac{(A - A^\dagger)Hn^* - A^\dagger B^\dagger}{A(B - H) - A^\dagger(B^\dagger - H)} \\ &= n^* + \frac{(A^\dagger - A)Hn^* + A^\dagger B^\dagger}{A^\dagger(B^\dagger - H) - A(B - H)} \\ &= n^* + \frac{(A^\dagger - A)Hn^* + A^\dagger(B^\dagger - H) - A(B - H) + AB + (A^\dagger - A)H}{A^\dagger(B^\dagger - H) - A(B - H)} \\ &= n^* + \frac{(A^\dagger - A)H(n^* + 1) + AB + A^\dagger(B^\dagger - H) - A(B - H)}{A^\dagger(B^\dagger - H) - A(B - H)} \\ &= n^* + 1 + \frac{(A^\dagger - A)H(n^* + 1) + AB}{A^\dagger(B^\dagger - H) - A(B - H)}. \end{aligned}$$

We conclude that (7) can be written as

$$f(n^*) < \nu \leq f(n^* + 1) \tag{8}$$

where $f(n)$ is defined as

$$f(n) = n + \frac{(A(n) - A(n-1))Hn + A(n-1)B(n-1)}{A(n)(B(n) - H) - A(n-1)(B(n-1) - H)}. \tag{9}$$

We seek to show that (8) holds for a unique $n^*$ by evaluating the expression for $f(n)$ through the following three properties.

- Note that $B((-S+1)-1) = B(-S) = 0$, so that

$$\begin{aligned} f(-S+1) &= (-S+1)\left(1 + \frac{(A(-S+1)-A(-S))H}{A(-S+1)B(-S+1)-(A(-S+1)-A(-S))H}\right) \\ &= (-S+1)\left(1+\frac{\rho H/(\rho^2-1)}{\lambda R(1-\rho)/(1-\rho^2)-\rho H/(\rho^2-1)}\right) \\ &= (-S+1)\left(1+\frac{H}{\underbrace{\lambda R(1-\rho)/\rho - H}_{<0}}\right) \\ &< -S+1 \\ &\leq 1. \end{aligned}$$

- $f(n)$ is increasing for $n > S+1$. Indeed, using the substitutions $C = \lambda R(1-\rho)/\rho$ and $x = \rho^{n+S}$, we first note that

$$-(A(n)-A(n-1))H = \frac{(1-\rho)x}{(1-x)(1-\rho x)}H,$$

$$A(n-1)B(n-1) = \frac{\lambda R(1-x/\rho)}{1-x} = \frac{\lambda R(1-\rho x - x/\rho + x^2)}{(1-x)(1-\rho x)},$$

$$A(n)B(n) - A(n-1)B(n-1) = \frac{(1-\rho)x}{(1-x)(1-\rho x)}C.$$

Then (9) can be written as

$$\begin{aligned} f(n) &= n + \frac{(A(n)-A(n-1)Hn + A(n-1)B(n-1)}{A(n)B(n)-A(n-1)(B(n-1)-(A(n)-A(n-1))H} \\ &= n + \frac{-(1-\rho)xHn + \lambda R\left(1-\rho x - x/\rho + x^2\right)}{(1-\rho)x(C+H)} \\ &= n + \frac{-Hn}{(C+H)} + \frac{\lambda R(x+1/x-\rho-1/\rho)}{(1-\rho)(C+H)} \\ &= \frac{C}{(C+H)}n + \frac{\lambda R}{(C+H)(1-\rho)}\left(\rho^{n+S}+\rho^{-(n+S)}-\rho-1/\rho\right) \end{aligned}$$

so that it is clear that

$$f(n+1)-f(n) = \frac{C}{C+H} + \frac{\lambda R}{C+H}\left(\rho^{-(n+S+1)}-\rho^{n+S}\right) > 0$$

for $n > S+1$.

- We assumed in Section 5.2 that $\nu = R\mu/c \geq 1$.

From these properties it follows by (8) that there exists a *unique* $n^* \geq -S+1$ satisfying (7), being the necessary conditions for maximizing $Z(n)$. If we find $n^* < 0$, we conclude that the parameters do not result in a system in which it makes sense for the producer to do business. Otherwise, we find the unique profit-optimizing joining threshold $n^*$ and corresponding profit-optimizing product price $p^* = p(n^*)$.

### 5.4 Social Welfare Optimization

We now turn once again to the social welfare, which takes into account the producer's holding cost, the customers' reward and the customer's waiting cost. For this setting, the social optimizer is able to control the target inventory level $S$ as well as the joining threshold $n$, which corresponds directly to the price $p = p(n)$ through (4). The objective function is the *social welfare* (per unit time) $T(p, S)$ given by

$$T(p, S) = \bar{\lambda}R - cE[L] - hE[I],$$

where $\bar{\lambda}$ is the effective arrival rate and $E[L]$ the expected amount of customers in the system.

Plugging in the expression for $\bar{\lambda}$ in (3), the expression for $E[I]$ in (5) as well as the expected amount of customers in the system

$$E[L] = \sum_{j=1}^{n} jq_j = \frac{\rho^{S+1}(1-(n+1)\rho^n + n\rho^{n+1})}{(1-\rho)(1-\rho^{n+S+1})},$$

we find that in equilibrium the social welfare is

$$\begin{aligned} T(n, S) &= \lambda R\frac{1-\rho^{n+S}}{1-\rho^{n+S+1}} - c\frac{\rho^{S+1}(1-(n+1)\rho^n + n\rho^{n+1})}{(1-\rho)(1-\rho^{n+S+1})} - h\frac{S(1-\rho)-\rho(1-\rho^S)}{(1-\rho)(1-\rho^{n+S+1})} \\ &= \frac{1}{1-\rho^{n+S+1}}\left(\lambda R(1-\rho^{n+S}) - c\frac{\rho^{S+1}(1-(n+1)\rho^n + n\rho^{n+1})}{(1-\rho)}\right. \\ &\qquad \left. - h\frac{S(1-\rho)-\rho(1-\rho^S)}{(1-\rho)}\right). \end{aligned} \tag{10}$$

#### 5.4.1 Producer Controls Target Inventory Level

Analogously to our analysis for the producer's profit, we first consider the case in which there are no holding costs, i.e. $h = 0$. Then the expected social welfare becomes

$$T(S) = \frac{1}{1-\rho^{n+S+1}}\left(\lambda R(1-\rho^{n+S}) - c\frac{\rho^{S+1}(1-(n+1)\rho^n + n\rho^{n+1})}{(1-\rho)}\right).$$

Obviously, $T$ is increasing in $S$: given that there are no holding costs the welfare increases with inventory as more inventory delays saturation of the system. This in turns results in more customers joining per unit time, netting them higher total reward at no holding cost to the producer.

Now we suppose that the holding costs are non-negative, so that $T(S)$ is given by (10). Again, we note that the practical upper limit $S_{max}$ on the feasible target inventory level allows us to find

$$S^* = \arg\max\{T(S) : 0 \leq S \leq S_{max}\}$$

in polynomial time.

### 5.4.2 Producer Controls Price

We now take the social planner's controllable variable to be the product price $p$. Again, we carry out the analysis in terms of $n$, using the expression for $p = p(n)$ in (4).

Similarly to the producer's profit case, we rewrite the expression for $T(n)$ in (10), yielding

$$T(n) = A(n)(\nu B(n) - C(n) - H^*)$$

where we define the substitutions

$$A(n) = \frac{1}{1-\rho^{n+S+1}},$$

$$B(n) = 1 - \rho^{n+S},$$

$$C(n) = c\frac{\rho^{S+1}(1-(n+1)\rho^n + n\rho^{n+1})}{(1-\rho)},$$

$$\nu = \lambda R,$$

$$H = h\frac{S(1-\rho) - \rho(1-\rho^S)}{1-\rho}.$$

We are interested in a joining threshold $n^*$ maximizing $T(n)$, requiring accordance with the inequalities

$$\begin{cases} T(n^*) > T(n^*-1) \\ T(n^*) \geq T(n^*+1). \end{cases} \tag{11}$$

Again we simplify the written-out expressions by introducing for $\cdot \in \{A, B, C\}$ the notation

$$\begin{cases} \cdot & = \cdot(n^*), \\ \cdot' & = \cdot(n^*-1), \\ \cdot^\dagger & = \cdot(n^*+1). \end{cases}$$

Then the inequalities in (11) become

$$\begin{cases} A(\nu B - C - H) > A'(\nu B' - C' - H), \\ A(\nu B - C - H) \geq A^\dagger(\nu B^\dagger - C^\dagger - H). \end{cases}$$

Since $A(n)B(n)$ is increasing in $n$ as we similarly saw in Section 5.3.2 (in which $A(n)B(n)$ included the factor $\lambda R$), we can summarize the inequalities as

$$\frac{A(C+H) - A'(C'+H)}{AB - A'B'} < \nu \leq \frac{A^\dagger(C^\dagger + H) - A(C+H)}{A^\dagger B^\dagger - AB}.$$

We conclude that (11) can be written as

$$f(n^*) < \nu \leq f(n^*+1) \tag{12}$$

where

$$f(n) = \frac{A(n)(C(n)+H) - A(n-1)(C(n-1)+H)}{A(n)B(n) - A(n-1)B(n-1)}. \tag{13}$$

Again we consider three properties of $f(n)$, from which will follow that (12) holds for a unique $n^*$.

- Note that $C(0) = 0$ as well as $C(-1) = 0$, so that

$$\begin{aligned} f(0) &= \frac{[A(0) - A(-1)]H}{A(0)B(0) - A(-1)B(-1)} \\ &= \frac{[1/(1-\rho^{S+1}) - 1/(1-\rho^S)]H}{(1-\rho^S)/(1-\rho^{S+1}) - (1-\rho^{S-1})/(1-\rho^S)} \\ &= -\frac{H\rho}{1-\rho} \\ &\leq 0. \end{aligned}$$

- $f(n)$ is increasing for $n > 0$. Indeed, we first note that the numerator in (13) equals

$$\frac{C(n)+H}{1-\rho^{n+S+1}} - \frac{C(n-1)+H}{1-\rho^{n+S+1}}$$

and the denominator equals

$$\frac{(1-\rho)^2\rho^{n+S-1}}{(1-\rho^{n+S})(1-\rho^{n+S+1})}.$$

Then (13) can be written as

$$\begin{aligned} f(n) &= \frac{(C(n)+H)\left(1-\rho^{n+S}\right) - (C(n-1)+H)\left(1-\rho^{n+S+1}\right)}{(1-\rho)^2\rho^{n+S-1}} \\ &= \frac{C(n)\left(1-\rho^{n+S}\right) - C(n-1)\left(1-\rho^{n+S+1}\right) - H\rho^{n+S}(1-\rho)}{(1-\rho)^2\rho^{n+S-1}} \\ &= \frac{c\rho^{n+S}\left(n(1-\rho) - \rho^{S+1}(1-\rho^n)\right) - H\rho^{n+S}(1-\rho)}{(1-\rho)^2\rho^{n+S-1}} \\ &= \frac{\rho}{1-\rho}\left(cn - \frac{c\rho^{S+1}(1-\rho^n)}{1-\rho} - H\right) \end{aligned}$$

so that it is clear that

$$f(n+1) - f(n) = \frac{\rho}{1-\rho}\left(c - \frac{c\rho^{S+1}}{1-\rho}\rho^n(1-\rho)\right) = \frac{\rho}{1-\rho}c\left(1-\rho^{n+S+1}\right) > 0$$

for $n > 0$.

- We have $\nu = \lambda R > 0$.

It now follows by (12) that there exists a *unique* $n^* \geq 0$ satisfying (11), being the necessary conditions for maximizing $T(n)$. This is the unique social-welfare-optimizing joining threshold $n^*$ from which the corresponding social-welfare-optimizing product price $p^* = p(n^*)$ follows.

## 5.5 Numerical Analysis

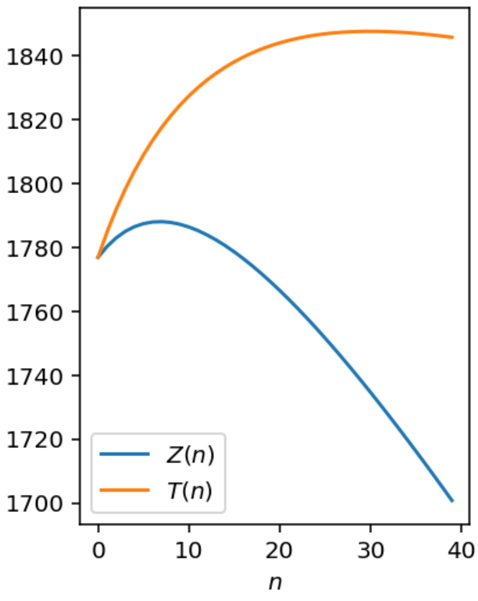
In this section we carry out a numerical analysis. We plot producer's profit and social welfare as functions of the joining threshold $n$ and target inventory level $S$, as well as the joining threshold and target inventory level maximizing these objectives as a function of the other controllable variable. Additionally, we discuss the effect of varying key system parameters.

*Base case*
For the base case, we suppose that the producer operates with a service rate of $\mu = 100$ and customers arrive at a rate of $\lambda = 98$. The producer incurs a holding cost of $h = 10$ per unit time per item in inventory. Customers are rewarded $R = 20$ upon receipt of the product and pay a waiting cost of $c = 5$ per unit time spent waiting. We also set $S = 20$ and $n = 20$ as defaults. See Figure 4 for plots of $Z$ and $T$ under this setting.

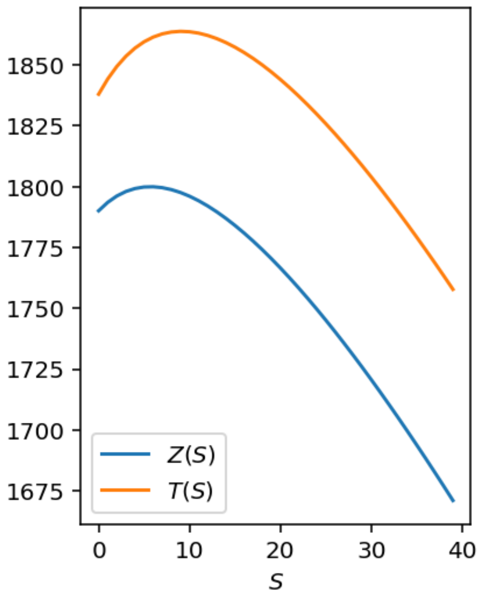
Optimizing over $n$ *and* $S$, we can compute the pair of joining threshold $n^*$ (yielding price $p^* = p(n^*)$) and target inventory level $S^*$ maximizing producer's profit $Z$ and social welfare $T$, see Table 1.

| | $n^*$ | $S^*$ | $p^*$ | $Z(n^*, S^*)$ | $T(n^*, S^*)$ |
|---|---|---|---|---|---|
| optimizing $Z(n, S)$ | 12 | 9 | 19.4 | 1808.08 | 1849.39 |
| optimizing $T(n, S)$ | 26 | 9 | 18.7 | 1781.11 | 1866.13 |

Table 1: Base-case optimal producer's profit and social welfare.



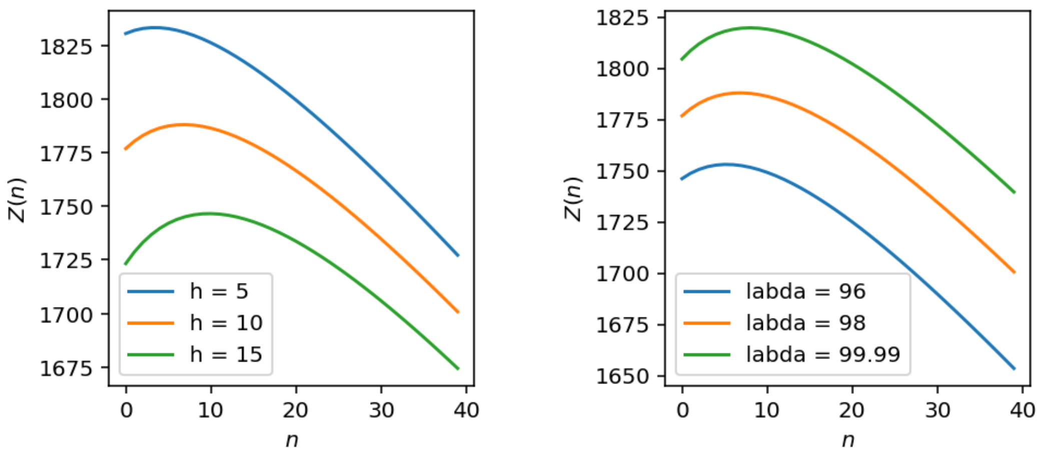
(a) $Z(n)$ and $T(n)$.



(b) $Z(S)$ and $T(S)$.

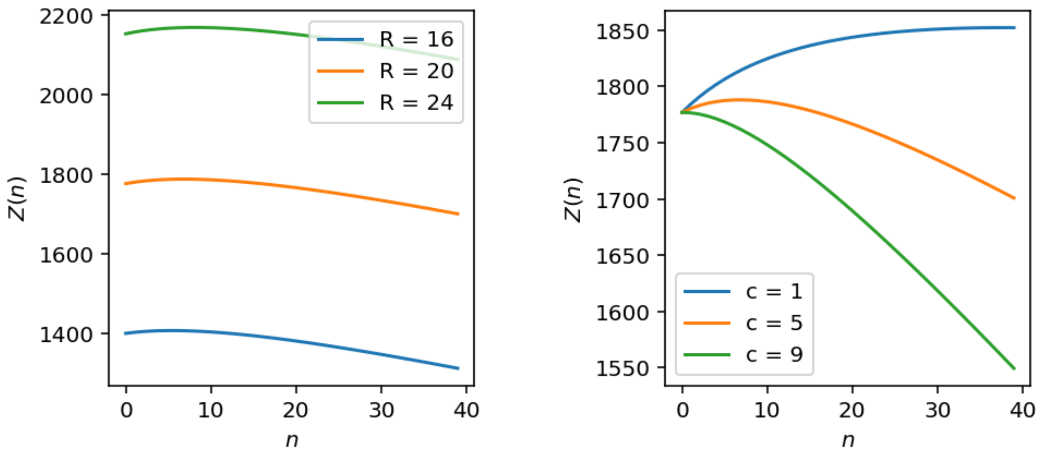
Figure 4: Base-case plots.

*Producer's Profit as a Function of Joining Threshold*
Consider the plots in Figure 5. For $n = 0$, the producer's profit is positive for all parameter choices. This can be explained by the presence of inventory (recall that $S = 20$); even if no queue ever forms, customers still receive product from inventory, resulting in revenue for the producer. As $n$ increases (or, the price $p$ decreases) the producer's profit increases up to the maximum we have proven to exist in Section 5.3.2 before decreasing again. This behavior illustrates the trade-off of decreasing price: less income per product versus an increase in clientiele.



(a) Varying holding cost $h$.

(b) Varying arrival rate $\lambda$.

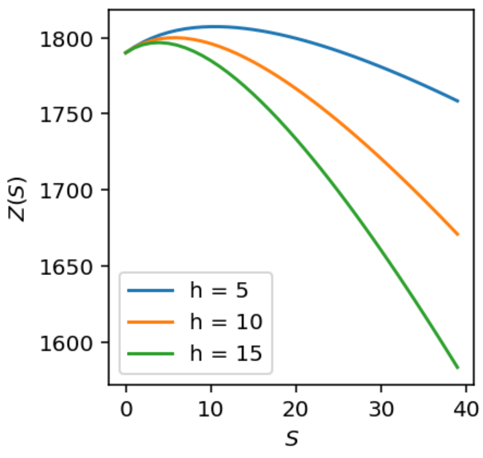


(c) Varying reward $R$.

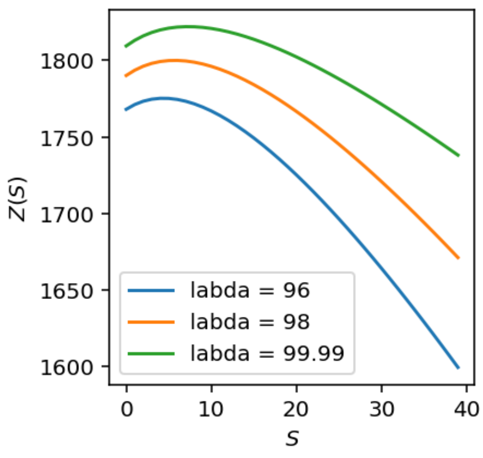
(d) Varying waiting cost $c$.

Figure 5: Sensitivity analysis for the producer's profit $Z(n)$.

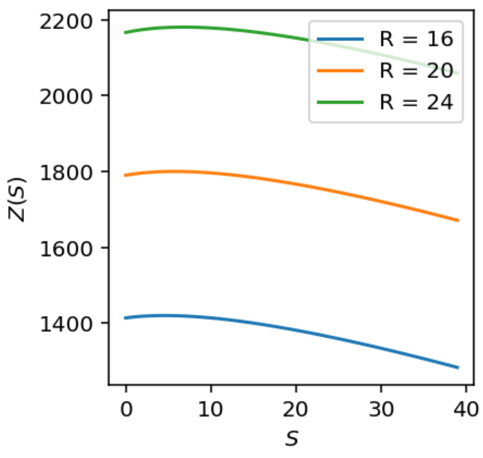
*Producer's Profit as a Function of Target Inventory Level*
Consider the plots in Figure 6. Though we have not proven it, the plots suggest that for fixed parameters there is indeed a profit-maximizing target inventory level $S^*$. The relevant trade-off of a higher target inventory level is more effectively meeting customer demand versus higher total holding costs due to an increase in inventory.



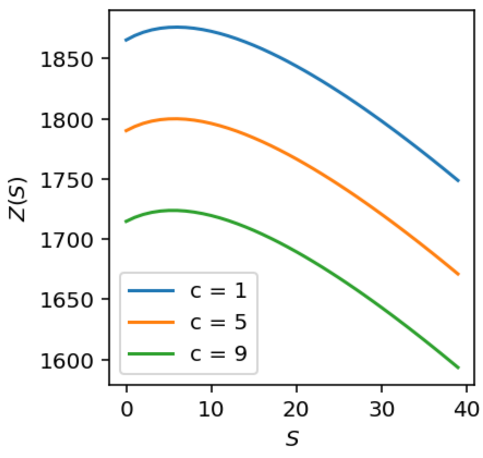
(a) Varying holding cost $h$.



(b) Varying arrival rate $\lambda$.



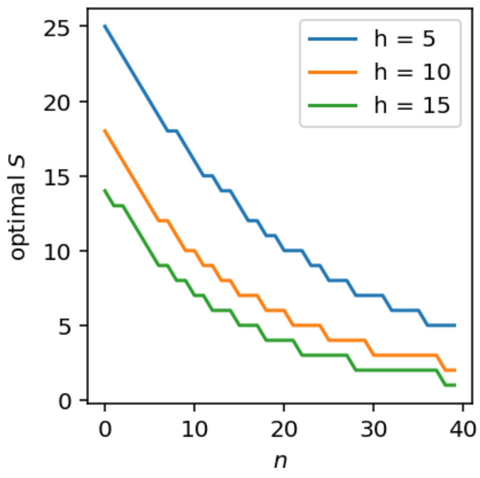
(c) Varying reward $R$.



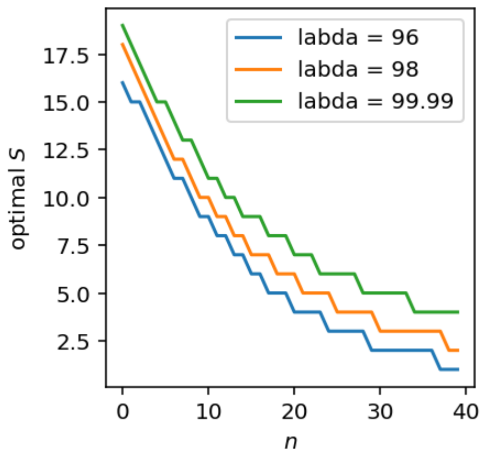
(d) Varying waiting cost $c$.

Figure 6: Sensitivity analysis for the producer's profit $Z(S)$.

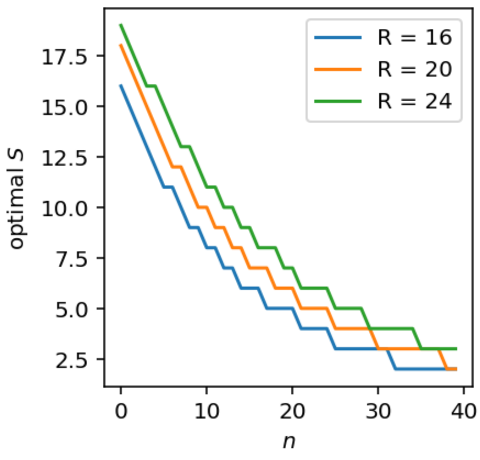
*Profit-Optimizing Target Inventory Level*
Consider the plots in Figure 7. As $n$ increases, the profit-maximizing target inventory level $S^*$ decreases. This is to be expected: given that customers are more prepared to wait in a queue, the producer is able to hold less items in inventory, paying less holding costs, while still meeting customer demand. Likewise, Subfigure 7a reveals that an increase in per-item holding cost decreases the profit-maximizing target inventory level $S^*$, suggesting that the increase in holding costs diminishes the benefit of maintaining inventory. Subfigure 7b shows that an increase in arrival rate increases $S^*$, suggesting that higher inventory levels can be utilized as a buffer against congestion.



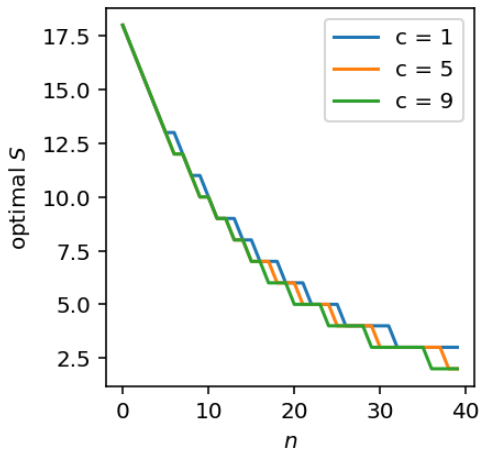
(a) Varying holding cost $h$.



(b) Varying arrival rate $\lambda$.



(c) Varying reward $R$.

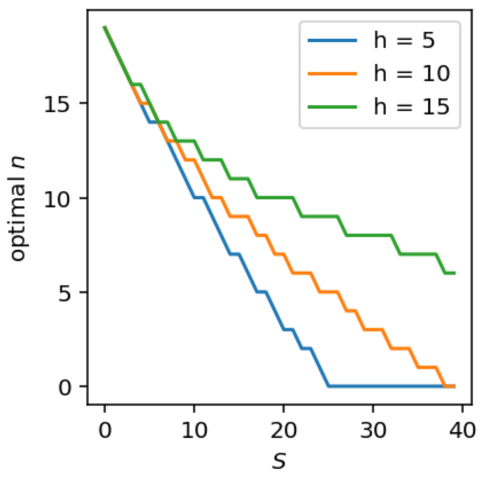


(d) Varying waiting cost $c$.

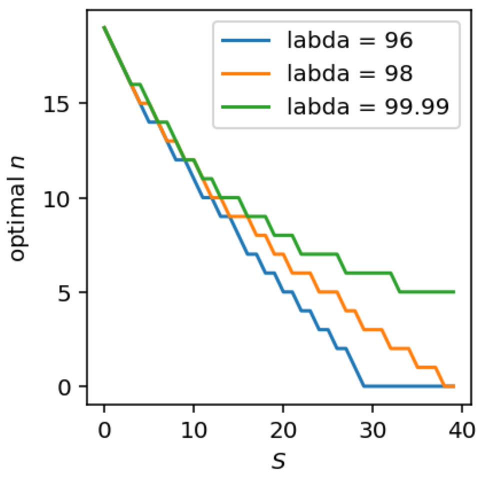
Figure 7: Sensitivity analysis for the profit-optimizing target inventory level $S^*$.

*Profit-Optimizing Joining Threshold*

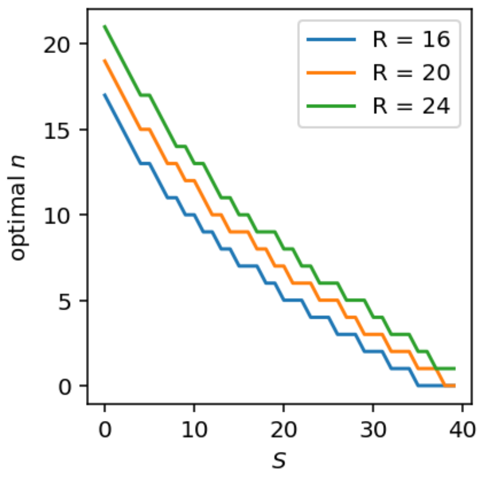
Consider the plots in Figure 8. As $S$ increases, the profit-maximizing joining threshold $n^*$ generally decreases. This is also to be expected: given that there are more products in inventory, there need be less customers in the queue to meet cusomer demand; the demand is partially satisfied by products in inventory. As per-item costs increase, $n^*$ increases as well, as can be seen in Subfigure 8a, which can be explained by an increase in queue length becoming more profitable than paying holding costs. Subfigure 8d suggests that customers tolerate a significantly higher queue length (as associated with an increase in $n^*$) as waiting cost decreases. For $c = 1$, $n^*$ even rises again as $S$ becomes large enough.



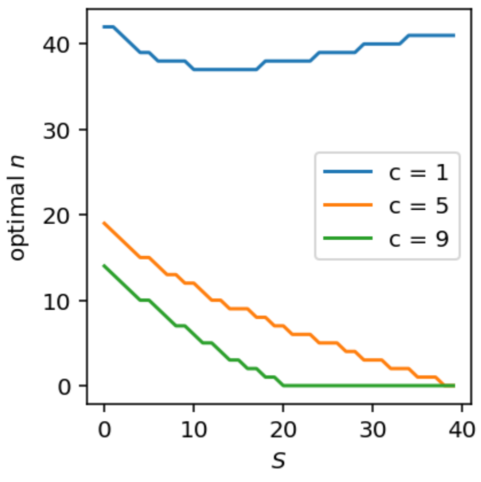
(a) Varying holding cost $h$.



(b) Varying arrival rate $\lambda$.



(c) Varying reward $R$.

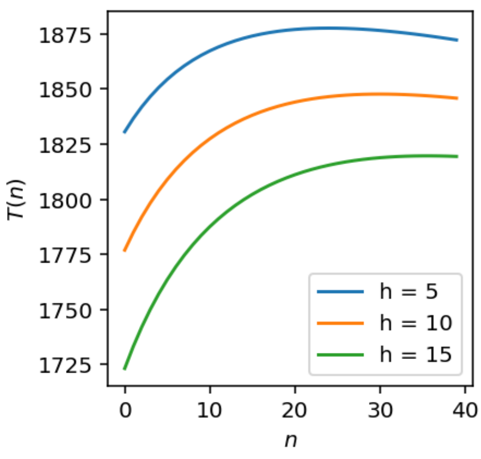


(d) Varying waiting cost $c$.

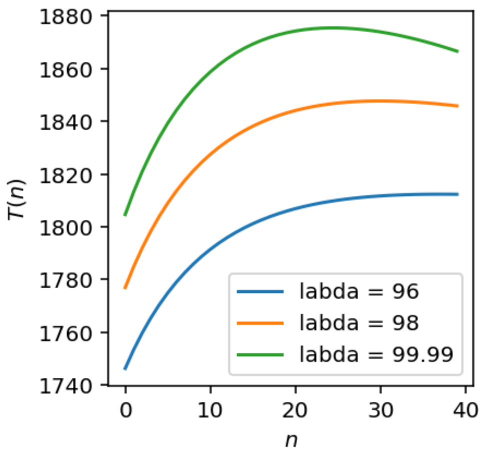
Figure 8: Sensitivity analysis for the profit-optimizing joining threshold $n^*$.

*Social Welfare as a Function of Joining Threshold*

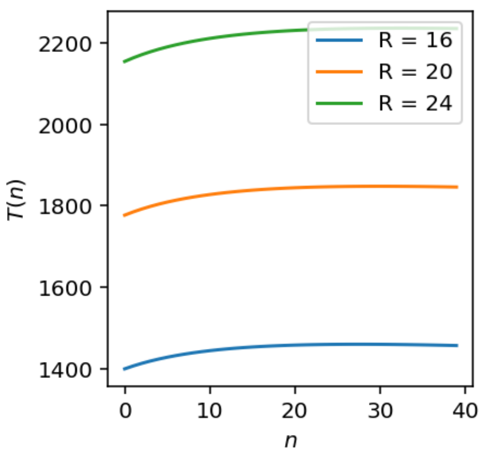
Consider the plots in Figure 9. We recognize the presence of an optimal joining threshold as found in Section 5.4.2. With increasing $n$, the producer's revenue increases (at no additional holding cost) but the customer's waiting costs increase as well, representing a trade-off for the social welfare.



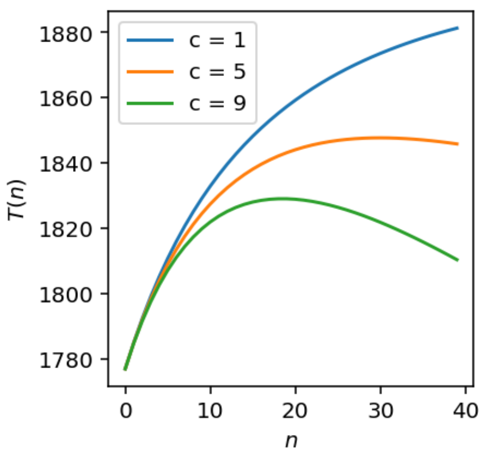
(a) Varying holding cost $h$.



(b) Varying arrival rate $\lambda$.



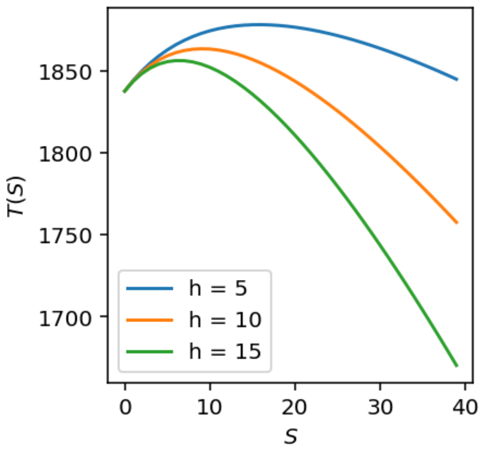
(c) Varying reward $R$.



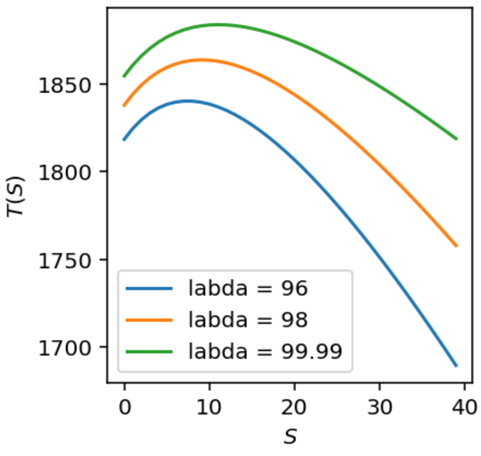
(d) Varying waiting cost $c$.

Figure 9: Sensitivity analysis for the social welfare $T(n)$.

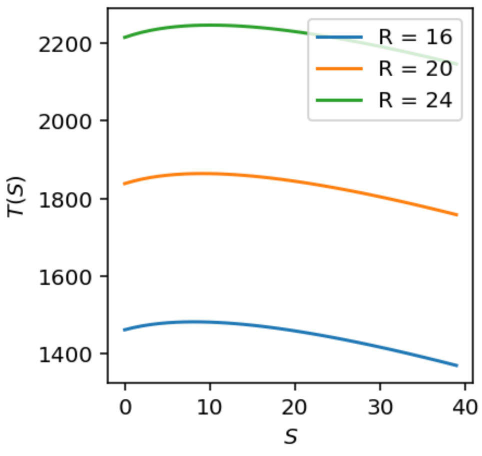
*Social Welfare as a Function of Target Inventory Level*
Consider the plots in Figure 10. We deduce the existence of a social-welfare-maximizing target inventory level $S^*$, though we did not prove its existence. As $S$ increases, customer's waiting costs decrease (due to the availability of inventory), but the producer's holding costs increase.



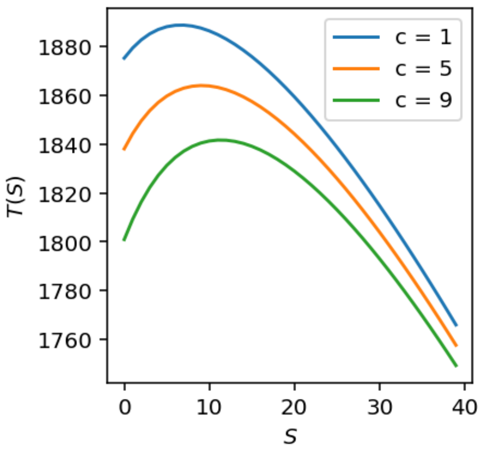
(a) Varying holding cost $h$.



(b) Varying arrival rate $\lambda$.



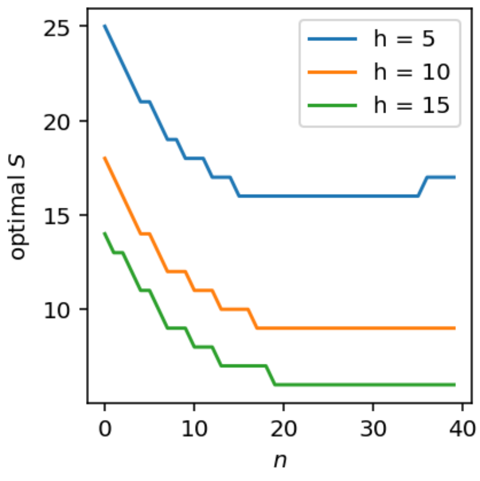
(c) Varying reward $R$.



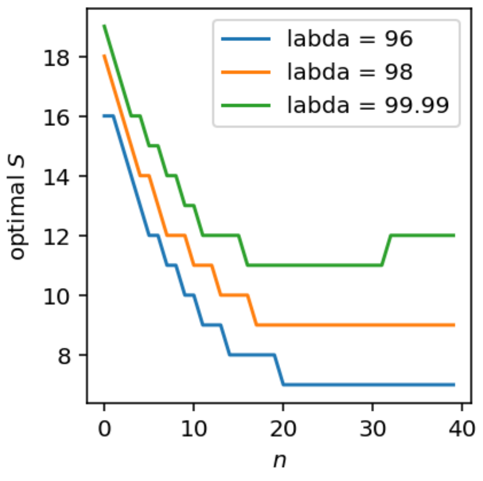
(d) Varying waiting cost $c$.

Figure 10: Sensitivity analysis for the social welfare $T(S)$.

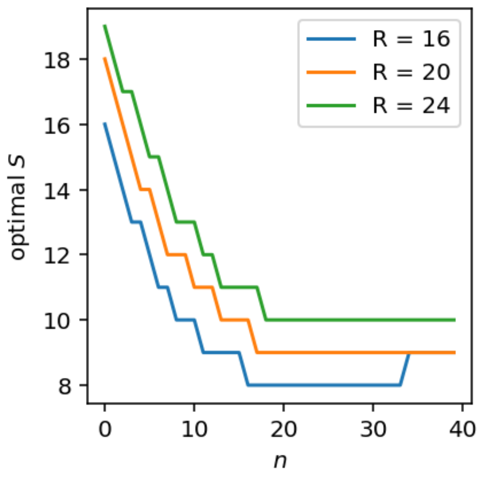
*Social-Welfare-Optimizing Target Inventory Level*
Consider the plots in Figure 11. As $n$ increases from 0, the social-welfare-maximizing target inventory level $S^*$ decreases at first, but at some point (as the maximum queue length, indicated by $n$, becomes too large) it becomes socially optimal to increase customer's benefit by *increasing* target inventory level at the expense of the producer's benefit through increased holding costs.



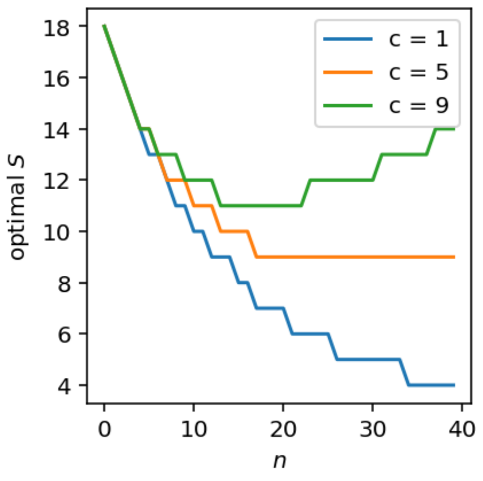
(a) Varying holding cost $h$.



(b) Varying arrival rate $\lambda$.



(c) Varying reward $R$.

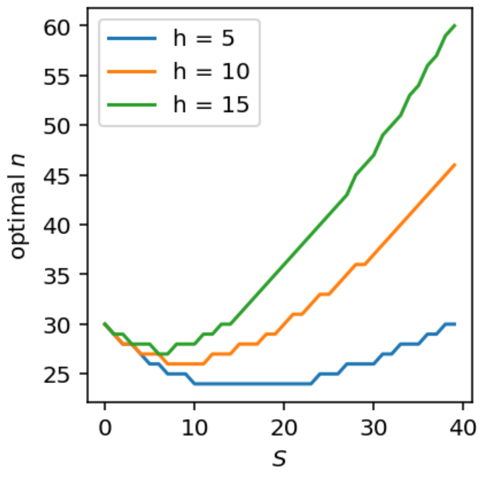


(d) Varying waiting cost $c$.

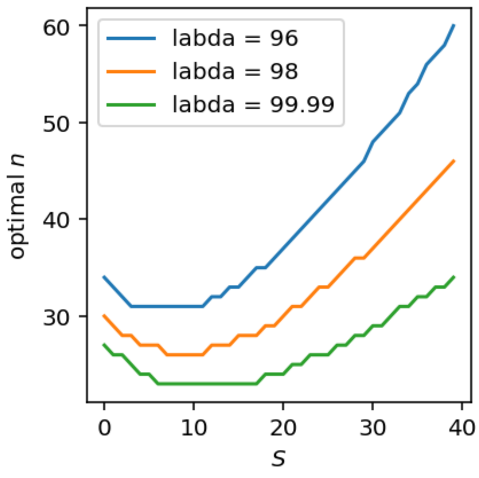
Figure 11: Sensitivity analysis for the social-welfare-optimizing target inventory level $S^*$.

*Social-Welfare-Optimizing Joining Threshold*

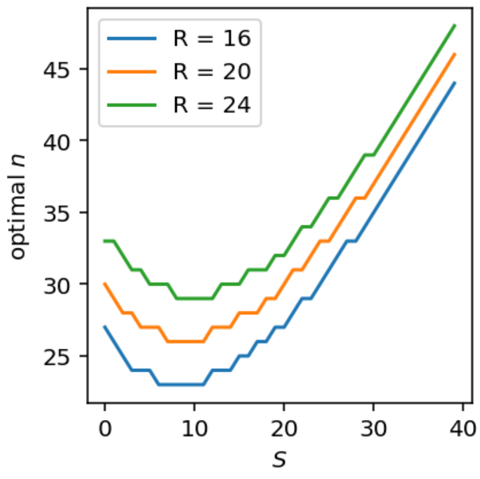
Consider the plots in Figure 12. Note that the social-welfare-optimizing joining thresholds $n^*$ are significantly higher than in the profit optimization case (see Figure 8).



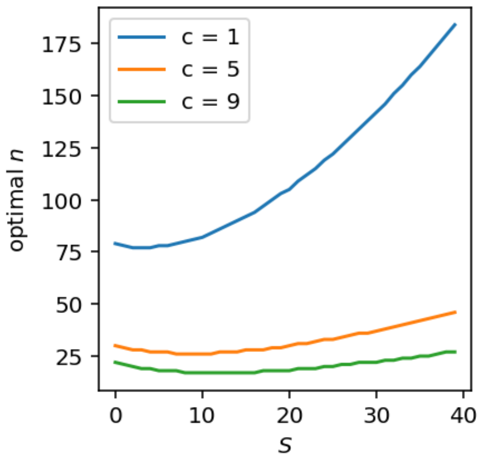
(a) Varying holding cost $h$.



(b) Varying arrival rate $\lambda$.



(c) Varying reward $R$.



(d) Varying waiting cost $c$.

Figure 12: Sensitivity analysis for the social-welfare-optimizing joining threshold $n^*$.

## 6 Description of the Two-Product Model

Our second model extends the previously discussed one-product model. We again consider a make-to-stock queueing system with strategic customers, now with *two* products instead of one. Specifically, we distinguish type-1 and type-2 products and system parameters corresponding to each type of product.

In this model, type-$t$ customers arrive according to a Poisson process with rate $\lambda_t > 0$. The singular server handles production at the service facility with production rate $\mu > 0$ and FCFS order *regardless of product type*. We assume that $\lambda_1 + \lambda_2 < \mu$, that is, $\rho = \frac{\lambda_1+\lambda_2}{\mu} < 1$.

For each type $t$, the producer sets an initially fixed target inventory level of $S_t$ products. When the inventory level of type $t$ falls short of $S_t$, the producer adds one type-$t$ replenishment job to the production queue. Hence, the producer sustains production (at rate $\mu$) for so long as the target inventory levels $S_1, S_2$ are not both met for their respective inventories. Each type-$t$ product in inventory incurs a holding cost of $h_t$ to be payed by the server. Additionally, the server charges to each joining type-$t$ customer a price of $p_t$.

If a type-$t$ customer decides to join, they receive reward $R_t > p_t$ upon service completion (receipt of a type-$t$ product). As in the one-product case, the presence of on-hand inventory results in immediate receipt of the product, departure, and addition of a replenishment job for the corresponding product type. If a type-$t$ customer joining the system ends up in the queue due to there being no inventory, they pay a waiting cost of $c_t$ per in-queue unit time and a type-$t$ replenishment job is added.

Again we make a distinction between an *unobservable* and an *observable* system, and assume that the target inventory levels $S_t$ are always known to customers.

# 7 Unobservable Two-Product Model

Kanavetas and Kosarevskaia[3] previously analyzed the case in which the two-product queueing system is unobservable to customers, meaning neither the amount of customers waiting nor the amount of products in inventory is known. The target inventory levels $S_t$ *are* known. This section will briefly go over this work. As in the one-product case we begin by solving the customer's problem before considering the producer's decisions based on customers' behavior.

## 7.1 Customer's Problem

The joining behavior of a type-$t$ customer can be described by joining probability $q_t \in [0,1]$. If we suppose that all customers adhere to strategy $(q_1, q_2)$, then it follows by the unobservability of the system that an arriving type-$t$ customer's decision to join or balk is informed by the *expected* utility $U_t(q_1, q_2)$ gained from joining. The expected waiting time for a type-$t$ customer $E[W_t(\bar{\lambda}_1, \bar{\lambda}_2, S_1, S_2)]$ for given *effective* arrival rates $\bar{\lambda}_1, \bar{\lambda}_2$ and target inventory levels $S_1, S_2$ has been shown to be

$$E[W_t(\bar{\lambda}_1, \bar{\lambda}_2, S_1, S_2)] = \frac{\bar{\lambda}_i^{S_i}}{(\mu - \bar{\lambda}_j)^{S_i}(\mu - \bar{\lambda}_1 - \bar{\lambda}_2)} \text{ where } j \neq t. \tag{14}$$

Hence, the expected utility function becomes

$$U_t(q_1, q_2) = R_t - p_t - c_t E[W_t(\lambda_1 q_1, \lambda_2 q_2, S_1, S_2)].$$

Two important properties for this function are its continuity over $[0,1]^2$ and it being strictly decreasing in both $q_1$ and $q_2$.

Equilibrium customer behavior is described by the so-called *Nash equilibrium*, which is here defined as a pair $(q_1^*, q_2^*)$ whose components are each other's best response, meaning that it satisfies

$$\begin{cases} q_1^* = \arg\max_{p\in[0,1]} p \cdot U_1(q_1^*, q_2^*), \\ q_2^* = \arg\max_{p\in[0,1]} p \cdot U_2(q_1^*, q_2^*). \end{cases}$$

This is well-defined as $(q_1^*, q_2^*)$ is the unique fixed point of the mapping

$$(\arg\max_{p\in[0,1]} p \cdot U_1(\cdot, q_2^*), \arg\max_{p\in[0,1]} p \cdot U_1(q_1^*, \cdot)).$$

We distinguish three cases.

- Suppose $S_1 = S_2 = 0$ and additionally $c_1/(R_1 - p_1) = c_2/(R_2 - p_2)$ as well as $c_1/(R_1 - p_1) < \mu < c_1/(R_1 - p_1) + \lambda_1 + \lambda_2$. Then there is a continuum of Nash equilibria $(q_1^*, q_2^*)$ satisfying $q_1^*\lambda_1 + q_2^*\lambda_2 = \mu - c_1/(R_1 - p_1)$.
- Suppose $S_1 = S_2 = 0$ and the additional requirements in the previous case do not hold. Then there exists a unique Nash equilibrium $(q_1^*, q_2^*)$, determinable by Table 2 in [3].
- If $S_1 > 0$ or $S_1 > 0$, there exists a unique Nash equilibrium $(q_1^*, q_2^*)$.

## 7.2 Producer's Profit Optimization

The producer's profit is equal to

$$\Pi(S_1, S_2) = p_1\lambda_1 q_1^{S_1,S_2} + p_2\lambda_2 q_2^{S_1,S_2} - h_1 E[I_1(S_1,S_2)] - h_2 E[I_2(S_1,S_2)]$$

with $q_t^{S_1,S_2}$ denoting the joining probability for a type-$t$ customer. It has been shown that

$$E[I_t(S_1,S_2)] = S_t - \frac{\lambda_t q_t^{S_1,S_2}}{\mu - \lambda_2 q_2^{S_1,S_2} - \lambda_2 q_2^{S_1,S_2}}\left(1 - \left(\frac{\lambda_t q_t^{S_1,S_2}}{\mu - \lambda_j q_j^{S_1,S_2}}\right)^{S_t}\right), j \neq t.$$

Furthermore, there exist natural bounds $\bar{S}_1, \bar{S}_2$ on the producer's inventory decisions. Targeting even higher target inventory levels does not improve profit. Hence, the optimal profit-maximizing target inventory level pair $(S_1^*, S_2^*)$ can be found as follows.

1. Calculate upper bounds $\bar{S}_1, \bar{S}_2$.
2. For each pair $(s_1, s_2)$ within these bounds.
   (a) Determine the Nash equilibrium joining probabilities $(q_1^{s_1,s_2}, q_2^{s_1,s_2})$.
   (b) Evaluate $\Pi(s_1, s_2)$.
3. Choose the pair yielding the highest profit.

## 7.3 Social Welfare Optimization

In this section we assume a social planner is able to modify the effective arrival rates $\bar{\lambda}_1, \bar{\lambda}_2$ as well as the target inventory levels $S_1, S_2$. The corresponding social welfare function is

$$T(\bar{\lambda}_1, \bar{\lambda}_2, S_1, S_2) = \bar{\lambda}_1 R_1 + \bar{\lambda}_2 R_2 - C(\bar{\lambda}_1, \bar{\lambda}_2, S_1, S_2).$$

where $\bar{\lambda}_t R_t$ represents the total reward received by type-$t$ customers and $C(\bar{\lambda}_1, \bar{\lambda}_2, S_1, S_2)$ captures the costs incurred by the producer (i.e. holding costs) as well as customers (i.e. waiting costs):

$$C(\bar{\lambda}_1, \bar{\lambda}_2, S_1, S_2) = \sum_{t=1}^{2}(h_t E[I_t(\bar{\lambda}_1, \bar{\lambda}_2)] + c_t \bar{\lambda}_t E[W_t(\bar{\lambda}_1, \bar{\lambda}_2)]),$$

where

$$E[I_t(\bar{\lambda}_1, \bar{\lambda}_2)] = S_t - \frac{\bar{\lambda}_t}{\mu - \bar{\lambda}_1 - \bar{\lambda}_2}\left(1 - \left(\frac{\bar{\lambda}_t}{\mu - \bar{\lambda}_j}\right)^{S_t}\right), j \neq t$$

and $E[W_t(\bar{\lambda}_1, \bar{\lambda}_2)])$ is as in (14).

For fixed $\bar{\lambda}_1, \bar{\lambda}_2$, optimal target inventory levels $(S_1^*, S_2^*)$ have been found explicitly, namely

$$S_t^* = \left\lceil \frac{\log(h_t/(h_t + c_t))}{\log(\bar{\lambda}_t/(\mu - \bar{\lambda}_j))} \right\rceil - 1, \text{ where } j \neq t.$$

The social-welfare-optimizing 4-tuple $(\bar{\lambda}_1^*, \bar{\lambda}_2^*, S_1^*, S_2^*)$ can be found as follows.

1. Establish upper bounds $\bar{S}_1, \bar{S}_2$ using

$$S_t < \left\lceil \frac{\log\left(h_t/(h_t + c_t)\right)}{\log\left(\bar{\lambda}_t/\mu\right)} \right\rceil .$$

2. For each possible inventory pair $(s_1, s_2)$ within these bounds:
   (a) Find the subdomain

   $$\mathcal{D} = \{(\bar{\lambda}_1, \bar{\lambda}_2) : 0 \leq \bar{\lambda}_t \leq \lambda_t, \bar{\lambda}_1 + \bar{\lambda}_2 < \mu\}$$

   for which holds $(S_1^*, S_2^*) = (s_1, s_2)$.

   (b) Within this subdomain, optimize effective arrival rates $(\bar{\lambda}_1, \bar{\lambda}_2)$.

   (c) Compute the resulting social welfare.

3. Choose the $(\bar{\lambda}_1^*, \bar{\lambda}_2^*, S_1^*, S_2^*)$ optimizing the social welfare.

# 8 Observable Two-Product Model

We now alter the two-product model by assuming that the system is *observable* to the customers. Analogously to the one-product case, this means that to an arriving customer the amount and types of customers waiting in the queue as well as the current amount and types of products in inventory are known, in addition to the target inventory levels $S_1, S_2$.

## 8.1 Customer's Problem

Since a customer's utility gained from joining is, as before, linearly dependent on the amount of customers in the queue, we conclude that the customers once again adopt a threshold strategy with joining thresholds $(n_1, n_2)$.

Suppose a customer of type $t$ arrives and encounters a non-empty queue. Naively, the average sojourn time would equal $\frac{i_1+i_2+1}{\mu}$ where $i_t$ is the amount of waiting type-$t$ customers. However, we must note that such a customer only waits for the departure of customers in the queue of the same type, bypassing the customers of the other type. On the other hand, the arrival of these other-type customers certainly influenced the production queue, leading to a longer waiting time for our customer.

For example, suppose that due to a surge in arriving customers both inventory types are empty, and that the producer is working on the production of a type-1 product. If a customer of type 2 arrives, they certainly have to wait on the production of this type-1 product. In fact, an arriving type-$t$ customer's expected waiting time is $\frac{m_t}{\mu}$, where $m_t$ is their place in the *production* queue.

This complicates our analysis greatly as it is no longer immediately clear how to define states such that the Markov property holds. Indeed, for any description of the system state involving a distinction between the product types, transitions are dependent on the production queue, which is in turn dependent on the history of arriving customers. For example, the waiting time of an arriving customer depends on their place in the production queue.

### 8.1.1 States

We define a state space in which each state consists of the full *current* production queue as a vector with elements in $\{1, 2\}$. For example, $(1, 2, 2)$ denotes a production queue in which the producer is currently producing a type-1 product before the production of 2 type-2 products. Note that transitions between states correspond directly to the arrival of customers and the fulfillment of production, and that the length of the state vectors is bounded by $n_1 + n_2 + S_1 + S_2$.

More precisely, for state $v$ we have

$$(v_1, v_2, \dots, v_n) \to \begin{cases} (v_1, v_2 \dots, v_n, 1) & \text{with rate } \lambda_1, \text{ if } n < n_1, \\ (v_1, v_2 \dots, v_n, 2) & \text{with rate } \lambda_2, \text{ if } n < n_2, \\ (v_2, \dots, v_n) & \text{with rate } \mu. \end{cases}$$

Moreover, since no type-$t$ customer joins if there are $n_t$ customers of the same type in the queue, and there can be at most $S_t$ replenishment jobs for type $t$, the length of any state is bounded by $n_1 + n_2 + S_1 + S_2$.

Note that this chain is irreducible and positive recurrent (since $\rho < 1$), so a stationary distribution $\boldsymbol{\pi}$ exists and is unique.

### 8.1.2 Customer's Strategy

Let $v = (v_1, \dots, v_n)$ be a state. Then the place in the production queue for an arriving type-$t$ customer, i.e. $m_t(v)$, can be found by Algorithm 2.

**Algorithm 2** Place in the production queue $m_t(v)$.

```
Require: $v = (v_1, \dots, v_n), t, S_1, S_2$
  $m \leftarrow n + 1$
  $s \leftarrow 0$
  while $s < S_t$ do
      $m \leftarrow m - 1$
      if $v_m = t$ then
          $s \leftarrow s + 1$
      end if
  end while
  $m_t(v) \leftarrow m$
```

Indeed, $m_t(v)$ equals the position of the $S_t$-th to last type-$t$ order in the production queue. For example, consider the situation in which $S_1 = 2, S_2 = 3$ and an arriving customer encounters production queue

$$v = (1, 2, 1, 1, 2, 2, 1, 2, 2),$$

then $m_1(v) = 4$ and $m_2(v) = 6$.

It follows that both $m_1(\cdot)$ and $m_2(\cdot)$ induce a total order on the set of possible production queues. As the expected waiting time for a type-$t$ customer having position $m_t(v)$ given production queue $v$ is equal to

$$E[W_t(m_t(v))] = \frac{m_t(v)}{\mu},$$

it follows that we also find that the expected utility for a joining type-$t$ customer encountering queue $v$, being given by

$$\begin{aligned} U_t(v) &= R - p - cE[W_t(m_t(v))] \\ &= R - p - \frac{cm_t(v)}{\mu} \end{aligned}$$

induces a total ordering $\leq_U$ on the set of possible production queues. From this, we can find threshold policy $(n_1, n_2)$: for type-$t$ customers there exists a $v_t^*$ such that $U_t(v) \geq 0$ for $v \leq_U v_t^*$, meaning that the customer joins, and $U(i) < 0$ for $v >_U v_t^*$, meaning that the customer balks.

## 8.2 Complications Regarding Producer's Profit and Social Welfare

Our description of states, necessitated by the complex relation between queue description and waiting time, complicates the analysis, and indeed, optimization, of the producer's profit and social welfare. In Section 5 we expressed these quantities in terms of effective arrival rate, expected amount of products in inventory and expected amount of customers in the system. The calculation of each of these metrics involved the stationary distribution of the system. However, in the observable two-product model we are now considering, such stationary distribution is difficult to find by the exponentially large state space.

# 9 Discussion

## 9.1 Results

Apart from the reviews of the previous work on the unobservable one-product and two-product models, we have found the following results.

- For the observable one-product make-to-stock queueing system with strategic customers:
  - the producer's profit increases indefinitely with the target inventory level $S$ given that there are no holding costs. If holding costs are non-zero, then we can exploit the fact that in practice an upper bound for feasible target inventory levels exists and calculate profit-maximizing $S^*$ in polynomial time;
  - there exists profit-maximizing joining threshold $n^*$, corresponding to profit-maximizing price $p^*$;
  - the social welfare increases indefinitely with the target inventory level $S$ given that there are no holding costs. If holding costs are non-zero, then we can exploit the fact that in practice an upper bound for feasible target inventory levels exists and calculate social-welfare-maximizing $S^*$ in polynomial time;
  - there exists social-welfare-maximizing joining threshold $n^*$, corresponding to social-welfare-maximizing price $p^*$;
  - we found that our numerical analysis reflected the expected behavior of the system.
- For the observable two-product make-to-stock queueing system with strategic customers, we described the subtleties of the customer's problem and touched upon the complexity of analyzing the producer's profit and social welfare.

## 9.2 Future Work

The following directions for future work on make-to-stock queueing systems with strategic customers are suggested.

- Study the optimization of the producer's profit and social welfare for the observable two-product make-to-stock queueing system.
- Extend the analysis of unobservable or observable make-to-stock queueing systems to settings in which there are $n > 2$ types of product.
- Repeat the analysis of the one-product or two-product queueing systems with *partial* observability, e.g. only the amount of customers or only the current inventory level is known to arriving customers.

## References

[1] John A Buzacott and J George Shanthikumar, *Stochastic models of manufacturing systems*, Prentice Hall International, 1993.

[2] Refael Hassin and Moshe Haviv, *To queue or not to queue: Equilibrium behavior in queueing systems*, vol. 59, Springer Science & Business Media, 2003.

[3] Odysseas Kanavetas and Ekaterina Kosarevskaia, *Two-product make-to-stock system: Strategic joining and optimal inventory levels*, preprint.

[4] David G Kendall, *Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain*, The Annals of Mathematical Statistics (1953), 338–354.

[5] Pinhas Naor, *The regulation of queue size by levying tolls*, Econometrica: journal of the Econometric Society (1969), 15–24.

[6] Can Öz and Fikri Karaesmen, *On a production/inventory system with strategic customers and unobservable inventory levels*, SMMSO 2015 (2015), 161.

[7] Michael H Veatch and Lawrence M Wein, *Optimal control of a two-station tandem production/inventory system*, Operations Research **42** (1994), no. 2, 337–350.

## A Python code (calc.py)

```
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = [3,3.1]
plt.rcParams['savefig.bbox'] = 'tight'

# product price
def p(n):
    return R-n*c/mu

# producer's profit
def Z(n,S):
    rho = labda/mu
    A = 1/(1-rho**(n+S+1))
    B = labda*R*(1-rho**(n+S))
    nu = R*mu/c
    H = h*(S*(1-rho)-rho*(1-rho**S))/(1-rho)
    return A*(B*(nu-n)/nu-H)

# social welfare
def T(n,S):
    rho = labda/mu
    A = 1/(1-rho**(n+S+1))
    B = 1-rho**(n+S)
    nu = labda*R
    C = c*rho**(S+1)*(1-(n+1)*rho**n+n*rho**(n+1))/(1-rho)
    H = h*(S*(1-rho)-rho*(1-rho**S))/(1-rho)
    return A*(nu*B-C-H)

# f-optimal (n*,S*) for n in n_vals and S in s_vals
def find_opt_n_S(f,n_vals,S_vals):
    len_n = len(n_vals)
    len_S = len(S_vals)
    f_vals = np.zeros((len_n,len_S))
    for i in range(len_n):
        for j in range(len_S):
            f_vals[i][j] = f(n_vals[i],S_vals[j])
    max_n_ind,max_S_ind = np.unravel_index(f_vals.argmax(),f_vals.shape)
    max_n = n_vals[max_n_ind]
    max_S = S_vals[max_S_ind]
    max_p = p(max_n)
    max_Z = Z(max_n,max_S)
    max_T = T(max_n,max_S)
    return(max_n,max_S,max_p,max_Z,max_T)

# f-optimal n* given S
def find_opt_n(f,S):
    n = 0
    while True:
        if p(n) < 0: raise Exception("Price negative")
        f0 = f(n,S)
        f1 = f(n+1,S)
        if f0 > f1: # only check f(n+1)<f(n) since opt. n* is unique
            return(round(f0,2),n,round(p(n),2))
        n = n+1
    return(round(f0,2),n,round(p(n),2))

# f-optimal S* given n
def find_opt_S(S_max,f,n):
    if p(n) < 0: raise Exception("Price negative")
    S_vals = range(S_max + 1)
    S_opt = np.argmax([f(n,S) for S in S_vals])
    f0 = f(n,S_opt)
    return(round(f0,2),S_opt)

def plt_Zn(n_vals, sens_var = None, sens_vals = [None]):
    plt.figure()
    plt.xlabel('$\it{n}$')
    plt.ylabel('$\it{Z(n)}$')
    sens_var_init = globals()[sens_var]
    for val in sens_vals:
        globals()[sens_var] = val
        plt.plot(n_vals, [Z(n,S) for n in n_vals],
                 label = str(sens_var) + " = " + str(val))
    globals()[sens_var] = sens_var_init
    plt.legend()
    plt.savefig("img/" + "Zn_" + str(sens_var) + ".png")

def plt_ZS(S_vals, sens_var = None, sens_vals = None):
    plt.figure()
    plt.xlabel('$\it{S}$')
    plt.ylabel('$\it{Z(S)}$')
    sens_var_init = globals()[sens_var]
    for val in sens_vals:
        globals()[sens_var] = val
        plt.plot(S_vals, [Z(n,S) for S in S_vals],
                 label = str(sens_var) + " = " + str(val))
    globals()[sens_var] = sens_var_init
    plt.legend()
    plt.savefig("img/" + "ZS_" + str(sens_var) + ".png")

def plt_ZoptS(n_vals, sens_var = None, sens_vals = None):
    plt.figure()
    plt.xlabel('$\it{n}$')
    plt.ylabel('optimal $\it{S}$')
    sens_var_init = globals()[sens_var]
    for val in sens_vals:
        globals()[sens_var] = val
        plt.plot(n_vals, [find_opt_S(S_max,Z,n)[1] for n in n_vals],
                 label = str(sens_var) + " = " + str(val))
    globals()[sens_var] = sens_var_init
    plt.legend()
    plt.savefig("img/" + "ZoptS_" + str(sens_var) + ".png")
```

```
def plt_Zoptn(S_vals, sens_var = None, sens_vals = None):
    plt.figure()
    plt.xlabel('$\it{S}$')
    plt.ylabel('optimal␣$\it{n}$')
    sens_var_init = globals()[sens_var]
    for val in sens_vals:
        globals()[sens_var] = val
        plt.plot(S_vals, [find_opt_n(Z,S)[1] for S in S_vals],
                 label = str(sens_var) + "␣=␣" + str(val))
    globals()[sens_var] = sens_var_init
    plt.legend()
    plt.savefig("img/" + "Zoptn_" + str(sens_var) + ".png")

def plt_Tn(n_vals, sens_var = None, sens_vals = None):
    plt.figure()
    plt.xlabel('$\it{n}$')
    plt.ylabel('$\it{T(n)}$')
    sens_var_init = globals()[sens_var]
    for val in sens_vals:
        globals()[sens_var] = val
        plt.plot(n_vals, [T(n,S) for n in n_vals],
                 label = str(sens_var) + "␣=␣" + str(val))
    globals()[sens_var] = sens_var_init
    plt.legend()
    plt.savefig("img/" + "Tn_" + str(sens_var) + ".png")

def plt_TS(S_vals, sens_var = None, sens_vals = None):
    plt.figure()
    plt.xlabel('$\it{S}$')
    plt.ylabel('$\it{T(S)}$')
    sens_var_init = globals()[sens_var]
    for val in sens_vals:
        globals()[sens_var] = val
        plt.plot(S_vals, [T(n,S) for S in S_vals],
                 label = str(sens_var) + "␣=␣" + str(val))
    globals()[sens_var] = sens_var_init
    plt.legend()
    plt.savefig("img/" + "TS_" + str(sens_var) + ".png")

def plt_ToptS(n_vals, sens_var = None, sens_vals = None):
    plt.figure()
    plt.xlabel('$\it{n}$')
    plt.ylabel('optimal␣$\it{S}$')
    sens_var_init = globals()[sens_var]
    for val in sens_vals:
        globals()[sens_var] = val
        plt.plot(n_vals, [find_opt_S(S_max,T,n)[1] for n in n_vals],
                 label = str(sens_var) + "␣=␣" + str(val))
    globals()[sens_var] = sens_var_init
    plt.legend()
    plt.savefig("img/" + "ToptS_" + str(sens_var) + ".png")
```

```
def plt_Toptn(S_vals, sens_var = None, sens_vals = None):
    plt.figure()
    plt.xlabel('$\it{S}$')
    plt.ylabel('optimal $\it{n}$')
    sens_var_init = globals()[sens_var]
    for val in sens_vals:
        globals()[sens_var] = val
        plt.plot(S_vals, [find_opt_n(T,S)[1] for S in S_vals],
                 label = str(sens_var) + " = " + str(val))
    globals()[sens_var] = sens_var_init
    plt.legend()
    plt.savefig("img/" + "Toptn_" + str(sens_var) + ".png")

S_max = 1000
h = 10
labda = 98
mu = 100
R = 20
c = 5
S = 20
n = 20

n_vals = range(40)
S_vals = range(40)

print(find_opt_n_S(Z,n_vals,S_vals))
print(find_opt_n_S(T,n_vals,S_vals))

plt.figure(figsize=[3,4])
plt.xlabel('$\it{n}$')
plt.plot(n_vals, [Z(n,S) for n in n_vals],label = '$\it{Z(n)}$')
plt.plot(n_vals, [T(n,S) for n in n_vals],label = '$\it{T(n)}$')
plt.legend(loc='lower left')
plt.savefig("img/" + "base_n" + ".png")


plt.figure(figsize=[3,4])
plt.xlabel('$\it{S}$')
plt.plot(S_vals, [Z(n,S) for S in S_vals],label = '$\it{Z(S)}$')
plt.plot(S_vals, [T(n,S) for S in S_vals],label = '$\it{T(S)}$')
plt.legend(loc='lower left')
plt.savefig("img/" + "base_S" + ".png")

for var_vals in [
        [ 'h', [5,10,15] ],
        [ 'labda', [96,98,99.99] ],
        [ 'R', [16,20,24] ],
        [ 'c', [1,5,9] ],
        ]:
    sens_var = var_vals[0]
    sens_vals = var_vals[1]
```

```
plt_Zn(n_vals, sens_var, sens_vals)
plt_ZS(S_vals, sens_var, sens_vals)
plt_ZoptS(n_vals, sens_var, sens_vals)
plt_Zoptn(S_vals, sens_var, sens_vals)
plt_Tn(n_vals, sens_var, sens_vals)
plt_TS(S_vals, sens_var, sens_vals)
plt_ToptS(n_vals, sens_var, sens_vals)
plt_Toptn(S_vals, sens_var, sens_vals)
```