Aalto University
School of Business

Three models of trendy goods

Economics

Master's thesis

Ville Mäkinen

2013

Department of Economics
Aalto University
School of Business

Three models of trendy goods

Ville Mäkinen

May 30, 2013

Abstract

Objectives: The objective of this thesis is to build models where the demand of a product has an explicit, recognizable trend effect. With the models we seek to infer the properties of a trend based on observable firm behavior.

Methods: I construct in total three different models: a monopoly model, a Stackelberg duopoly model and a simultaneous-choice duopoly model. Moreover, I study three different trend types in each of the models: a nonexistent trend, a linear trend and a parabel trend. The demand in each of the models is determined through initial value problems that incorporate the trends into the probabilities for buying a product. All presented initial value problems are modified versions of the initial value problems of the Susceptible-Infective-Susceptible epidemiological model. I am unable to find analytical solutions for the initial value problems, and so I use numerical methods for solving them. The numerical solutions are done with MATLAB. The use of numerical solutions for the initial value problems means that also the firm behavior is solved numerically. The firms seek to maximize their profits by choosing the price, quality and free samples of their products. I limit the firms' choices to finite choice sets in order to be able to optimize using simple brute force. Brute force is used to calculate the firms' profits for every possible choice combination from the finite choice sets. This allows us to describe firm behavior either directly (the monopoly model and the Stackelberg duopoly model) or by studying the subgame-perfect Nash equilibrium of an interaction game that the firms play (the simultaneous-choice duopoly model).

Results: We are not able to deduce the properties of a trend solely by observing firm behavior. In each model the values for exogenous variables may be chosen such that firm behavior is the same for different trend types. We may, however, formulate an existence result for trends: If we observe at least one firm giving out free products, we may infer the existence of a trend. The result does not run in the other direction, i.e. we may not infer the nonexistence of a trend if we observe no firm giving out free products.

Keywords: trend, Susceptible-Infective-Susceptible, subgame-perfect Nash equilibrium, duopoly

Contents

List of Figures

List of Tables

1

Introduction

Trends are widely accepted to exist in consumer markets. It is also widely accepted that not all products are trendy. This thesis is an attempt to build models of trendy products to study the existence of trends. I propose to do this through using models derived from the epidemiological Susceptible-Infective-Susceptible (SIS) model (see e.g. Brauer and Castillo-Chávez (2001, p. 411)). Specifically, I adopt the SIS model to describe the demand of a product. I study three different models in total: 1. a monopoly model, 2. a Stackelberg duopoly model and 3. a simultaneous-choice duopoly model. While the modeling approach allows observing the market state (i.e. who owns what when) explicitly, I study the existence of trends through firm behavior. For example, if we were studying whether a particular pair of jeans is trendy or not, it would be clearly much easier to observe the behavior of the firm that makes the jeans rather than how many consumers own a pair at a particular point in time.

I argue that the use of an epidemiological model is appropriate when we note the similarities between having a disease and having a product. Both are measurable binary states, i.e. you either have or do not have a disease or a product. Both products and diseases have finite lifetimes. When there are many infected persons in a population, the likelihood of becoming infected ourselves increases ceteris paribus. Likewise, when there are many persons with a certain product, arguably the likelihood of buying that product should increase. For example, we could infer that a product is good from the product's popularity. Thus we might reasonably ask ourselves whether the dynamics behind epidemics are similar to the dynamics behind the demand for trendy goods.

The evolution of the market, and thereby demand, in each model is described by an initial value problem inspired by the SIS model. The initial value problems are unfortunately analytically unwieldy, which then leads to the use of numerical methods for solving them. The use of numerical methods places demands on the way that the firms' decision variables, i.e. the price, quality and free samples of their products, are endogenized. I assume that the models' firms are only interested in maximizing their profits. This allows us to determine the firms' behavior by determining what choices maximize the firms' profits. I assume that the firms' choices are limited to finite choice sets to help with solving the maximization problems and to guarantee the existence of solutions in the duopoly models. I acknowledge that the use of finite choices sets is undesirable as it requires us to "define the range of possible endogenous behavior exogenously" beforehand. MATLAB was used for the numerical numerical solutions.

Consumer behavior is simple and uniform across the proposed models. I assume that consumers arrive at buying decisions, i.e. visit a store, based on the Poisson process. The consumer's buying decision is determined through the consumer's indirect utility function such that the consumer makes the decision that maximizes his indirect utility. The indirect utility function is defined over the possible options of the buying decision, i.e. whether to buy the product of some firm or to buy no product at all. The indirect utility of an option is dependent on the quality and the price of the product it represents (the "quality and price" of not buying any product are assumed to be equal to zero). Moreover, I assume that there is a trend effect in the indirect utility function for options representing products. The strength of the trend effect is dependent on how many others own the product that the option represents. Finally, there is a random variable in the indirect utility to account for the quirks of human behavior. The random variable leads us to study the probabilities that the consumer buys a particular product.

I use the following definition for a trend: A product is trendy, or a trend exists, if the probability of buying the product is in any way dependent on how many others own the product at the time of the buying decision. Otherwise the product is not trendy, or no trend exists. This definition of a trend is very general and allows for a multitude of different trend types. I consider three different trend types in this thesis: 1. the "no trend" trend type, 2. a linear trend type and 3. a parabel trend type. With the "no trend" trend type, there is no such effect to the indirect utility of a product that could be interpreted as stemming from a trend. Firm behavior with the "no trend" trend type gives us a benchmark to which we may compare the firm behavior stemming from the two other trend types. With the linear trend type, the indirect utility of the product increases linearly as the number of people owning the product increases. With the parabel trend type, the indirect utility of the product increases up to some saturation point and then starts to decrease forming a downward-opening parabel as the number of people owning the product increases.

The linear and parabel trend types may be interpreted through well-known concepts in economics. Products with the linear trend type can be seen as examples of network goods. The telephone may serve as an example of a network good, as the usefulness of a telephone increases as the number of other people with telephones increases since now there are more people one may call. Next, products with parabel trend type can be seen as examples of "trends in the traditional sense" or of negative externalities in consumption. Examples for products with this trend type could be clothing or cars. A piece of clothing may start to lose its appeal when too many others already have the same piece, leading to behavior typically associated with the notion of a trend. Similarly the usefulness of a car decreases when there is congestion, i.e. too many cars already on the road, which in turn could be interpreted as a negative externality in consumption.

I pose the following research questions:

1. Do different trend types lead to observable differences in firm behavior?
2. In particular, are there markets where firms find it optimal to give out free products as samples?
3. Can we deduce the type and the strength of a trend based solely on observable firm behavior?

The research questions are answered by looking at firm behavior which is solved numerically in each of the three proposed models. I vary a number of exogenous variables to provide sensitivity analysis on how the exogenous variables affect firm behavior in each model. These exogenous variables are the planning horizon and the discount rate of firms and the strength of the trend. Each of these variables is difficult to observe, and each plays a large part in determining the profits of a firm and thereby the behavior of the firm.

The thesis is structured as follows: Chapter 2 describes the assumption behind consumer behavior in all three models. Chapter 3 describes the necessary assumptions and the methods for solving the initial value problems (and the firms' profits) numerically. Chapter 4 presents the monopoly model, chapter 5 the Stackelberg duopoly model and chapter 6 the simultaneous-choice duopoly model. Chapter 7 summarizes the work, criticizes the models and proposes some possible extensions to them.

2

Consumer behavior

This chapter describes the assumptions regarding consumer behavior in each of the proposed models. These assumptions form the basis for writing the initial value problems that describe the evolution of the markets. The assumptions are chosen such that they mimic the assumptions of the SIS model and can be interpreted in terms of economics. Consumer behavior is interpreted in terms of how a consumer moves between so-called consumer compartments. The consumer compartments describe what products a consumer in a particular compartment owns. The movement between consumer compartments is due to either a consumer purchasing a product or a product breakdown. The movement between compartments then determines the demand for the products through the purchases.

The rate of movement between compartments depends on three factors: 1. how often a consumer will face a buying decision, 2. how the consumer makes his buying decision and 3. how long do the the products last. These factors are determined by the following assumptions: Consumers face buying decisions according to a Poisson process. Consumers make their buying decisions by choosing the option in the buying decision that maximizes their indirect utility. Finally, the product lifetimes are assumed to have an exponential distribution.

The buying decisions are modeled with the multinomial logit model of discrete choice (see e.g. Anderson et al. (1992, p. 39)). Consumer behavior at the individual level is assumed be non-deterministic as the consumer's indirect utility function has a random variable by assumption. The consumer makes his choices to maximize his indirect utility, which is in part determined by the realization of the random variable. As the realizations vary, we may merely derive the buying probabilities for particular products. The buying probabilities depend on the trend effects, the qualities and the prices of the products.

While the modeling approach for the buying probabilities is well-established for describing how a consumer without any products might buy a product, the duopoly models require us to describe the buying probabilities for a particular product for consumers who already own a (similar) product. I was unable to find prior research on this issue and was therefore forced to make my own assumptions to handle it. A critical reader should note these assumptions.

2.1 The compartmental view of the market

Let us suppose that we have a set of characteristics and a population where individuals exhibit these characteristics. For epidemiological models, including the SIS model, these characteristics would describe whether an individual has a particular disease. For the proposed models, the characteristics describe whether an individual owns a particular product. The population is then divisible into compartments depending on the characteristics such that each compartment represents mutually exclusive combination of the characteristics. For example, if we consider an epidemiological model for a single disease, the population would be divided into two compartments representing those who have the disease and those who are healthy. Further, if we consider an epidemiological model for two diseases, the population would be divided into four compartments representing those who are healthy, those who have either one of the diseases and those who have both diseases. We may naturally divide the population similarly in the proposed models based on what products the consumers own.

Let us next limit our attention to the compartmental structures of the proposed models and name the models' compartments. I use the naming notation of the SIS model for the compartments of the proposed models. The SIS model names the compartment of those who are susceptible as compartment S and the compartment of those who are infective as compartment I. Following this notation, we name the compartment of those who do not own any product as compartment S and the compartments of those who own some particular product as compartments I. In the monopoly model there is only a single I-compartment as the consumers may own only the monopoly's product. In the duopoly models there are in total three I-compartments where the subscript of each compartment describes what the consumers in that compartment own. Let us name the duopoly firms as firm i and firm j. Then the I-compartments are named I_i, I_j and I_{ij} to denote that the consumers own the product of firm i or j or both firms.

Why is it necessary to present such complicated tools to model demand instead of, for example, defining an explicit demand function? The compartmental view of the market provides us with a tractable way to model the effects of trends to demand. We aim to find functions $S(t)$ and $I(t)$ (or functions $I_i(t)$, $I_j(t)$ and $I_{ij}(t)$ in the duopoly models) through studying initial value problems to describe how many consumers are at which compartment at time t. As the functions describe how many consumers own a particular product, we may model the effects of trends to the product's buying probability with them.

2.2 Arrivals to buying decision and product lifetime

Let us now describe the assumptions regarding the consumer arrivals and the product breakdowns. These assumptions are made on the basis of the discussion in Brauer and Castillo-Chávez (2001, p. 351-2) concerning the background assumptions of the SIR model. The background assumptions of the SIR model are equivalent with the background assumptions of the SIS model as the difference between models concerns only what happens to a person at the end of the disease. We therefore wish to formulate similar assumptions regarding the consumer arrival and product breakdown rates. From these assumptions we wish to show that the consumer arrivals and product breakdowns happen at a constant rate per unit time. This means that given an infinitesimally small time interval, the probability of a single consumer arrival or a single product breakdown is independent of time and approximately proportional to the length of the time interval with some rate parameter. The rate parameter is determined by the chosen background assumptions in both cases.

I assume that consumer optimization plays no part in determining consumer arrivals or product lifetimes. The consumers cannot therefore affect the product lifetime by, for example, varying the product's rate of use. The product lifetime is then based solely on the durability of the product, which allows us to assume that lifetime of a product has an exponential distribution. Next, I see the consumer arrivals as incidental and offer the following interpretation for a consumer arrival: Consumers visit a store on other business, happen to pass by a window exhibition and make the decision whether to buy based on the viewed window exhibition. The consumer arrival rate is then constant and independent of the consumer's compartment. This allows us to model the consumer arrivals as arising from a Poisson process.

2.2.1 Arrivals to buying decision

Let us now consider the arrivals to buying decisions. I model the consumer arrivals as a Poisson process $\{N(t) : t \geq 0\}$ with parameter β, where the process $\{N(t) : t \geq 0\}$ is a sequence of random variables $N(t)$. I use the definition of Feldman and Valdez-Flores (2010, p. 116) for a Poisson process: Process $\{N(t) : t \geq 0\}$ is a Poisson process with parameter β if 1. $P(N(t) = k) = \exp(-\beta t)(\beta t)^k/k!$ for all $k \geq 0, k \in \mathbb{N}$ and $t \geq 0$, 2. the event $N(s+u) - N(s) = i$ is independent of the event $N(t) = j$ if $t < s$, and 3. the probability $P(N(s+u) - N(s) = i)$ depends only on the value of u. Let us now consider the number of arrivals on a short time interval $[t, t+\Delta]$ where time t is some arbitrary starting time and length Δ a small positive constant. Because a Poisson process has independent (condition 2 of the definition) and stationary (condition 3 of the definition) increments (Feldman and Valdez-Flores (2010, p. 117)), the probability that k consumers arrive on time interval $[t, t+\Delta]$ is given by equation (2.1):

$$P(N(t+\Delta) - N(t) = k) = \exp(-\beta\Delta)\frac{(\beta\Delta)^k}{k!} = P(N(\Delta) = k) \tag{2.1}$$

From equation (2.1) we note that the probability that k consumers arrive on time interval $[t, t+\Delta]$ is independent of the starting time t, and that it is therefore sufficient to study the arrivals on time intervals starting from time $t = 0$. The equation (2.1) allows us to determine the approximate probability of a single person arriving on a short time interval $[0, \Delta]$:

$$P(N(\Delta) = 1) = \exp(-\beta\Delta)\beta\Delta = (1 + (-\beta\Delta) + \frac{(-\beta\Delta)^2}{2!} + \frac{(-\beta\Delta)^3}{3!} + \frac{(-\beta\Delta)^4}{4!} + \ldots)\beta\Delta \approx \beta\Delta \tag{2.2}$$

The approximate (2.2) follows from the shortness of the considered time interval: The higher order terms $(\beta\Delta)^2$, $(\beta\Delta)^3$, $(\beta\Delta)^4$ etc. are vanishingly small when length Δ is small, which gives us the presented approximate. We therefore note that the consumer arrivals happen at a constant rate per unit time, and that the rate parameter is β.

For estimating the value of parameter β of a Poisson process, the interarrival times of a Poisson process have an attractive charasteristic: The interarrival times have an exponential distribution with parameter β. Let T denote a random variable describing the time interval between two arrivals. Let t denote some point in time, and let us consider the probability that, starting from the last arrival, no one has arrived before t. This is equivalent to the interarrival time T being longer than t, which gives us equivalence (2.3):

$$P(T > t) = P(N(t) = 0) = \exp(-\beta t) \Leftrightarrow P(T \leq t) = 1 - \exp(-\beta t) \tag{2.3}$$

Now, the equation on the right in equivalence (2.3) states that the cumulative distribution function of random variable T is the cumulative distribution function of the exponential distribution with parameter β. The interarrival times therefore have an exponential distribution with parameter β.

The Poisson process has also an attractive characteristic called the superposition principle (see e.g. Feldman and Valdez-Flores (2010, p. 120)): Given two independent Poisson processes $\{N_1(t) : t \geq 0\}$ and $\{N_2(t) : t \geq 0\}$ with parameters β_1 and β_2, the superpositioning of these processes, i.e. $\{M(t) : t \geq 0\} = \{N_1(t) + N_2(t) : t \geq 0\}$, forms a Poisson process with parameter $\beta_1 + \beta_2$. We may then superposition any number of Poisson processes by induction and only consider the resulting "sum process." It is therefore sufficient study only a single Poisson process describing market-wide arrivals.

2.2.2 Product lifetime

I assume that the product lifetime follows an exponential distribution with the parameter λ. This assumption is equivalent to the assumption (ii) in Brauer and Castillo-Chávez (2001, p. 352) with a new interpretation. The following analysis is reinterpreted from Brauer and Castillo-Chávez (2001, p. 39-40) that deals specifically with the lifetime of an organism. Let L denote a random variable describing the product lifetime such that $L \sim Exp(\lambda)$. We may now derive an approximation for the probability that a product breaks down on a small time interval $]t, t+\Delta]$ given that it has not broken down before time t.

$$\begin{aligned} P(\text{Product breaks down in }]t, t+\Delta] \mid \text{Product working at } t) &= P(L \leq t+\Delta \mid L > t) \\ &= \frac{P(t < L \leq t+\Delta)}{P(L > t)} \\ &= \frac{P(t < L \leq t+\Delta)}{1 - P(L \leq t)} \end{aligned} \tag{2.4}$$

Using the cumulative distribution function $F(t)$, we have the approximate probability $P(t < L \leq t+\Delta) \approx F(t+\Delta) - F(t)$. Moreover, when the time interval $]t, t+\Delta]$ is small, we have a further approximate (2.5) for the probability $P(t < L \leq t+\Delta)$:

$$P(t < T \leq t+\Delta) \approx \Delta\left(\lim_{\Delta \to 0}\left(\frac{F(t+\Delta) - F(t)}{\Delta}\right)\right) = \Delta f(t) \tag{2.5}$$

In approximate (2.5), $f(t) = d/dtF(t)$ is the probability density function of random variable T.

The cumulative distribution function of the exponential distribution with parameter λ is $F(t) = 1 - \exp(-\lambda t), t \geq 0$ and the corresponding probability density function is $f(t) = \lambda \exp(-\lambda t), t \geq 0$. Using these functions and approximate (2.5), we may manipulate equation (2.4) further:

$$P(T \leq t+\Delta \mid T > t) = \frac{P(t < T \leq t+\Delta)}{1 - P(T \leq t)} \approx \frac{f(t)\Delta}{1 - F(t)} = \frac{\lambda \exp(-\lambda t)\Delta}{\exp(-\lambda t)} = \lambda\Delta \tag{2.6}$$

Based on the approximate (2.6) we then note that the probability of the product breaking down before time $t+\Delta$ is independent of time t and approximately proportional to the length Δ of the time interval. The product breakdowns therefore happen at a constant rate per unit time, and the rate parameter is λ.

2.2.3 Estimating β and λ

Since the interarrival times and product lifetimes have exponential distributions, we have an easy way to estimate both β and λ. Let $X \sim Exp(\alpha)$ denote a random variable with exponential distribution and consider the expected value of X:

$$\begin{aligned} EX &= \int_{-\infty}^{\infty} xf(x)dx \\ &= \int_{0}^{\infty} x\alpha \exp(-\alpha x)\, dx \\ &= -\left[\Big|_{0}^{\infty} x \exp(-\alpha x) - \int_{0}^{\infty} \exp(-\alpha x)dx\right] \\ &= -\left[(1/\alpha)\Big|_{0}^{\infty} \exp(-\alpha x)\right] = 1/\alpha \end{aligned} \tag{2.7}$$

The arithmetic mean $\bar{X}$ of the realizations of X is an estimator the expected value of X. We may now estimate α using equation (2.7) as $\alpha \approx 1/\bar{X}$. If we have data on the average interarrival times $\bar{t}$ and average product lifetimes $\bar{l}$, we have the estimates $\hat{\beta} = 1/\bar{t}$ and $\hat{\lambda} = 1/\bar{l}$ for parameters β and λ.

2.3 Buying decision

The buying decisions are modeled through buying probabilities. The buying probabilities are derived from a modified multinomial logit model (see e.g. Anderson et al. (1992)). Specifically, I modify the multinomial logit model in Anderson et al. (1992, p. 66) and present here the derivation of the buying probabilities from Anderson et al. (1992, p. 39-40). The multinomial logit model determines the buying probabilities through the indirect utility function of the consumer. I make two modifications to the indirect utility functions: First, I introduce a trend effect term to the function to model the effects of trends. Next, as I wish to study duopoly models, I introduce a compartment-based modification to the indirect utilities. I assume that already owning a similar product decreases the indirect utility when purchasing a new product.

2.3.1 Buying probabilities

Let us suppose that at the buying decision there are n different products of which the consumer may choose from. We model this situation by considering a decision over $n+1$ different variants. The extra $n+1$th variant is the not-buying option, where the consumer chooses not to buy anything. The buying probabilities are derived using indirect utilities. I assume that the indirect utility $\tilde{V}_i$ of variant i is given by equation (2.8). If the term $a_i g(I_i)$ is dropped, the equation (2.8) is identical to the indirect utility function defined in Anderson et al. (1992, p. 66).

$$\tilde{V}_i = y - p_i + a_i + a_i g(I_i) + \epsilon_i \tag{2.8}$$

The terms in equation (2.8), where applicable, are interpreted as in Anderson et al. (1992, p. 66). The term y is the consumer's real income at the buying decision. The term p_i is the price of variant i. The term a_i is the quality index of variant i, i.e. a_i summarizes all the observable characteristics of variant i in money terms. I refer to a_i as the intrinsic quality of variant i. The term I_i is the amount of variant i in the population at the moment of the buying decision, i.e. how many others own the product at the time of the buying decision. The term $g(I_i)$ is the effect of the trend associated with variant i to the consumers indirect utility. Finally, the term ϵ_i is a random variable and describes the consumer's idiosyncratic, fluctuating tastes at the buying decision. The realizations of random variable ϵ_i would describe why I would prefer Levi's jeans to Lee jeans on a certain day and the other way around on another day ceteris paribus. The expected value of random variable ϵ is assumed to be zero, i.e. $E\epsilon = 0$, meaning that on average the observable terms determine the consumer's behavior.

The consumer chooses variant i if the variant i's indirect utility is the largest indirect utility of all variants, i.e. when condition $\tilde{V}_i = \max_{1 \le j \le n+1} \tilde{V}_j$ holds. I assume, as in Anderson et al. (1992, p. 66), that consumers can always afford any variant i, i.e. $0 \le p_i \le y, i = 1, \ldots, n+1$. The real income y then has no bearing on consumer choice. Let $\hat{\epsilon}_i$ be the realization of the random variable ϵ_i, and $\hat{V}_i$ the associated indirect utility. Then the consumer chooses variant i when the equivalent conditions (2.9) hold:

$$\begin{aligned} \hat{V}_i &= \max_{1 \le j \le n+1} \hat{V}_j \\ \Leftrightarrow y - p_i + a_i + a_i g(I_i) + \hat{\epsilon}_i &= \max_{1 \le j \le n+1} y - p_j + a_j + a_j g(I_j) + \hat{\epsilon}_j \\ \Leftrightarrow a_i - p_i + a_i g(I_i) + \hat{\epsilon}_i &= \max_{1 \le j \le n+1} a_j - p_j + a_j g(I_j) + \hat{\epsilon}_j \end{aligned} \tag{2.9}$$

Following Anderson et al. (1992, p. 36), we may now derive the buying probability for variant i. Let $\epsilon = (\epsilon_1, \ldots, \epsilon_{n+1})$ be a random vector with the joint cumulative distribution function F. The random variables $\epsilon_i, i = 1, \ldots, n+1$ are by assumption independent. We may then write random vector ϵ's cumulative distribution function as $F(x_1, \ldots, x_{n+1}) = F_1(x_1) \cdot \ldots \cdot F_{n+1}(x_{n+1})$. The cumulative functions for each random variable ϵ_i are next assumed to be the same, meaning that we may write $F_1(x) = \ldots = F_{n+1}(x) = H(x)$. To lighten the notation, let us write $u_i = a_i - p_i + a_i g(I_i)$. Now, the probability of the consumer choosing variant i is given by equation (2.10):

$$\begin{aligned} P(\text{Consumer chooses variant } i) &= P(\tilde{V_i} = \max_{1 \leq j \leq n+1} \tilde{V_j}) \\ &= P(u_1 + \epsilon_1 \leq u_i + \epsilon_i, \ldots, u_{n+1} + \epsilon_{n+1} \leq u_i + \epsilon_i) \\ &= P(\epsilon_1 \leq u_i + \epsilon_i - u_1, \ldots, \epsilon_{n+1} \leq u_i + \epsilon_i - u_{n+1}) \end{aligned} \tag{2.10}$$

Again, let $\hat{\epsilon}_i$ be a realization of random variable ϵ_i. Then from equation (2.10) and from random variables ϵ_i being i.i.d., we may write the probability of choosing variant i as in equation (2.11):

$$\begin{aligned} P(\text{Consumer chooses variant } i \mid \epsilon_i = \hat{\epsilon}_i) &= P(\epsilon_1 \leq u_i + \hat{\epsilon}_i - u_1, \ldots, \epsilon_{n+1} \leq u_i + \hat{\epsilon}_i - u_{n+1}) \\ &= F(u_i + \hat{\epsilon}_i - u_1, \ldots, u_i + \hat{\epsilon}_i - u_{n+1}) \\ &= F_1(u_i + \hat{\epsilon}_i - u_1) \cdot \ldots \cdot F_{n+1}(u_i + \hat{\epsilon}_i - u_{n+1}) \\ &= \prod_{j \neq i} H(u_i + \hat{\epsilon}_i - u_j) \end{aligned} \tag{2.11}$$

Accounting then for all possible realizations of random variable ϵ_i with the probability density function $h(x) = d/dx H(x)$, the buying probability is determined by equation (2.12):

$$\begin{aligned} P(\text{Consumer chooses variant } i) &= \int_{-\infty}^{\infty} P(\epsilon_i = x) P(\epsilon_1 \leq u_i + x - u_1, \ldots, \epsilon_{n+1} \leq u_i + x - u_{n+1}) dx \\ &= \int_{-\infty}^{\infty} h(x) \prod_{j \neq i} H(u_i + x - u_j) dx \end{aligned} \tag{2.12}$$

To proceed further it is necessary to fix the cumulative distribution function $H(x)$. Let us assume that the random variable ϵ_i has a double exponential distribution as this assumption leads to tractable forms for the buying probabilities. The cumulative distribution function for ϵ_i then is $H(x) = \exp(-\exp(-\frac{x}{\mu} - \gamma))$, where $\mu > 0$ is a constant and γ is Euler's constant. The probability density function of ϵ_i is $h(x) = \frac{1}{\mu} \exp(-\frac{x}{\mu} - \gamma) \exp(-\exp(-\frac{x}{\mu} - \gamma))$. Then, following Anderson et al. (1992, p. 39-40), let us first write $z_j = \exp(u_j/\mu)$ and change the variable of integration of equation (2.12) to $\delta = \exp(-x/\mu - \gamma)$. For the change of variable of integration, we have $d\delta = -\frac{1}{\mu} \exp(-x/\mu - \gamma) dx$, and for the limits of the integration, we note that $\exp(-\infty/\mu - \gamma) = 0$ and $\exp(-(-\infty)/\mu - \gamma) = \infty$. We may now, using equation (2.12), determine the buying probabilities:

$$\begin{aligned}
P(\text{Consumer chooses variant } i) &= \int_{-\infty}^{\infty} h(x)\prod_{j\neq i} H(u_i + x - u_j)dx \\
&= \int_{-\infty}^{\infty} \frac{1}{\mu}\exp(-\frac{x}{\mu} - \gamma)\exp(-\exp(-\frac{x}{\mu}\exp -\gamma)) \cdot \\
&\quad \prod_{j\neq i}\exp(-\exp(-\frac{u_i + x - u_j}{\mu} - \gamma))dx \\
&= -\int_{\infty}^{0} \exp(-\delta)\prod_{j\neq i}\exp\Big(-\delta\exp(\frac{u_j}{\mu})\exp(-\frac{u_i}{\mu})\Big)d\delta \\
&= \int_{0}^{\infty} \exp(-\delta)\prod_{j\neq i}\exp\Big(-\frac{\delta z_j}{z_i}\Big)d\delta \\
&= \int_{0}^{\infty} \exp\Big(-\delta(\sum_{j=1}^{n+1}\frac{z_j}{z_i})\Big)d\delta \\
&= \frac{-z_i}{\sum_{j=1}^{n+1} z_j}\Big|_0^{\infty}\exp\Big(-\delta(\sum_{j=1}^{n+1}\frac{z_j}{z_i})\Big) \\
&= \frac{z_i}{\sum_{j=1}^{n+1} z_j} = \frac{\exp(u_i/\mu)}{\sum_{j=1}^{n+1}\exp(u_j/\mu)} \qquad (2.13)
\end{aligned}$$

The not-buying option $n+1$ has no intrinsic quality or price, i.e. we have $a_{n+1} = p_{n+1} = 0$ which means $u_{n+1} = 0$. We may therefore rewrite equation (2.13) as equation (2.14):

$$P(\text{Consumer chooses variant } i) = \frac{\exp(u_i/\mu)}{1 + \sum_{j=1}^{n}\exp(u_j/\mu)} = \frac{\exp(\frac{a_i - p_i + a_i g(I_i)}{\mu})}{1 + \sum_{j=1}^{n}\exp(\frac{a_j - p_j + a_j g(I_j)}{\mu})} \qquad (2.14)$$

Anderson et al. (1992, p. 42-5) discusses the properties of systems such as (2.14). Two notable characteristics from the discussion bear restating: Let $P(i)$ denote $P(\text{Consumer chooses variant } i)$. As $\mu \to 0$, the variance of ϵ_i tends to 0. Then $\lim_{\mu\to 0} P(i) = 1$ when $u_i > \max_{j\neq i} u_j$, and as $\mu \to \infty$, $\lim_{\mu\to\infty} P(i) = 1/n$ when there are n variants. The parameter μ therefore describes the informational value of u_i. A small μ implies that consumers are prudent in their choices, and thus the prices and the intrinsic qualities of the products play a relatively larger part in the buying decision. A large μ implies that consumers do not care as much about the prices or the intrinsic qualities, and thus firms have more leeway in their price-quality choices.

2.3.2 State-dependent indirect utility

The previously presented buying probabilities describe how a consumer who does not own any products buys a product from one of a large number of firms. However, if a consumer already owns a product from a firm, the probability of buying a similar product from a different firm is surely affected by this fact. Since the models in this thesis describe either a monopoly or a duopoly, I will describe the buying probabilities in only these types of markets.

I cannot cite previous research that supports (or weakens) the modifications I propose. Nevertheless I argue that the modifications are in line with common sense. Fortunately, the proposed models allow changing the type of buying probabilities relatively easily, should it become necessary.

Monopoly

First let us consider a monopoly. Let a denote the quality and p the price that the monopoly sets. Let $I(t)$ denote the amount of consumers who own the monopoly's product at time t. The buying probability for the monopoly's product is now given by equation (2.15):

$$P(\text{Buy from monopoly}) = \frac{\exp\left(\frac{a-p+ag(I(t))}{\mu}\right)}{1+\exp\left(\frac{a-p+ag(I(t))}{\mu}\right)} \tag{2.15}$$

As there is only a single firm of in the market, the consumer faces a buying decision between two variants. These variants are the monopoly's product and the not-buying option. We then note that equation (2.15) is a special case of equation (2.14) with two variants.

Duopoly

Next let us consider a duopoly. Let us name the duopoly firms firm 1 and firm 2. Let a_i denote the quality and p_i the price firm $i = 1, 2$ sets. Let $I_i(t)$ denote the amount of consumers who own the product of firm i at time t. Additionally, let $I_{12}(t)$ denote the amount of consumers who own products from both firms at time t.

The buying probability for a consumer who does not own any products is now given by equation (2.16). The buying decision is now between three variants: These variants are firm 1's product, firm 2's product and the not-buying option. As with the monopoly buying probability, equation (2.16) is then a special case of equation (2.14).

$$P(\text{Buy from firm } i \mid \text{No products}) = \frac{\exp\left(\frac{a_i-p_i+a_ig(I_i(t)+I_{12}(t))}{\mu}\right)}{1+\exp\left(\frac{a_1-p_1+a_1g(I_1(t)+I_{12}(t))}{\mu}\right)+\exp\left(\frac{a_2-p_2+a_2g(I_2(t)+I_{12}(t))}{\mu}\right)} \tag{2.16}$$

As we are considering a duopoly, we may ask what effect does owning product j have on the buying probability for product $i \neq j$. I assume that the products firms produce are "substitutes-in-intrinsic-quality." When a consumer already owns a product j, the intrinsic quality of product i is the maximum of $a_i - a_j$ and 0. The consumers do not therefore experience indirect utility when the product they already own is of higher intrinsic quality than the new product, i.e. when $a_j > a_i$ we have $\max\{a_j - a_i, 0\} = 0$.

The buying probabilities are now given in equation (2.17): When the consumer already owns product j and considers buying the other product i, the buying decision is between two variants, product i and the not-buying option. Equation (2.17) now follows the same logic as equation (2.14) but with the alternation that the intrinsic qualities have been changed from a_i to $\max\{a_i - a_j, 0\}$.

$$P(\text{Buy from firm } i \mid \text{Own product from firm } j) = \frac{\exp\left(\frac{\max\{a_i-a_j,0\}-p_i+a_ig(I_i(t)+I_{12}(t))}{\mu}\right)}{1+\exp\left(\frac{\max\{a_i-a_j,0\}-p_i+a_ig(I_i(t)+I_{12}(t))}{\mu}\right)} \tag{2.17}$$

From equations (2.16) and (2.17) we may note two implicit assumptions about the trend effect. First, I assume that the trend effect is independent of the types of consumers owning the product. Consumers in compartments I_i and I_{12} contribute to the trend effect in the same exact way. This can be seen in term $I_i(t) + I_{12}(t)$ as the sum of the consumers enters into the function $g(I_i(t) + I_{12}(t))$. Next, I assume that the trend effect and the effect of the intrinsic quality are separate in the buying probability. The trend effect is seen as an effect of the consumer's environment (for example as peer pressure to conform) and is not therefore affected by the ownership of a product. This assumption can be seen in equation (2.17) as intrinsic quality is adjusted in term $\max\{a_i - a_j, 0\}$ but not in term $a_i g(I_i(t) + I_{12}(t))$.

3

Methods and assumptions for the numerical solution

The market evolution, and through it the firms' profits, are determined by initial value problems. I am unfortunately unable to provide analytical solutions for these initial value problems and therefore have to settle for numerical solutions. The numerical solutions require us to make a number of assumptions regarding explicit functional forms and the values of exogenous variables. This chapter describes the chosen assumptions and the methods used for the numerical solutions.

The use of numerical methods naturally leads to questioning what part of firm behavior is due to the chosen values for the exogenous variables and what part to model formulation. To shed light on this issue, I divide the exogenous variables to constant exogenous variables and experimental exogenous variables. Constant exogenous variables remain, as their name suggests, constant while the experimental exogenous variables are varied to provide information on the behavior of the models with respect to these variables. The aim of this exercise is to do sensitivity analysis or "comparative-statics-by-other-means," as the traditional way of doing comparative statics through linearization is ruled out by the lack of analytical solutions to the initial value problems.

Solving the firm behavior in the models requires us to define the firms' choice sets beforehand. This requirement is due to the way optimization is implemented given the game theoretical structures of the models. The choice sets describe how a firm may choose its decision variables, and as such they define the range of possible observable firm behavior. The assumptions regarding these sets are then significant when considering the research questions.

3.1 Numerical methods

In order to be able to maximize profits, we must first be able to calculate profits. In order to calculate profits, we must be able to solve an initial value problem and to integrate a function numerically. I solve the initial value problems by using the (fourth-order) Runge-Kutta method and integrate numerically by using the Trapezoidal rule. Both of these methods were chosen because they are extremely well-known. Moreover, MATLAB provides suitable ready-made functions for both tasks.

3.1.1 Runge-Kutta Method

As mentioned before, the market behavior in each of the following models is described by an initial value problem. An initial value problem, in turn, consists of one or more differential equations and initial values. I give here a brief presentation of the classical fourth-order Runge-Kutta method for an initial value problem with a single (first-order) differential equation and a single initial value from Nagle et al. (2011, p. 134). For the same method applied to initial value problems with more than one differential equation and initial value, see for example Iyengar and Jain (2009, p. 208) or Chapra (2005, p. 500).

Let us consider the initial value problem (3.1):

$$y'(x) = f(x, y(x)), y(x_0) = y_0 \tag{3.1}$$

We wish to find the function $y(x)$ that satisfies the conditions of the initial value problem (3.1) on a given interval. Since finding the function analytically might be exceedingly difficult, we might choose to approximate the solution $y(x)$ instead. The classical fourth-order Runge-Kutta method approximates the solution in the following manner: Suppose that we are interested in how the solution $y(x)$ behaves on some interval $[x_0, c]$. We first choose the number of steps N that determine the points at which the solution $y(x)$ is approximated. Naturally, in order to stay in the interval in our steps, our stepsize h satisfies $Nh = c - x_0$ or $h = (c - x_0)/N$. We then start to move from point x_0 with step size h and approximate the function $y(x)$ using the following equations (3.2) and (3.3):

$$\begin{aligned} x_{n+1} &= x_n + h \\ y_{n+1} &= y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned} \tag{3.2}$$

where

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \\ k_3 &= hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}) \\ k_4 &= hf(x_n + h, y_n + k_3) \end{aligned} \tag{3.3}$$

Given the initial value $y(x_0) = y_0$ and the step size h, equations (3.2) and (3.3) can then be used to approximate the solution $y(x)$ with each y_n-value. We note that the presented method is recursive as we use the value y_n for calculating the value y_{n+1}. There are also more sophisticated versions of the Runge-Kutta method. Nagle et al. (2011, p. 136) describes an algorithm that replaces the need of choosing the number of steps with a tolerance.

The ready-made function for the fourth-order Runge-Kutta method in MATLAB is *ode45*. This function was used for solving the initial value problems of all three proposed models.

3.1.2 Trapezoidal rule

In order to be able to calculate profits, it is also necessary to be able to integrate numerically. While there exists more sophisticated numerical integration methods (for example Simpson's rule or Gaussian quadrature), I chose to use the Trapezoidal rule for its ease of implementation. I give here a brief presentation of it adopted from Chapra (2005, p. 399-402). Strictly speaking, the method presented here is the composite trapezoidal rule (Chapra (2005, p. 401)). Suppose that we have a function $R(t)$ that describes the revenue of a firm at time t. Moreover, suppose that we are interested in the total revenue R_{total} of the firm over some time interval $[a, b]$. To find the total revenue, we need to integrate $R(t)$ over the interval $[a, b]$:

$$R_{\text{total}} = \int_a^b R(t)dt \tag{3.4}$$

There is no guarantee that the integral on the right-hand side in equation (3.4) can be calculated analytically. We therefore wish to find a way to approximate the integral. First, we note that the interval $[a, b]$ may be divided into n smaller intervals $[x_0, x_1], \ldots, [x_{n-1}, x_n]$ such that $x_0 = a, x_n = b$ and $x_i - x_{i-1} = x_j - x_{j-1}$ for all $i, j = 1, \ldots, n$. We may then write the integral on the right-hand side of equation (3.4) as follows:

$$\int_a^b R(t)dt = \int_{x_0}^{x_1} R(t)dt + \ldots + \int_{x_{n-1}}^{x_n} R(t)dt \tag{3.5}$$

Calculating the integrals over the smaller intervals in equation (3.5) is usually as difficult as calculating the integral on the right-hand side of equation (3.4). However, if we can evaluate the function values $R(t_0)$ at an arbitrary time t_0, we may approximate the integrals using the function values and reduce the approximation error by considering smaller intervals. Let us consider segment $[x_{i-1}, x_i]$. We approximate the integral with the area of a trapezoid that is under the straight line connecting points $(x_{i-1}, R(x_{i-1}))$ and $(x_i, R(x_i))$. The area of the trapezoid is given by the product of the trapezoid's width and its average height.

$$\int_{x_{i-1}}^{x_i} R(t)dt \approx (x_i - x_{i-1})\frac{R(x_{i-1}) + R(x_i)}{2} = \text{width} \cdot \text{average height} \tag{3.6}$$

As the width of every segment $[x_{i-1}, x_i]$ is the same by assumption, let us denote this width by the term $h = x_i - x_{i-1} = (b - a)/n$. We may then use the approximate (3.6) for approximating the integral over the whole interval $[a, b]$:

$$\begin{aligned}
\int_a^b R(t)dt &= \int_{x_0}^{x_1} R(t)dt + \ldots + \int_{x_{n-1}}^{x_n} R(t)dt \\
&\approx (x_1 - x_0)\frac{R(x_0) + R(x_1)}{2} + \ldots + (x_n - x_{n-1})\frac{R(x_{n-1}) + R(x_n)}{2} \\
&= h\frac{R(x_0) + R(x_1)}{2} + \ldots + h\frac{R(x_{n-1}) + R(x_n)}{2} \\
&= \frac{h}{2}[R(x_0) + 2\sum_{i=1}^{n-1} R(x_i) + R(x_n)] \qquad (3.7)
\end{aligned}$$

We have now constructed a way to approximate the total revenues over the time interval $[a, b]$ with the function values $R(x_i)$. We may note that trapezoids over smaller intervals provide a better approximates for the area between the t-axis and $R(t)$. Then as the number of segments n increases, the width of the smaller intervals decreases and the approximate (3.7) approaches the correct value of the integral in equation (3.4).

The ready-made function for the Trapezoidal rule in MATLAB is *trapz*. This function was used for calculating the profits in all three proposed models.

3.2 Parameter choices

3.2.1 Firm's choice set

The models allow a firm to control three aspects of its product: the quality of the product, the price of the product and the amount of free samples the firm gives. The natural approach in modeling these aspects would be to allow them to be chosen from positive real numbers. However, this natural approach poses problems for the numerical solution of the duopoly models. The way the firms interact in the duopoly models is dictated by the game theoretical structures of the models. This interaction can be seen as best-responding where each firm seeks to maximize its profits by optimizing its choices with respect to the choices of the other firm within the structures of the duopoly models. I was regrettably unable to find suitable tools to handle best-responding with choices from positive real numbers, and so I limit the firms' choices to the following sets (3.8):

$$\begin{aligned} \text{The set of qualities } A &= \{0, 1, 2, 3\} \\ \text{The set of free samples } Q &= \{0, 200\,000, 400\,000, 600\,000\} \\ \text{The set of prices } P &= \{0, 3, 6, 9\} \end{aligned} \tag{3.8}$$

With the values in sets (3.8) I hope to allow a multitude of choices to the firm while keeping the models computable. I offer the following interpretations for the values: The choice 0 in quality or the free samples represents the case where the firm does not invest anything to the particular aspect of the product. The choice 1 in quality and the choice $200\ 000$ in free samples represents a small investment in that particular aspect. The choice 2 in quality and the choice $400\ 000$ in free samples represents an average or mid-size investment in that particular aspect. The choice 3 in quality and the choice $600\ 000$ in free samples represents a large investment in that particular aspect. Likewise, the price choice 0 represents a free product, the price choice 3 represents a low price, the price choice 6 represents an average price and the price choice 9 represents a high price.

3.2.2 Constant exogenous variables

Table 3.1 gives the constant exogenous variable values. These values were chosen arbitrarily. Following the analysis in section 2.7, we note that with these value choices the average interval between arriving consumers is 1 and the average product lifetime is 2. Products should not therefore disappear from the population purely due to their short lifetime, since on average two buying decisions will be made per one product breakdown.

Parameter name	Parameter interpretation	Parameter value
N	The amount of consumers in market	1 000 000
μ	The impact of indirect utility	0.5
β	Consumer arrival rate	1
λ	Product breakdown rate	0.5
c_1	Cost scaling coefficient	0.5
c_2	Minimum cost constant	1
m	Trend saturation point	333 333

Table 3.1: Constant exogenous variables

Why choose to keep these exogenous variables as constant? The reason for this is two-fold: In table 3.1, the first set of variables $N, \mu, \beta, \lambda, c_1, c_2$ are perhaps the easiest parameters to measure. If we were interested in applying the models in serious research, we could perhaps be able find appropriate values for these variables. Next, the variable m is kept constant as it is required by a one of the proposed specifications of the trend effect $g(I)$. Since the trend effects in this thesis only serve as examples of possible trend types, I choose to leave m as a constant.

3.2.3 Experimental exogenous variables

After describing the constant exogenous variables, I present the experimental variables and their prospective values in table 3.2.

Parameter name	Parameter interpretation	Parameter value	Value interpretation
T	Firm's planning horizon	$T = 10$	Short planning horizon
		$T = 100$	Long planning horizon
r	Discount rate	$r = 0$	Patient firms
		$r = 0.05$	Normal Firms
		$r = 10$	Impatient Firms
d	Strength of the trend	$d = 2$	Weak trend
		$d = 8$	Strong trend

Table 3.2: Experimental exogenous variables

Why choose to vary these exogenous variables? None of the variables are easily observable or measurable. We might reasonably expect that firms would keep their planning horizons T and discount rates r as secrets. The planning horizon T and the discount rate r, along with the strength of the trend d, should additionally have large effects on the profit calculations of a firm and thus on the behavior of the firm. These variables are then the most interesting exogenous variables with regard to firm behavior which is why they are chosen for the "comparative-statics-by-other-means."

3.3 Functional form choices

3.3.1 Trend term functions $g(I)$

As there seems to be no clear, exact and universal definition of what is meant by the word "trend," I propose to consider a product trendy if the probability of buying it is affected by the number of other consumers already owning the product. This broad definition allows for an infinite number of ways to define the relationship between the buying probability and the number of other consumers owning the product. In this thesis I study perhaps the three simplest ways: the nonexistent relationship, a linear relationship in indirect utility and a parabel-shaped relationship in indirect utility. I refer to those products where there is no relationship between the buying probability and the number of other consumers owning the product as products with no trend, to those products where there is linear relationship as products with a linear trend and to those products with a parabel-shaped relationship as products with a parabel trend. Let us next give explicit forms for these relationships by defining the functional forms of the trend term functions $g(I)$.

The functional forms of the trend types are constructed using the previously mentioned exogenous variables and the compartment-specific functions discussed in section 2.1. The parameter N denotes the number of consumers in the market. The term I denotes the number of other consumers who already own the product. For the models in this thesis, the term I takes the form $I(t)$ or $I_i(t) + I_{ij}(t)$ as we may express the number of other consumers owning the product at time t with the compartment-specific functions. The functional forms for the trend term functions $g(I)$ are as follows:

1. Products with no trend have the form $g(I) = 0$

2. Products with a linear trend have the form $g(I) = d\frac{I}{N}$, $d > 0$
3. Products with a parabel trend have the form $g(I) = -\frac{d}{m^2}I^2 + 2\frac{d}{m}I$, $d > 0$, $0 \leq m \leq N$

The parameter d is interpreted as the strength of the trend effect. When the trend is at its peak, the trend increases the perceived intrinsic quality of product i by da_i. The parameter m is interpreted as the saturation point for the parabel-shaped relationship. The trend effect will start to diminish after I becomes larger than m. The differences between the functional forms may best be seen in figures 3.1, 3.2 and 3.3.

These functional forms are only a brief glance into all possible trend. For example, we might reasonably assume that the effects of trends are time-dependent. Instead of the forms $g(I)$ for the trend terms, we might argue that an appropriate functional form could be $g^*(I) = g^*(I(t), t) = g(I(t))/(1 + t)$. The trend effect would now diminish as time increases as we might reasonably expect in many cases. Arguing further, we might even require that the trend term has, for example, the form $g^{**}(I) = g^{**}(I(t), t) = g(I(t)) - \alpha t^2$ where $\alpha > 0$ is some small constant. As time increases, the second term αt^2 will start to dominate which will mean that the buying probability tends to 0. This would mean that the product would eventually disappear from the market. However, this type of analysis is out of the scope of this thesis.

We may also question how the trend effect is tied to the quality of the product through the term $ag(I)$ (see e.g. equation (2.14)). This implies that there is no trend effect when the intrinsic quality is nonexistent, i.e. $a = 0 \Rightarrow ag(I) = 0$ for all I. There is no fundamental reason to assume this. However, I would argue that this is a reasonable assumption for the purposes of this thesis. After all, common sense seems to suggest that higher quality products are associated with stronger trends.

3.3.2 Cost function

I assume that the firm has a constant, quality-dependent per-unit cost given by the function $C(a)$. The cost is determined purely by the intrinsic quality a of the product. I assume that cost function $C(a)$ is increasing and convex in intrinsic quality a. A similar formulation may be found in Anderson et al. (1992, p. 239).

Why not to normalize per-unit costs to 0? Let us consider firm i's intrinsic quality choice in a monopoly or a duopoly. A positive quality a_i can only increase the buying probabilities (2.15), (2.16) and (2.17) through terms a_i, $\max\{a_i - a_j, 0\}$ and $a_i g(I_i(t))$. Normalizing the per-unit costs to 0 would then lead firm i to always choose the highest possible quality.

I arbitrarily chose the functional form of the per-unit cost function as $C(a) = c_1(a^2 + c_2)$, $c_1, c_2 > 0$. This cost function $C(a)$ is increasing and convex in a as $d/daC(a) > 0$ and $d^2/da^2C(a) > 0$. The parameter c_1 may be interpreted as the cost scaling term of the firm, i.e. a low c_1 implies that the firm is efficient. The parameter c_2 may be interpreted as the minimum cost constant of the firm, i.e. the parameter value c_2 describes the minimum cost of making a product with 0 quality given some cost scaling term c_1.

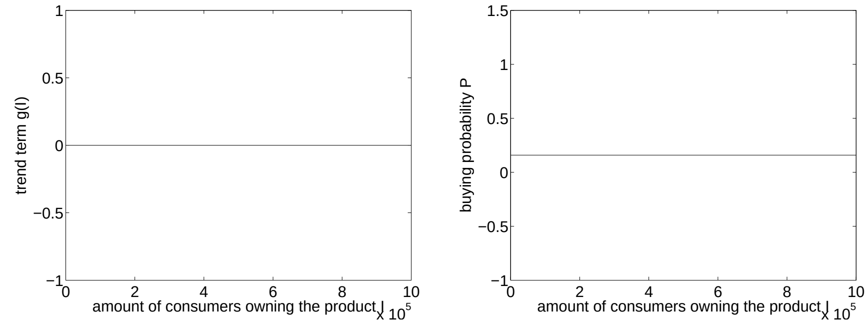


Figure 3.1: Monopoly buying probability for product with no trend, $g(I) = 0, a = 1, p = 2, \mu = 0.6$

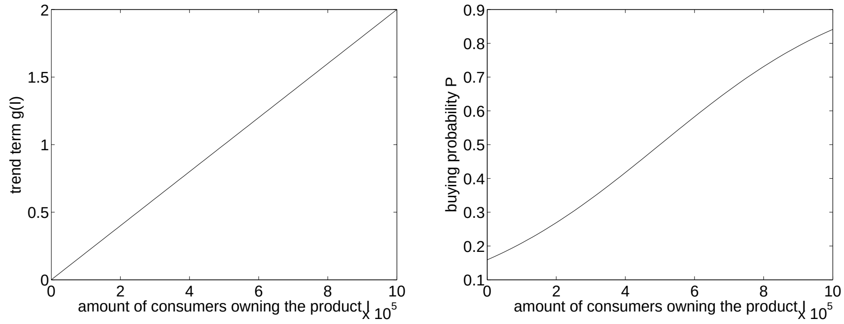


Figure 3.2: Monopoly buying probability for product with linear trend, $g(I) = d\frac{I}{N}, a = 1, p = 2, \mu = 0.6, N = 1000000, d = 2$

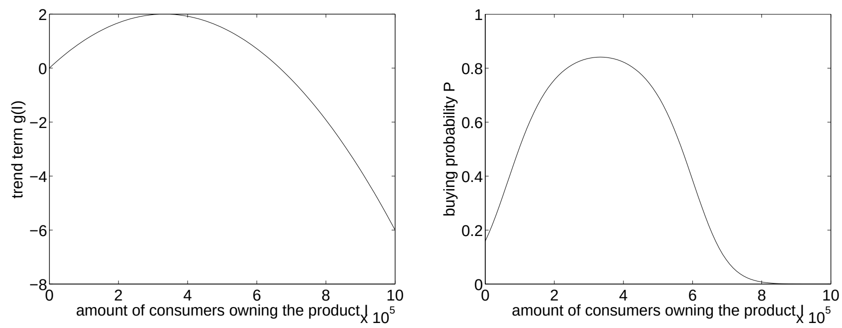


Figure 3.3: Monopoly buying probability for product with parabel trend, $g(I) = -\frac{d}{m^2}I^2 + 2\frac{d}{m}I, a = 1, p = 2, \mu = 0.6, N = 1\,000\,000, d = 2, m = 333\,333$

4

Monopoly

This chapter presents the monopoly model. The key characteristics of the model can be summarized as follows: The monopoly sells a durable good to large population. There is no secondary market for the monopoly's product. Consumers in the market will buy only a single product and will hold the product until it breaks down. The monopoly may choose the product's quality and price. The monopoly may also give out free products to create demand through a possible trend. The monopoly sets its quality and price once and for all, and gives out free products only at the time it enters the market. The monopoly's goal is to maximize its profits, and the monopoly makes its choices accordingly.

The demand for the monopoly's product is determined through an initial value problem inspired by the SIS model. The initial value problem is determined by the monopoly's choices which allows the monopoly to influence the demand of its product. The initial value problem is therefore incorporated into the monopoly's (profit maximization) problem.

The monopoly's behavior is determined by the monopoly's problem which is solved in the following manner: We first calculate the monopoly's profits for every possible choice combination from the predefined choice sets. The monopoly then naturally chooses the combination with the highest profits. After we have found the monopoly's optimal choices, we may study the effects of different trends on them.

4.1 Market Structure

The market structure of the model is fairly simple. The population is divided into two compartments, S and I, representing whether a consumer has or has not the monopoly's product. Being in compartment S means that the consumer does not own the monopoly's product, and correspondingly being in compartment I means that the consumer owns the monopoly's product. Next, we define compartment-specific functions $S(t)$ and $I(t)$ to describe how many consumers are at which compartment. The function $S(t)$ then describes the number of consumers who do not own the product at time t, and similarly the function $I(t)$ the number of consumers who own the product at time t. The functions $S(t)$ and $I(t)$ are found by solving the initial value problem adopted from the SIS model. These functions may be used to describe how people move between the compartments, which in turn is partly the result of consumer purchasing the monopoly's product, i.e. the demand of the product.

Figure 4.1 gives the compartmental structure and describes the movement between compartments. Movement between compartments is due to either the breakdown of the product (leading the consumer to move from compartment I to compartment S) or buying the product (leading to the consumer to move from compartment S to compartment I). In chapter 2 we have established the arrival and breakdown intensities β and λ and the buying probability $P_{S\to I}$ for the monopoly's product. These now form the basis on which the consumers move between compartments in this model.

4.2 Initial value problem

The initial value problem for the monopoly model naturally consists of the differential equations that describe how the functions $S(t)$ and $I(t)$ evolve over time and the initial values $S(0) = A$ and $I(0) = B$ at the starting time $t = 0$. The initial value problem takes the form (4.1):

$$\begin{aligned} \dot{I}(t) &= \beta P_{S\to I}(a,p,I(t))S(t) - \lambda I(t) \\ \dot{S}(t) &= -\beta P_{S\to I}(a,p,I(t))S(t) + \lambda I(t) \\ \text{with} \quad & I(0) = A, S(0) = B \end{aligned} \tag{4.1}$$

The buying probability $P_{S\to I}(a,p,I(t))$ depends on the price p and the quality a of the monopoly's product and the amount of the product already in the market given by the function $I(t)$. The form of the buying probability $P_{S\to I}(a,p,I(t))$ is given by equation (4.2) which is naturally adopted from equation (2.15).

$$P_{S\to I}(a,p,I(t)) = \frac{\exp(\frac{a-p+ag(I(t))}{\mu})}{1+\exp(\frac{a-p+ag(I(t))}{\mu})} \tag{4.2}$$

The monopoly determines the initial conditions of initial value problem (4.1) by choosing what amount of free samples it gives out. Let q denote the amount of free samples that the monopoly chooses to give out. Then naturally the initial conditions are given as $I(0) = A = q$ and $S(0) = B = N - q$. We may, for example, interpret giving out free samples as the monopoly giving steep discounts to the first q consumers as an opening offer.

The terms in the differential equations in initial value problem (4.1) have intuitive interpretations: At time t, $\beta S(t)$ consumers who do not own the product arrive at buying decisions and of these consumers a total of $\beta P_{S\to I}(a,p,I(t))S(t)$ consumers buy the product. Simultaneously, a total of $\lambda I(t)$ products break down decreasing the number of consumers who own the product. Thus at time t $I(t)$ increases by $\beta P_{S\to I}(a,p,I(t))$ and decreases by $\lambda I(t)$. The effects on $S(t)$ are naturally opposite.

The population size stays constant over time by assumption, as $d/dt(S(t)+I(t)) = \dot{S}(t)+\dot{I}(t) = 0$. This leads to identity (4.3):

$$I(t) + S(t) = N \tag{4.3}$$

The differential equation system of initial value problem (4.1) may be reduced to a single differential equation by using the identity (4.3), as we may denote $S(t) = N - I(t)$. The function $S(t)$ is therefore completely driven by $I(t)$, i.e. if we know how the function $I(t)$ behaves over time, we, by using the identity (4.3), also know how the function $S(t)$ behaves over time. This observation is used in the numerical solution of initial value problem (4.1).

4.3 Firm behavior

The behavior of the firm is determined by its objective of profit maximization. As the firm in this model is a monopoly, the process of maximizing its profits is a straight-forward optimization problem.

4.3.1 Monopoly's problem

The monopoly's problem consists of three components: the objective function, the decision variables and the constraints. The objective function of the monopoly is naturally the monopoly's profits. The decision variables of the monopoly are in this case the quality, the price and the amount of free samples of its product. The constraints are formed by the market structure and the natural constraints on the decision variables. This section describes the construction of the monopoly's problem.

Let us determine how the monopoly's profits are calculated. The demand for the monopoly's product at time t is $\beta P_{S\to I}(a,p,I(t))S(t)$. The monopoly's per-unit discounted net profit is $\exp(-rt)(p-C(a))$ where r is the discount rate and $C(a)$ the per-unit cost at quality a. The monopoly's discounted profit at time t is given by the term $\exp(-rt)(p-C(a))\beta P_{S\to I}(a,p,I(t))S(t)$. The total net profit with a planning horizon T may then be calculated by integrating the monopoly's discounted profit over the planning horizon and accounting for the free sample costs $C(a)q$. The total net profit $\pi(a,q,p)$ as function of quality a, free samples q and price p is given by equation (4.4):

$$\pi(a,q,p)=\int_0^T \exp(-rt)(p-C(a))\beta P_{S\to I}(a,p,I(t))S(t)dt - C(a)q \tag{4.4}$$

Let us next consider the decision variables. The monopoly can change the price p and the quality a which enter into the buying probability (4.2). The monopoly may also determine the initial condition A of initial value problem (4.1) by giving out q free samples. These are then the decision variables of the monopoly.

We may now write the monopoly's problem. The monopoly wishes to maximize its profits (4.4) giving us the objective function. The market structure imposes restrictions on the monopoly's behavior: The functions $I(t)$ and $S(t)$ are determined by (or, by assumption, are solutions of) the initial value problem (4.1). Likewise, the monopoly cannot set a negative quality or price nor give out a negative amount of free samples. The monopoly's problem then is:

$$\begin{aligned}\max_{a,p,q} \quad & \pi(a,q,p) \quad \text{s.t.}\\ & S(t), I(t) \text{ solutions of initial value problem (4.1) on } [0,T],\\ & a,q,p \geq 0\end{aligned} \tag{4.5}$$

The structure of the monopoly's problem (4.5) hints at a way to solve problem in general terms. The constraints of the problem do not specifically limit the monopoly's choices to the sets (3.8). If analytical solutions could be found for the initial value problem (4.1) and the integral in equation (4.4), the problem would then become a matter of constrained optimization. This would allow us to use the standard tools such as the KKT conditions for solving the monopoly's optimal quality, price and the amount of free samples.

4.3.2 Solving the monopoly's problem

The monopoly's choices are, as mentioned before, next assumed to be limited to the sets (3.8). Using limited choice sets allows us to implement a simple algorithm for finding the solution to the monopoly's problem within those sets. For the sets (3.8), there is finite number of combinations of qualities, prices and free samples. The finite number of choices in each of these aspects means that the total number of combinations for all three aspects is finite as well. This makes it possible to calculate the profits for all combinations. After calculating the profits, we assume naturally that the monopoly chooses the combination that yields the largest profit. The implemented script mimics this by using simple brute force to calculate the profits for all combinations and choosing the combination with the highest profits.

Figure 4.2 describes the structure of the implemented script. The subroutine `MonopolyOptimizes` calculates the profits for all possible combinations of quality, free samples and price, and chooses the combination with the largest profits. The subroutine `MonopolyProfits` calculates the profits for a given quality, free samples and price. It solves the initial value problem (4.1) using *ode45*, uses this solution to generate a vector describing the discounted revenues over time, and finally generates the profits by using *trapz* to calculate the integral in equation (4.4) and accounting for the free sample costs. The subroutine `monopolyDE` is the (reduced) differential equation system of the initial value problem (4.1) required by *ode45*. The subroutine `MonopolyRevenue` is used to generate a revenue vector required for numerical integration. See appendix C for the implemented script.

4.4 Outcomes

Tables 4.1-4.3 give the monopoly behavior.

Planning horizon	Trend type	(a, q, p)	Profits
$T = 10$	no trend	$(2, 0, 3)$	499768
	linear, $d = 2$	$(2, 0, 3)$	$1.67055 \cdot 10^6$
	parabel, $d = 2$	$(2, 200\,000, 6)$	$7.319 \cdot 10^6$
	linear, $d = 8$	$(2, 600\,000, 9)$	$2.04485 \cdot 10^7$
	parabel, $d = 8$	$(2, 200\,000, 9)$	$2.02274 \cdot 10^7$
$T = 100$	no trend	$(2, 0, 3)$	$4.83128 \cdot 10^6$
	linear, $d = 2$	$(3, 600\,000, 6)$	$2.77606 \cdot 10^7$
	parabel, $d = 2$	$(3, 200\,000, 9)$	$8.08869 \cdot 10^7$
	linear, $d = 8$	$(2, 600\,000, 9)$	$2.15405 \cdot 10^8$
	parabel, $d = 8$	$(2, 200\,000, 9)$	$1.89226 \cdot 10^8$

Table 4.1: Patient monopoly with $r = 0$, Choices and profits

Based on the results there is little need to doubt the internal validity of the model. The monopoly will always set a positive price and that the monopoly gets a positive profit. The planning horizon T and discount rate r seem to have the expected effects: For cases with the same monopoly behavior (for example cases with no trend term with firm behavior $(a, q, p) = (2, 0, 3)$), we note that the profits are the highest when the discount rate r is the lowest and that the profits are higher as the monopoly's planning horizon T is longer. Moreover, the strength of the trend d seems to also have the expected result: We note that the monopoly's profits are higher with a strong trend with strength $d = 8$ than a weak trend with strength $d = 2$, although the strength of the trend may also affect the firm behavior.

Let us now consider the research questions. Based on the results in tables 4.1-4.3 we note that different trend types lead to different monopoly behavior across cases. There are indeed cases where the optimal behavior of the monopoly involves giving out free samples. However, we also note that there are many cases where different trend types lead to the same behavior, as for example the cases with no trend and a linear trend in table 4.3. We cannot therefore determine the type or the strength of a trend solely by observing firm behavior.

Based on the results I nevertheless offer the following, admittedly weak result regarding the existence of a trend: If

Planning horizon	Trend type	(a, q, p)	Profits
$T = 10$	no trend	$(2, 0, 3)$	395859
	linear, $d = 2$	$(2, 0, 3)$	$1.309 \cdot 10^6$
	parabel, $d = 2$	$(2, 200\,000, 6)$	$5.72545 \cdot 10^6$
	linear, $d = 8$	$(2, 600\,000, 9)$	$1.58237 \cdot 10^7$
	parabel, $d = 8$	$(2, 200\,000, 9)$	$1.61853 \cdot 10^7$
$T = 100$	no trend	$(2, 0, 3)$	973207
	linear, $d = 2$	$(2, 0, 3)$	$3.28312 \cdot 10^6$
	parabel, $d = 2$	$(3, 200\,000, 9)$	$1.51731 \cdot 10^7$
	linear, $d = 8$	$(2, 600\,000, 9)$	$4.18082 \cdot 10^7$
	parabel, $d = 8$	$(2, 200\,000, 9)$	$3.87099 \cdot 10^7$

Table 4.2: Normal monopoly with $r = 0.05$, Monopoly choices and profits

Planning horizon	Trend type	(a, q, p)	Profits
$T = 10$	no trend	$(2, 0, 3)$	5898.28
	linear, $d = 2$	$(2, 0, 3)$	6432.81
	parabel, $d = 2$	$(2, 0, 3)$	13306
	linear, $d = 8$	$(2, 0, 3)$	10162.9
	parabel, $d = 8$	$(1, 200\,000, 6)$	166090
$T = 100$	no trend	$(2, 0, 3)$	5898.27
	linear, $d = 2$	$(2, 0, 3)$	6432.81
	parabel, $d = 2$	$(2, 0, 3)$	13306
	linear, $d = 8$	$(2, 0, 3)$	10162.8
	parabel, $d = 8$	$(1, 200\,000, 6)$	166089

Table 4.3: Impatient monopoly with $r = 10$, Monopoly choices and profits

we observe the monopoly giving out free samples, we may infer the existence of a trend. However, if the monopoly chooses not to give out free samples, we may not infer the non-existence of a trend.

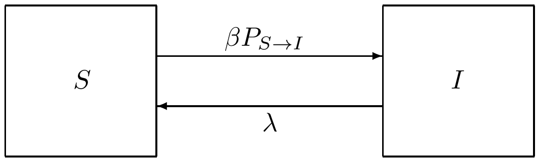


Figure 4.1: Compartmental Graph, Monopoly

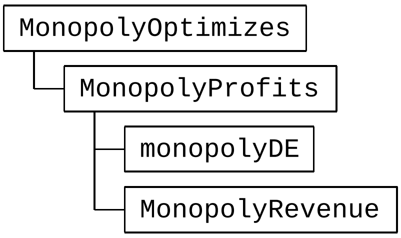


Figure 4.2: Script structure, Monopoly

5

Stackelberg duopoly

This chapter presents the first of the duopoly models of this thesis. The model is akin to the classical Stackelberg duopoly model where two firms compete by choosing quantities sequentially one after another. The key characteristics of the model are summarized as follows: There are two firms, Leader and Follower, who sell durable goods to a large population of consumers. There are no secondary markets for the products. A consumer will buy only a single product from a particular firm and hold it until the product breaks down. The consumer may buy the products from both firms. Both firms choose the quality and price of their products, and whether to give out products as free samples. Prices and qualities are set once and for all, and free samples are given only at the time of entry into the market. Both firms seek to maximize their profits and will make their choices accordingly. The firms enter the markets sequentially: First, Leader enters the market and remains the only firm for a (small) period of time. Then Follower enters the market, after which the firms compete with each other. Leader knows Follower's entry time and will use this knowledge in making its choices.

The demand for the both firms' products are determined through initial value problems inspired by the SIS model. As there are two distinct time periods for the market, there are two initial value problems describing these periods. The firms determine the initial value problems through their choices. The initial value problems are therefore incorporated into the firms' optimization problems as the firms' profits depend on the demand determined by the solutions of the initial value problems.

The behavior of both firms is determined with the help of the sequential entry assumption. We assume further that Leader knows the form of the Follower's problem. Leader then anticipates how Follower will behave given its own choices and incorporates this knowledge into its own problem. The Follower's problem is therefore nested into the Leader's problem as Leader has to determine Follower's behavior in order to calculate the profits from its own choices. We may therefore limit our attention to solving the Leader's problem as its solution will also describe how Follower behaves.

The Leader's problem is solved similarly to the way the monopoly's problem was solved in the preceeding chapter: We calculate the Leader's profits for every possible choice combination and choose the combination with the highest profits. However, as calculating the Leader's profits for a given choice combination requires determining the Follower's behavior, we solve the Follower's problem with the given choice combination every time we calculate the Leader's profits. The Follower's problem is structurally equivalent to the monopoly's problem and is therefore solved with the same method: Given the Leader's choices, we calculate the Follower's profits for every possible choice combination and choose the combination with the highest profits.

5.1 Market structure

The market's two time periods naturally have different market structures. The market structures are determined by the presence of one or both firms in the market and lead to two distinct initial value problems.

The market structure before the Follower's entry is similar to the monopoly case. The superscript $*$ denotes that we are in the pre-entry market. As the sole supplier of products is Leader, there are two compartments S^* and I_L^* with their respective functions $S^*(t)$ and $I_L^*(t)$. The function $S^*(t)$ describes the number of consumers who do not own the Leader's product at time t, and similarly $I_L^*(t)$ the number of consumers who own the Leader's product at time t. The compartmental structure of the pre-entry market is described in figure 5.1. The movement between compartments is again determined by the breakdown intensity λ and the arrival intensity β and the buying probability $P_{S^*\to I_L^*}$.

After the Follower's entry the market structure naturally changes. The population is now divided into four different compartments: S, I_L, I_F and I_{LF} with the respective functions $S(t)$, $I_L(t)$, $I_F(t)$ and $I_{LF}(t)$. The function $S(t)$ describes the amount of people who have neither product, the function $I_L(t)$ the amount of people who own the Leader's product, the function $I_F(t)$ the amount of people who own the Follower's product and the function $I_{LF}(t)$ the amount of people who own a product from both firms at time t. Figure 5.2 describes the compartmental structure of the post-entry market. The movement is again determined by the breakdown and arrival intensities λ and β, which remain unchanged after the market structure changes, and the buying probabilities $P_{i\to j}, i \in \{S, I_L, I_F\}, j \in \{I_L, I_F, I_{LF}\}$.

5.2 Initial value problems

To calculate the profits for either firm we first need to determine the demand for the firm's product. The demand for the product is again described with the pre- and post-entry functions $S^*(t)$ and $I_L^*(t)$ and $S(t)$, $I_L(t)$, $I_F(t)$ and $I_{LF}(t)$. These functions are again found by solving the initial value problems for the pre- and post-entry markets respectively.

5.2.1 Pre-entry initial value problem

The initial value problem describing the pre-entry market evolution is equivalent to the initial value problem (4.1) of the monopoly model. The pre-entry initial value problem again consists of differential equations that describe how the functions $S^*(t)$ and $I_L^*(t)$ evolve over time and the initial values $S^*(0) = A$ and $I_L^*(0) = B$ at the starting time $t = 0$. The initial value problem takes the form (5.1):

$$\begin{aligned}
\dot{I}_L^*(t) &= \beta P_{S^*\to I_L^*}(a_L, p_L, I_L^*(t))S^*(t) - \lambda I_L^*(t) \\
\dot{S}^*(t) &= -\beta P_{S^*\to I_L^*}(a_L, p_L, I_L^*(t))S^*(t) + \lambda I_L^*(t) \\
\text{with} \quad & I_L^*(0) = A^*, S^*(0) = B^*
\end{aligned} \tag{5.1}$$

At time t the buying probability of the Leader's product depends on the product's price p_L and quality a_L and the amount of the product already in the market given by the function $I_L^*(t)$. The form of the buying probability $P_{S^*\to I_L^*}(a_L, p_L, I_L^*(t))$ for the Leader's product is given by equation (5.2) which is adopted from equation (2.15).

$$P_{S^*\to I_L^*}(a_L, p_L, I_L^*(t)) = \frac{\exp\left(\frac{a_L - p_L + a_L g(I_L^*(t))}{\mu}\right)}{1 + \exp\left(\frac{a_L - p_L + a_L g(I_L^*(t))}{\mu}\right)} \tag{5.2}$$

Leader determines the initial conditions of the initial value problem (5.1) by choosing how many free samples it gives out. Let q_L denote the number of free samples Leader gives out. The initial conditions naturally are then $I_L^*(0) = A^* = q_L$ and $S^*(0) = B^* = N - q_L$.

The terms in the initial value problem (5.1) again have intuitive interpretations: At time t a total of $\beta S^*(t)$ consumers who do not own the Leader's product arrive at the buying decision, and of these consumers a total of $\beta P_{S^* \to I_L^*}(a_L, p_L, I_L^*(t))S^*(t)$ consumers buy the Leader's product. Simultaneously, a total of $\lambda I^*(t)$ products break down. Thus at time t, $I^*(t)$ increases by $\beta P_{S^* \to I_L^*}(a_L, p_L, I_L^*(t))S^*(t)$ and decreases by $\lambda I^*(t)$.

The population again stays constant over time as $d/dt(S^*(t) + I_L^*(t)) = \dot{S}^*(t) + \dot{I}_L^*(t) = 0$. We therefore have the following identity:

$$S^*(t) + I_L^*(t) = N \tag{5.3}$$

The identity (5.3) again allows us to reduce the differential equation system of initial value problem (5.1) to a single differential equation. This observation is used in the numerical solution of initial value problem (5.1).

Finally, it is necessary to introduce a new variable to describe the length of the pre-entry time period. Let time t_{Monopoly} denote the time Follower enters the market. Leader is then the only firm in the market on the time interval $[0, t_{\text{Monopoly}}]$, and the market evolution is described by the initial value problem (5.1) only on this time interval.

5.2.2 Post-entry initial value problem

The construction of the post-entry initial value problem is very much similar to the construction of the pre-entry initial value problem. Determining the buying probabilities now requires some extra care while the arrival and breakdown intensities remain exactly the same. For sake of completeness, let us first construct the buying probabilities from their simplest components.

The buying probabilities at time t are dependent on the quality of both products a_L and a_F and the prices of the products p_L and p_F and how many other consumers already own the products, i.e. $I_L(t)$, $I_F(t)$ and $I_{LF}(t)$. To shorten the notation, I denote the qualities as $\bar{a} = (a_1, a_2)$, the prices as $\bar{p} = (p_1, p_2)$ and the state of the market at time t as $\bar{I}(t) = (I_L(t), I_F(t), I_{LF}(t))$.

I define the terms $U_{i \to j}(\bar{a}, \bar{p}, \bar{I}(t)), i \in \{S, I_L, I_F\}, j \in \{I_L, I_F, I_{LF}\}$ in equations (5.4) to shorten the notation for the buying probabilities.

$$\begin{aligned}
U_{S \to I_L}(\bar{a}, \bar{p}, \bar{I}(t)) &= \exp\left(\frac{a_L - p_L + a_L g(I_L(t) + I_{LF}(t))}{\mu}\right) \\
U_{S \to I_F}(\bar{a}, \bar{p}, \bar{I}(t)) &= \exp\left(\frac{a_F - p_F + a_F g(I_F(t) + I_{LF}(t))}{\mu}\right) \\
U_{I_F \to I_{LF}}(\bar{a}, \bar{p}, \bar{I}(t)) &= \exp\left(\frac{\max\{a_L - a_F, 0\} - p_L + a_L g(I_L(t) + I_{LF}(t))}{\mu}\right) \\
U_{I_L \to I_{LF}}(\bar{a}, \bar{p}, \bar{I}(t)) &= \exp\left(\frac{\max\{a_F - a_L, 0\} - p_F + a_F g(I_F(t) + I_{LF}(t))}{\mu}\right)
\end{aligned} \tag{5.4}$$

By using the terms $U_{i \to j}(\bar{a}, \bar{p}, \bar{I}(t)), i \in \{S, I_L, I_F\}, j \in \{I_L, I_F, I_{LF}\}$ we may now define the actual buying probabilities $P_{i \to j}(\bar{a}, \bar{p}, \bar{I}(t)), i \in \{S, I_L, I_F\}, j \in \{I_L, I_F, I_{LF}\}$ in equations (5.5):

$$
\begin{aligned}
P_{S\to I_L}(\bar{a},\bar{p},\bar{I}(t)) &= \frac{U_{S\to I_L}(\bar{a},\bar{p},\bar{I}(t))}{1+U_{S\to I_L}(\bar{a},\bar{p},\bar{I}(t))+U_{S\to I_F}(\bar{a},\bar{p},\bar{I}(t))}\\
P_{I_F\to I_{LF}}(\bar{a},\bar{p},\bar{I}(t)) &= \frac{U_{I_F\to I_{LF}}(\bar{a},\bar{p},\bar{I}(t))}{1+U_{I_F\to I_{LF}}(\bar{a},\bar{p},\bar{I}(t))}\\
P_{S\to I_F}(\bar{a},\bar{p},\bar{I}(t)) &= \frac{U_{S\to I_F}(\bar{a},\bar{p},\bar{I}(t))}{1+U_{S\to I_L}(\bar{a},\bar{p},\bar{I}(t))+U_{S\to I_F}(\bar{a},\bar{p},\bar{I}(t))}\\
P_{I_L\to I_{LF}}(\bar{a},\bar{p},\bar{I}(t)) &= \frac{U_{I_L\to I_{LF}}(\bar{a},\bar{p},\bar{I}(t))}{1+U_{I_L\to I_{LF}}(\bar{a},\bar{p},\bar{I}(t))} \qquad (5.5)
\end{aligned}
$$

The buying probabilities described in equations (5.5) are equivalent to the probabilities in equations (2.16) and (2.17).

We may now write the initial value problem describing the market evolution for the post-entry time period. The initial value problem again consists of the differential equations that describe how the functions $S(t)$, $I_L(t)$, $I_F(t)$ and $I_{LF}(t)$ evolve over time and of the initial values $S(t_{\text{Entry}}) = A$, $I_L(t_{\text{Entry}}) = B$, $I_F(t_{\text{Entry}}) = C$ and $I_{LF}(t_{\text{Entry}}) = D$ at the starting time $t = t_{\text{Entry}}$. The initial value problem now takes the form (5.6):

$$
\begin{aligned}
\dot{S}(t) &= -\beta P_{S\to I_L}(\bar{a},\bar{p},\bar{I}(t))S(t) - \beta P_{S\to I_F}(\bar{a},\bar{p},\bar{I}(t))S(t) + \lambda I_L(t) + \lambda I_F(t)\\
\dot{I}_L(t) &= \beta P_{S\to I_L}(\bar{a},\bar{p},\bar{I}(t))S(t) - \beta P_{I_L\to I_{LF}}(\bar{a},\bar{p},\bar{I}(t))I_L(t) + \lambda I_{LF}(t) - \lambda I_L(t)\\
\dot{I}_F(t) &= \beta P_{S\to I_F}(\bar{a},\bar{p},\bar{I}(t))S(t) - \beta P_{I_F\to I_{LF}}(\bar{a},\bar{p},\bar{I}(t))I_F(t) + \lambda I_{LF}(t) - \lambda I_F(t)\\
\dot{I}_{LF}(t) &= \beta P_{I_L\to I_{LF}}(\bar{a},\bar{p},\bar{I}(t))I_L(t) + \beta P_{I_F\to I_{LF}}(\bar{a},\bar{p},\bar{I}(t))I_F(t) - 2\lambda I_{LF}(t)\\
\text{with} &\quad S(t_{\text{Entry}}) = A, I_L(t_{\text{Entry}}) = B, I_F(t_{\text{Entry}}) = C, I_{LF}(t_{\text{Entry}}) = D \qquad (5.6)
\end{aligned}
$$

The starting time t_{Entry} is different for Leader and Follower respectively. For Leader, the market structure changes at time $t = t_{\text{Monopoly}}$ with Follower's entry, meaning that we have the starting time $t_{\text{Entry}} = t_{\text{Monopoly}}$ for Leader. For Follower, we have the starting time $t_{\text{Entry}} = 0$, as in the Follower's point of view there is no history prior to entry.

The initial values of initial value problem (5.6) are determined by the pre-entry evolution of the market and the free samples choices of Follower. The pre-entry evolution is in turn described by the solutions of the initial value problem (5.1) which is in turn determined by the choices of Leader. Suppose that we have the functions $S^*(t)$ and $I^*(t)$ that are solutions of the initial value problem (5.1). Then $S^*(t_{\text{Monopoly}})$ denotes the number of consumers who do not have the Leader's product at the time of Follower's entry. Correspondingly, $I_L^*(t_{\text{Monopoly}})$ denotes the number of consumer who have the Leader's product at the time of Follower's entry. Next, I assume that Follower cannot observe whether a consumer has or does not have the Leader's product. Follower therefore gives out free samples randomly with every consumer having an equal probability of receiving a product. The initial values then take the form (5.7):

$$
A = S^*(t_{\text{Monopoly}})(1-\frac{q_F}{N}), B = I_L^*(t_{\text{Monopoly}})(1-\frac{q_F}{N}), C = S^*(t_{\text{Monopoly}})\frac{q_F}{N}, D = I_L^*(t_{\text{Monopoly}})\frac{q_F}{N} \qquad (5.7)
$$

The terms in the differential equation system of the initial value problem (5.6) again have natural interpretations: Due to purchases, the consumers move from compartment S to compartment I_L at rate $\beta P_{S\to I_L}(\bar{a},\bar{p},\bar{I}(t))S(t)$ and to compartment I_F at rate $\beta P_{S\to I_F}(\bar{a},\bar{p},\bar{I}(t))S(t)$. Again due to purchases, the consumers move to compartment I_{LF} at rate $\beta P_{I_L\to I_{LF}}(\bar{a},\bar{p},\bar{I}(t))I_L(t)$ from compartment I_L and at rate $\beta P_{I_F\to I_{LF}}(\bar{a},\bar{p},\bar{I}(t))I_F(t)$ from compartment I_F. At the same time the products are breaking down at intensity λ. Then due to product breakdowns, consumers move from compartments I_L and I_F to compartment S at rates $\lambda I_L(t)$ and $\lambda I_F(t)$ respectively. Additionally, since the consumers in compartment I_{LF} own both products, they move from this compartment at rate $2\lambda I_{LF}(t)$ to compartments I_L and I_F.

The formulation of initial value problem (5.6) implies that both firms' products have the same expected lifetime as the breakdown intensity λ is the same for both firms. I interpret the parameter λ as reflecting the "state of technology" in production, with both firms choosing the best available technology to provide the most durable product. As the same technology is available to both firms, they both make equally durable products leading to the same breakdown intensity λ.

The population again stays constant over time as $d/dt(S(t)+I_L+I_F+I_{LF}) = \dot{S}(t)+\dot{I}_L(t)+\dot{I}_F(t)+\dot{I}_{LF}(t) = 0$. This leads to identity (5.8):

$$S(t) + I_L(t) + I_F(t) + I_{LF}(t) = N \tag{5.8}$$

As before, the identity (5.8) may be used to eliminate one of the differential equations in the initial value problem (5.6). This assumption is again used in the numerical solution of initial value problem (5.6).

5.3 Firm behavior

The behavior of both firms is determined by their profit maximization objective. The profits of both firms depend also on the choices of the other firm through initial value problem (5.6). This dependence forces both firms to consider how their own choices will affect the choices of the other firm. The sequential entry of firms allows us describe this choice-making process explicitly. Specifically, we may write the firms' problems in such a way that the strategic thinking behind the choices becomes explicit. The firms' problems again consist of an objective function, decision variables and constraints. Both of the problems are constructed by defining these components.

The choice-making process can be seen as a game with two stages. At the first stage, Leader makes its choices. At the second stage, Follower observes the choices of Leader and makes its own choices. We note that during each stage only a single player chooses an action whilst knowing the entire history up to that stage, i.e. the choices other players have made previously. We therefore use backward induction implicitly as it is the natural approach in solving games of this type. We first start to study the choices of Follower as it is the last player to act. If we can deduce how Follower will act given the choices of Leader, we may implement this behavior to the Leader's problem. This approach is equivalent to assuming that Leader can deduce how its choices will induce the choices of Follower.

5.3.1 Follower's problem

First let us describe the information of Follower. Prior to Follower's entry, Leader has chosen its quality a_L, free samples q_L and price p_L. The market has then evolved to some state based on Leader's choices. I assume that Follower is able to observe or calculate the state of the market at time of entry, i.e. Follower observes the values $S^*(t_{\text{Monopoly}})$ and $I_L^*(t_{\text{Monopoly}})$ at time t_{Monopoly}. While it has been previously stated that observing the state of the market is difficult, I nevertheless offer the following rationalization for this assumption: The market state is observable to Follower because it is an operator within the market while the market state remains opaque to the outside observer. Naturally in addition to the market state, Follower knows the choices of Leader. Follower bases its choices on this information.

Follower chooses its own quality a_F, free samples q_F and price p_F which are thus the decision variables of the Follower's problem. Naturally Follower may not choose a negative quality or price nor give out a negative amount of free samples. We thus have the natural constraints $a_F, q_F, p_F \geq 0$ for the Follower's problem.

Let us next determine the objective function of Follower. Follower seeks to maximize its profits which are determined by the demand of its product. The term $\beta P_{S\to I_F}(\bar{a}, \bar{p}, \bar{I}(t))S(t)$ describes the demand for the Follower's product from those consumers who have neither product at time t, and correspondingly the term $\beta P_{I_L\to I_{LF}}(\bar{a}, \bar{p}, \bar{I}(t))I_L(t)$ describes the demand for the Follower's product from those who have the Leader's product at time t. The per-unit discounted profits at time t of Follower are described by the term $\exp(-rt)(p_F - C(a_F))$. The total profits are then calculated by integrating the product term $\exp(-rt)(p_F - C(a_F))[\beta P_{S\to I_F}(\bar{a}, \bar{p}, \bar{I}(t))S(t) + \beta P_{I_L\to I_{LF}}(\bar{a}, \bar{p}, \bar{I}(t))I_L(t)]$ over the planning horizon $[0, T]$. After accounting for the Follower's free sample costs $C(a_F)q_F$, Follower's profits π^{Follower} are given by equation (5.9):

$$\begin{aligned} \pi^{\text{Follower}}(a_F, q_F, p_F \mid a_L, q_L, p_L) \quad = \quad & \int_0^T \exp(-rt)(p_F - C(a_F))\big[\beta P_{S\to I_F}(\bar{a}, \bar{p}, \bar{I}(t))S(t) + \\ & \beta P_{I_L\to I_{LF}}(\bar{a}, \bar{p}, \bar{I}(t))I_L(t)\big]\,dt - C(a_F)q_F \end{aligned} \tag{5.9}$$

Follower operates within the post-entry market structure. The functions $S(t), I_F(t), I_L(t)$ and $I_{LF}(t)$ affecting the profits (5.9) are therefore determined by the initial value problem (5.6). This gives us the final constraint for Follower's problem: Functions $S(t), I_L(t), I_F(t), I_{LF}(t)$ are solutions of initial value problem 5.6 over the interval $[0, T]$.

We may now write Follower's problem (5.10):

$$\begin{aligned} \max_{a_F, p_F, q_F} \quad & \pi^{\text{Follower}}(a_F, q_F, p_F \mid a_L, q_L, p_L) \qquad \text{s.t.} \\ & S(t), I_L(t), I_F(t), I_{LF}(t) \text{ solutions of initial value problem (5.6) on } [0, T], \\ & a_F, q_F, p_F \geq 0 \end{aligned} \tag{5.10}$$

Let us next define the Follower's three best-response functions that will aid in constructing Leader's problem. Given the choices (a_L, q_L, p_L) of Leader, each best-response function describes Follower's optimal choice of a decision variable. We define the best-response functions as $\text{BR}_a^{\text{Follower}}(a_L, q_L, p_L) = a_F^*$, $\text{BR}_q^{\text{Follower}}(a_L, q_L, p_L) = q_F^*$ and $\text{BR}_p^{\text{Follower}}(a_L, q_L, p_L) = p_F^*$ such that quality a_F^*, free samples q_F^* and price p_F^* are a solution of the Follower's problem (5.10) given the Leader's choices (a_L, q_L, p_L). By assumption Leader knows the form of the Follower's problem and that Follower will best-respond to its own choices. We may then use these functions in constructing the Leader's problem to account for this knowledge.

5.3.2 Leader's problem

We construct the Leader's problem by utilizing the knowledge we assume Leader to have regarding Follower's behavior, i.e. Leader's information. We have assumed that Leader knows time of Follower's entry and the form of the Follower's problem. This knowledge allows Leader to anticipate the Follower's choices given its own choices. These assumptions are questionably strong, but arguably also in line with the assumptions of the classical Stackelberg model. Leader makes its choices with this information in mind.

Leader chooses the quality a_L and the price p_L of its product and how many free samples q_L to give out. These are then the natural decision variables of the Leader's problem. Naturally Leader may not set a negative quality or price nor give out a negative amount of free samples. This gives us the natural constraints $a_L, q_L, p_L \geq 0$ on the decision variables.

Leader's profits now depend on the demand of its product from two different time periods. These time periods are defined by the Follower's presence in the market: First Leader operates alone in the pre-entry market. Next, after the Follower's entry, Leader operates in the post-entry market together with Follower. Constructing Leader's objective function requires us then to first construct the Leader's profits from the pre-entry and the post-entry periods.

Let us now construct the pre-entry profits. The demand for Leader's product at time t is given by the term $\beta P_{S^* \to I_L^*}(a_L, p_L, I_L^*(t))S_L^*(t)$. The per-unit discounted profits are given by term $\exp(-rt)(p_L - C(a_L))$. The discounted profits for this time period are then calculated by integrating the product of these terms over the interval $[0, t_{\text{Monopoly}}]$. Leader's pre-entry profits are then given by equation (5.11):

$$\pi^{\text{Leader}}_{\text{Pre-Entry}}(a_L, q_L, p_L) = \int_0^{t_{\text{Monopoly}}} \exp(-rt)(p_L - C(a_L))\beta P_{S^* \to I_L^*}(a_L, p_L, I_L^*(t))S_L^*(t)dt \tag{5.11}$$

Let us next construct the post-entry profits. The Follower's entry changes the market structure to the post-entry market structure described by initial value problem (5.6). The term $\beta P_{S \to I_L}(\bar{a}, \bar{p}, \bar{I}(t))S(t)$ then describes the demand for Leader's product from those consumers who have neither product at time t, and the term $\beta P_{I_F \to I_{LF}}(\bar{a}, \bar{p}, \bar{I}(t))I_F(t)$ the demand from those consumers who have the Follower's product at time t. The total demand at time t is then given by the sum $\beta P_{S \to I_L}(\bar{a}, \bar{p}, \bar{I}(t))S(t) + \beta P_{I_F \to I_{LF}}(\bar{a}, \bar{p}, \bar{I}(t))I_F(t)$. The per-unit discounted profits are again given by term $\exp(-rt)(p_L - C(a_L))$. We may then calculate the profits by integrating the product of these terms over the interval $[t_{\text{Monopoly}}, T + t_{\text{Monopoly}}]$. Leader's post-entry profits are thus given by equation (5.12):

$$\pi^{\text{Leader}}_{\text{Post-Entry}}(a_L, q_L, p_L) = \int_{t_{\text{Monopoly}}}^{T+t_{\text{Monopoly}}} \exp(-rt)(p_L - C(a_L))[\beta P_{S \to I_L}(\bar{a}, \bar{p}, \bar{I}(t))S(t) + \beta P_{I_F \to I_{LF}}(\bar{a}, \bar{p}, \bar{I}(t))I_F(t)]dt \tag{5.12}$$

The total profits of Leader are given by the sum of the pre-entry and post-entry profits minus the sample costs $C(a_L)q_L$. The equations (5.11) and (5.12) then allow us to construct Leader's total profits π^{Leader} in equation (5.13) giving us the objective function of Leader:

$$\pi^{\text{Leader}}(a_L, q_L, p_L) = \pi^{\text{Leader}}_{\text{Pre-Entry}}(a_L, q_L, p_L) + \pi^{\text{Leader}}_{\text{Post-Entry}}(a_L, q_L, p_L) - C(a_L)q_L \tag{5.13}$$

As Leader operates both in the pre- and the post-entry markets, the pre- and post-entry market structures impose the following restrictions: The functions $S^*(t)$ and $I_L^*(t)$ are solutions of initial value problem (5.1) over the interval $[0, t_{\text{Monopoly}}]$, and the functions $S(t), I_L(t), I_F(t), I_{LF}(t)$ are solutions of initial value problem (5.6) over the interval $[t_{\text{Monopoly}}, T + t_{\text{Monopoly}}]$.

The Leader's problem requires a final set of constraints stemming from the game theoretical structure of the model. Leader anticipates that Follower will best-respond to its own choices and what these best responses will be. Using the previously defined best-response functions we have the constraints $a_F = \text{BR}_{a,F}(a_L, q_L, p_L)$, $q_F = \text{BR}_{q,F}(a_L, q_L, p_L)$ and $p_F = \text{BR}_{p,F}(a_L, q_L, p_L)$ to signify that the Follower's choices (a_F, q_F, p_F) are indeed the best responses to the Leader's choices (a_L, q_L, p_L).

Leader's problem then has the form (5.14):

$$\max_{a_L, p_L, q_L} \quad \pi^{\text{Leader}}(a_L, q_L, p_L) \quad \text{s.t.}$$
$$S^*(t), I_L^*(t) \text{ solutions of initial value problem (5.1) on } [0, t_{\text{Monopoly}}],$$
$$S(t), I_L(t), I_F(t), I_{LF}(t) \text{ solutions of initial value problem (5.6) on } [t_{\text{Monopoly}}, T + t_{\text{Monopoly}}],$$
$$a_F = \text{BR}_a^{\text{Follower}}(a_L, q_L, p_L), q_F = \text{BR}_q^{\text{Follower}}(a_L, q_L, p_L), p_F = \text{BR}_p^{\text{Follower}}(a_L, q_L, p_L),$$
$$a_L, q_L, p_L \geq 0 \tag{5.14}$$

The structure of the Leader's problem (5.14) hints at the way the problem may be solved without resorting to finite choice sets. With the use of best-response functions the problem becomes a matter of constrained optimization. If the initial value problems (5.1) and (5.6) and the integrals (5.9), (5.11) and (5.12) had easy analytical solutions and we could find the forms of the best-response functions, the problem (5.14) itself could perhaps be solved by using the KKT conditions. Should we wish to apply more sophisticated numerical optimization methods to the Leader's problem, we would need a more imaginative approach as the best-response functions makes it difficult to, for example, calculate gradients numerically.

The choices given by the solution of the Leader's problem (5.14) form a Nash equilibrium as the solution first gives us the optimal choices of Leader to which Follower best-responds, i.e. makes its own choices optimally given the choices of Leader. Neither firm can then improve its profits by unilaterally deviating from the choices of the solution.

5.3.3 Solving the Leader's problem

As Follower best-responds to the Leader's choices, we note that the solution of the Leader's problem will also describe the Follower's behavior. Solving the Leader's problem is then sufficient to determine the behavior of both firms. We therefore limit our attention to it.

I use a simple nested brute force algorithm to find the solution of Leader's problem for the finite choice sets (3.8). With finite choice sets, the number of all possible choice combinations is also finite allowing us to calculate the Leader's profits for all of these to find the combination with the largest profits. However, as Follower best-responds to all of the Leader's choices and thereby affects the Leader's profits, it is necessary to solve the Follower's problem (5.10) for each of the Leader's choice combinations. The Follower's problem can be seen to be a slight modification of the monopoly's problem (4.5) and is therefore solved similarly by calculating the Follower's profits for all possible choice combinations of Follower to find the combination with the largest profits. Thus brute force is used in two instances: firstly to calculate the Leader's profits for Leader's all possible choice combinations, and secondly to calculate the Follower's profits for Follower's all possible choice combinations given a single choice combination of Leader. See appendix D for the script.

Figure 5.3 gives the structure of the implemented script. I divide the structure of the script into three parts based on the role that each part plays in solving the Leader's problem (5.14). These parts are calculating the Leader's pre-entry profits, which comprises of the subroutines `preentryProfitsLeader`, `monopolyDE` and `MonopolyRevenue`, calculating the Follower's best responses to the Leader's choices, which comprises of the subroutines `FollowerOptimizes`, `profitsFollower`, `duopolyDES` and `revenueFollower`, and calculating the Follower's post-entry profits, which comprises of the subroutines `profitsLeader`, `duopolyDES` and `revenueLeader`. The subroutine `LeaderOptimizes` forms the backbone of the script as it utilizes the presented subroutines to calculate the Leader's profits (5.13) for all Leader's choice combinations, and after which it chooses the combination with the highest profits thereby solving the Leader's problem.

The subroutine `preentryProfitsLeader` calculates the Leader's pre-entry profits (5.11) for a given quality, free samples and price choice. It solves the initial value problem (5.6) with *ode45*, generates a revenue vector and uses the vector to calculate the integral determining the pre-entry profits. Additionally, the state of the pre-entry market at the time of entry is saved in order to determine the initial values of initial value problem (5.6). The subroutine `monopolyDE` is the reduced differential equation system of the initial value problem (5.1). The subroutine `MonopolyRevenue` is used to generate the revenue vector for numerical integration.

The Follower's best responses are calculated using the method for solving the monopoly's problem (with finite choice sets), as the Follower's problem (5.10) can be seen, given the Leader's choices, as the monopoly's problem (4.5) with a different initial value problem. The subroutine `FollowerOptimizes` calculates the Follower's profits for all combinations of quality, free samples and price, and chooses the combination with the Follower's highest profits. It uses the subroutine `profitsFollower` to calculate the profits for a given combination of quality, free samples and price. The subroutine `profitsFollower` in turn solves the initial value problem (5.6) using *ode45*, generates a revenue vector for the integral in equation (5.9), uses *trapz* to calculate the integral and determines the profits after accounting for the Follower's free sample costs. The subroutine `duopolyDES` is the (reduced) differential equation system of initial value problem (5.6) required by *ode45*. The subroutine `revenueFollower` is used to generate the revenue vector for numerical integration.

Finally, the post-entry profits of Leader are calculated by the subroutine `profitsLeader`. The calculation method is the same as in the subroutine `profitsFollower`: For the given choices of Leader and the Follower's best responses to these choices, the subroutine `profitsLeader` solves the initial value problem (5.6) using *ode45*, generates a revenue vector for the integral in equation (5.12) and uses *trapz* to calculate the integral. The subroutine `duopolyDES` is the same (reduced) differential equation system of initial value problem (5.6) required by *ode45*. The subroutine `revenueLeader` is used to generate the revenue vector for numerical integration.

5.4 Outcomes

5.4.1 New parameter choices

The model requires an additional, model-specific parameter t_{Monopoly} that has not been discussed in chapter 3. I chose to use the value $t_{\text{Monopoly}} = 1$. This value was chosen to model a case where Leader has a relatively small amount of monopoly time considering the planning horizons $T = 10$ and $T = 100$.

5.4.2 Optimal firm behavior

Tables 5.1-5.3 give the behavior of Leader and Follower.

The results do not offer us a reason to doubt the internal validity of the model. We note that both firms always set a positive price and get a positive profit. The parameters r and T also have the expected effects on profits. First, we note that in the no trend cases both firms always choose $(a_i, q_i, p_i) = (2, 0, 3), i = L, F$ providing us a way to study the effects of parameters r and T. We then note that profits of both firms are larger with the longer planning horizon $T = 100$ than with the shorter planning horizon $T = 10$. Likewise, the profits of both firms are larger with smaller discount rates.

Rather surprisingly, we may note that there is no universal first-mover advantage in this model. With a long planning horizon $T = 100$, a small parabel trend with strength $d = 2$ and a reasonable discount rate $r = 0$ or $r = 0.05$, Leader chooses to produce cheaper, "less promoted" (i.e. with less free samples) product with a lower price ($a_L = 1 < 3 = a_F$, $q_L = 200\,000 < 400\,000 = q_F$ and $p_L = 3 < 9 = p_F$) that leads to greatly smaller profits compared with Follower ($3.35463 \cdot 10^7 < 6.40932 \cdot 10^7$ and $6.66545 \cdot 10^6 < 1.10186 \cdot 10^7$). The result

Planning horizon	Trend type	Leader (a_L, q_L, p_L)	Leader Profits	Follower (a_F, q_F, p_F)	Follower Profits
$T = 10$	no trend	$(2, 0, 3)$	447156	$(2, 0, 3)$	390692
	linear, $d = 2$	$(2, 200\,000, 3)$	$1.41643 \cdot 10^6$	$(0, 0, 3)$	40590.9
	parabel, $d = 2$	$(2, 200\,000, 6)$	$4.04282 \cdot 10^6$	$(2, 400\,000, 6)$	$4.8304 \cdot 10^6$
	linear, $d = 8$	$(3, 400\,000, 9)$	$1.33737 \cdot 10^7$	$(3, 400\,000, 9)$	$8.90953 \cdot 10^6$
	parabel, $d = 8$	$(2, 200\,000, 9)$	$2.07803 \cdot 10^7$	$(2, 200\,000, 9)$	$1.90821 \cdot 10^7$
$T = 100$	no trend	$(2, 0, 3)$	$3.82225 \cdot 10^6$	$(2, 0, 3)$	$3.76579 \cdot 10^6$
	linear, $d = 2$	$(2, 600\,000, 3)$	$1.51397 \cdot 10^7$	$(0, 0, 3)$	413134
	parabel, $d = 2$	$(1, 200\,000, 3)$	$3.35463 \cdot 10^7$	$(3, 400\,000, 9)$	$6.40932 \cdot 10^7$
	linear, $d = 8$	$(3, 400\,000, 9)$	$1.33333 \cdot 10^8$	$(3, 400\,000, 9)$	$1.04931 \cdot 10^8$
	parabel, $d = 8$	$(2, 200\,000, 9)$	$1.80691 \cdot 10^8$	$(2, 200\,000, 9)$	$1.78989 \cdot 10^8$

Table 5.1: Patient firms with $r = 0$, Choices and profits

Planning horizon	Trend type	Leader (a_L, q_L, p_L)	Leader Profits	Follower (a_F, q_F, p_F)	Follower Profits
$T = 10$	no trend	$(2, 0, 3)$	349776	$(2, 0, 3)$	309763
	linear, $d = 2$	$(2, 200\,000, 3)$	991930	$(0, 0, 3)$	31814.9
	parabel, $d = 2$	$(2, 400\,000, 6)$	$3.53393 \cdot 10^6$	$(2, 400\,000, 6)$	$2.62231 \cdot 10^6$
	linear, $d = 8$	$(3, 400\,000, 9)$	$9.96657 \cdot 10^6$	$(3, 400\,000, 9)$	$6.62314 \cdot 10^6$
	parabel, $d = 8$	$(2, 200\,000, 9)$	$1.62338 \cdot 10^7$	$(2, 200\,000, 9)$	$1.52606 \cdot 10^7$
$T = 100$	no trend	$(2, 0, 3)$	777698	$(2, 0, 3)$	759625
	linear, $d = 2$	$(2, 600\,000, 3)$	$1.78793 \cdot 10^6$	$(0, 0, 3)$	82044.8
	parabel, $d = 2$	$(1, 200\,000, 3)$	$6.66545 \cdot 10^6$	$(3, 400\,000, 9)$	$1.10186 \cdot 10^7$
	linear, $d = 8$	$(3, 400\,000, 9)$	$2.51758 \cdot 10^7$	$(3, 400\,000, 9)$	$1.94214 \cdot 10^7$
	parabel, $d = 8$	$(2, 200\,000, 9)$	$3.65084 \cdot 10^7$	$(2, 200\,000, 9)$	$3.6574 \cdot 10^7$

Table 5.2: Normal firms with $r = 0.05$, Choices and profits

is very peculiar and seemingly valid as I was unable to find any obvious mistakes that would lead to the described behavior.

Let us now consider the research questions: Different trend types indeed lead to observable differences in firm behavior but the specific type of the trend may not be inferred purely from firm behavior. For example, suppose that we observe Leader's and Follower's behavior as $(a_L, q_L, p_L) = (a_F, q_F, p_F) = (2, 0, 3)$ and we are asked to give an opinion on the existence of trend. Based on tables 5.1-5.3, we would not be able to distinguish between a no trend case or a linear case trend with impatient firms (see table 5.3, $T = 10$, $d = 2$).

We also note that there indeed cases with trends such that one or both firms optimally give out products as free samples. We now again have an "existence result" for a trend, namely if we observe a firm giving out free samples we may infer the existence of some trend. As before, this result does not run in the other direction, i.e. we may not infer the nonexistence of a trend from observing that neither firm gives out free samples.

Planning horizon	Trend type	Leader (a_L, q_L, p_L)	Leader Profits	Follower (a_F, q_F, p_F)	Follower Profits
$T = 10$	no trend	$(2, 0, 3)$	5898.34	$(2, 0, 3)$	4793.41
	linear, $d = 2$	$(2, 0, 3)$	6432.83	$(2, 0, 3)$	3656.58
	parabel, $d = 2$	$(2, 0, 3)$	13306.3	$(0, 0, 3)$	312.674
	linear, $d = 8$	$(2, 0, 3)$	10160.7	$(0, 0, 3)$	295.01
	parabel, $d = 8$	$(1, 200\,000, 6)$	166082	$(1, 200\,000, 6)$	147940
$T = 100$	no trend	$(2, 0, 3)$	5898.34	$(2, 0, 3)$	4793.41
	linear, $d = 2$	$(2, 0, 3)$	6432.83	$(2, 0, 3)$	3656.58
	parabel, $d = 2$	$(2, 0, 3)$	13306.3	$(0, 0, 3)$	312.673
	linear, $d = 8$	$(2, 0, 3)$	10160.7	$(0, 0, 3)$	295.01
	parabel, $d = 8$	$(1, 200\,000, 6)$	166082	$(1, 200\,000, 6)$	147939

Table 5.3: Impatient firms with $r = 10$, Choices and profits

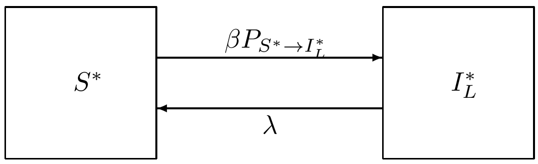


Figure 5.1: Compartmental Graph, Pre-entry

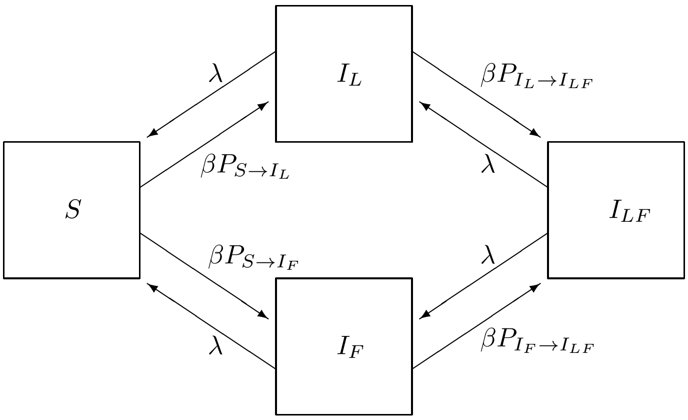


Figure 5.2: Compartmental Graph, Post-entry

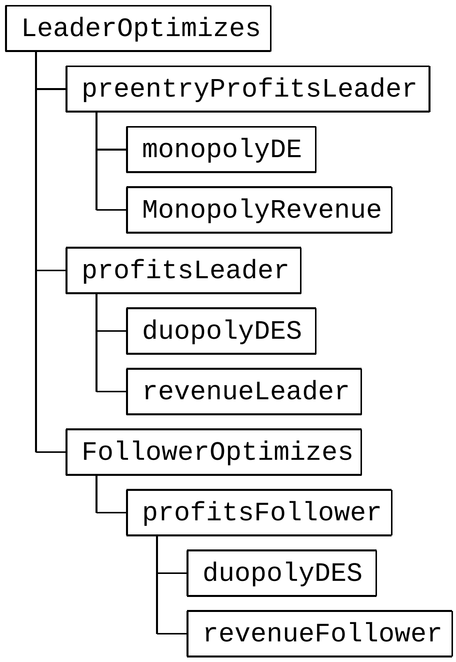


Figure 5.3: Script structure, Stackelberg duopoly

6

Simultaneous-choice Duopoly

This chapter presents the second of the duopoly models of this thesis. Where as in the Stackelberg duopoly model the firms chose their product's quality, free samples and price sequentially, these choices in this model are made simultaneously similarly to the classical Cournot and Bertrand duopoly models. The key characteristics of the model are as follows: There are two firms, firm 1 and firm 2, who sell durable goods to a large population of consumers. There is no secondary market for the products of either firm. A consumer will buy only a single product from a particular firm and hold it until the product breaks down. The consumer may choose to buy the products of both firms. Both firms choose the quality and price of their products, and whether to give out products as free samples. Prices and qualities are set once and for all, and free samples are given only at the time of entry into the market. Both firms seek to maximize their profits and will make their choices accordingly. The interaction between the firms is modeled as a game.

The demand for the both firms' products are determined through a initial value problem inspired by the SIS model. The firms determine the initial value problem through their choices. Differing from the two preceeding models, we use the initial value problem to calculate payoffs for the interaction game rather than implementing it to the firms' optimization problems.

The interaction game is a multi-stage game with three stages which determines the firms' behavior by its subgame-perfect Nash equilibrium (SPNE). The SPNE of the game is constructed by solving Nash equilibria (NEa) in the three stages, as each possible stage can be seen as an individual normal-form game. The range of possible observable firm behavior may be seen from the probability distributions that form the constructed SPNE.

6.1 Market Structure

The market structure of this model is equivalent to the post-entry market structure of the Stackelberg duopoly model. The population is divided into four different compartments: S, I_1, I_2 and I_{12} with the respective functions $S(t)$, $I_1(t)$, $I_2(t)$ and $I_{12}(t)$. The function $S(t)$ describes the amount of people who have neither product, the function $I_1(t)$ the amount of people who own a product from firm 1, the function $I_2(t)$ the amount of people who own a product from firm 2 and the function $I_{12}(t)$ the amount of people who own a product from both firms at time t. Figure 6.1 describes the compartmental structure and the movement between compartments. The movement between the compartments is determined by the breakdown and arrival intensities λ and β and the buying probabilities $P_{i\to j}, i \in \{S, I_1, I_2\}, j \in \{I_1, I_2, I_{12}\}$.

As with the two previous models, we wish to determine the profits of both firms given their choices of quality, free samples and price. The profits are determined by the products' demands which can be described using the functions

$S(t)$, $I_1(t)$, $I_2(t)$ and $I_{12}(t)$. The functions are again found by solving the initial value problem describing the market.

6.2 Initial value problem

The initial value problem of this model is almost equivalent to the post-entry initial value problem of the Stackelberg duopoly model. The differential equations in both initial value problems are the same after changing the names L, F, LF to $1, 2, 12$. However, the initial conditions are now slightly different. Whereas the construction of the initial conditions of the post-entry initial value problem of the Stackelberg duopoly model required the solution of the pre-entry initial value problem, the initial conditions of the initial value problem of the simultaneous-choice duopoly model are constructed simply from the choices of firms 1 and 2. For the sake of completeness, let us next reiterate the construction of the differential equations.

The buying probabilities $P_{i\to j}, i \in \{S, I_1, I_2\}, j \in \{I_1, I_2, I_{12}\}$ at time t are dependent on three factors: the qualities a_1 and a_2, the prices p_1 and p_2 of firms 1 and 2 and how many other consumers already own the product at that time, i.e. the state of the market $I_1(t)$, $I_2(t)$ and $I_{12}(t)$. To shorten the notation, I choose to denote the qualities as $\bar{a} = (a_1, a_2)$, the prices as $\bar{p} = (p_1, p_2)$ and the state of the market at time t as $\bar{I}(t) = (I_1(t), I_2(t), I_{12}(t))$.

I again define terms $U_{i\to j}(\bar{a}, \bar{p}, \bar{I}(t)), i \in \{S, I_1, I_2\}, j \in \{I_1, I_2, I_{12}\}$ in equations (6.1) to shorten the notation for the buying probabilities.

$$
\begin{aligned}
U_{S\to I_1}(\bar{a}, \bar{p}, \bar{I}(t)) &= \exp\left(\frac{a_1 - p_1 + a_1 g(I_1(t) + I_{12}(t))}{\mu}\right) \\
U_{S\to I_2}(\bar{a}, \bar{p}, \bar{I}(t)) &= \exp\left(\frac{a_2 - p_2 + a_2 g(I_2(t) + I_{12}(t))}{\mu}\right) \\
U_{I_2\to I_{12}}(\bar{a}, \bar{p}, \bar{I}(t)) &= \exp\left(\frac{\max\{a_1 - a_2, 0\} - p_1 + a_1 g(I_1(t) + I_{12}(t))}{\mu}\right) \\
U_{I_1\to I_{12}}(\bar{a}, \bar{p}, \bar{I}(t)) &= \exp\left(\frac{\max\{a_2 - a_1, 0\} - p_2 + a_2 g(I_2(t) + I_{12}(t))}{\mu}\right)
\end{aligned} \tag{6.1}
$$

With the terms $U_{i\to j}(\bar{a}, \bar{p}, \bar{I}(t), i \in \{S, I_1, I_2\}, j \in \{I_1, I_2, I_{12}\}$ we may now define the actual buying probabilities $P_{i\to j}(\bar{a}, \bar{p}, \bar{I}(t)), i \in \{S, I_1, I_2\}, j \in \{I_1, I_2, I_{12}\}$ in equations (6.2):

$$
\begin{aligned}
P_{S\to I_1}(\bar{a}, \bar{p}, \bar{I}(t)) &= \frac{U_{S\to I_1}(\bar{a}, \bar{p}, \bar{I}(t))}{1 + U_{S\to I_1}(\bar{a}, \bar{p}, \bar{I}(t)) + U_{S\to I_2}(\bar{a}, \bar{p}, \bar{I}(t))} \\
P_{I_2\to I_{12}}(\bar{a}, \bar{p}, \bar{I}(t)) &= \frac{U_{I_2\to I_{12}}(\bar{a}, \bar{p}, \bar{I}(t))}{1 + U_{I_2\to I_{12}}(\bar{a}, \bar{p}, \bar{I}(t))} \\
P_{S\to I_2}(\bar{a}, \bar{p}, \bar{I}(t)) &= \frac{U_{S\to I_2}(\bar{a}, \bar{p}, \bar{I}(t))}{1 + U_{S\to I_1}(\bar{a}, \bar{p}, \bar{I}(t)) + U_{S\to I_2}(\bar{a}, \bar{p}, \bar{I}(t))} \\
P_{I_1\to I_{12}}(\bar{a}, \bar{p}, \bar{I}(t)) &= \frac{U_{I_1\to I_{12}}(\bar{a}, \bar{p}, \bar{I}(t))}{1 + U_{I_1\to I_{12}}(\bar{a}, \bar{p}, \bar{I}(t))}
\end{aligned} \tag{6.2}
$$

The buying probabilities in equations (6.2) are equivalent to the buying probabilities in (2.16) and (2.17).

After defining the buying probabilities we may write the initial value problem which describes the market evolution. The initial value problem consists once again of the differential equations that describe how the functions $S(t)$, $I_1(t)$, $I_2(t)$ and $I_{12}(t)$ evolve over time and of the initial values $S(0) = A$, $I_1(0) = B$, $I_2(0) = C$, $I_{12}(0) = D$ at the starting time $t = 0$. The initial value problem now takes the form (6.3):

$$\begin{aligned}
\dot{S}(t) &= -\beta P_{S\to I_1}(\bar{a},\bar{p},\bar{I}(t))S(t) - \beta P_{S\to I_2}(\bar{a},\bar{p},\bar{I}(t))S(t) + \lambda I_1(t) + \lambda I_2(t) \\
\dot{I}_1(t) &= \beta P_{S\to I_1}(\bar{a},\bar{p},\bar{I}(t))S(t) - \beta P_{I_1\to I_{12}}(\bar{a},\bar{p},\bar{I}(t))I_1(t) + \lambda I_{12}(t) - \lambda I_1(t) \\
\dot{I}_2(t) &= \beta P_{S\to I_2}(\bar{a},\bar{p},\bar{I}(t))S(t) - \beta P_{I_2\to I_{12}}(\bar{a},\bar{p},\bar{I}(t))I_2(t) + \lambda I_{12}(t) - \lambda I_2(t) \\
\dot{I}_{12}(t) &= \beta P_{I_1\to I_{12}}(\bar{a},\bar{p},\bar{I}(t))I_1(t) + \beta P_{I_2\to I_{12}}(\bar{a},\bar{p},\bar{I}(t))I_2(t) - 2\lambda I_{12}(t) \\
\text{with} \quad & S(0) = A, I_1(0) = B, I_2(0) = C, I_{12}(0) = D
\end{aligned} \tag{6.3}$$

The initial conditions of the initial value problem (6.3) are determined by the following simple assumptions: All consumers are assumed to have an equal and independent probability to get either product. The firms cannot observe by assumption whether a person already owns a product from the other firm. Moreover, a single consumer is content by assumption with a single product from a particular firm. Let q_1 and q_2 denote the amounts of free samples that firms 1 and 2 give out. Then the probability of getting firm 1's product is q_1/N, and the probability of getting firm 2's product is q_2/N. Since the probabilities are independent, the initial conditions A, B, C, D are given by equations (6.4):

$$A = N - B - C - D,\ B = N\frac{q_1}{N}(1 - \frac{q_2}{N}),\ C = N(1 - \frac{q_1}{N})\frac{q_2}{N},\ D = N\frac{q_1}{N}\frac{q_2}{N} \tag{6.4}$$

The terms in the differential equation system of the initial value problem (6.3) again have the natural interpretations. At time t consumers arrive at buying decisions at intensity β. Due to purchases, consumers move from compartment S to compartment I_1 at rate $\beta P_{S\to I_1}(\bar{a},\bar{p},\bar{I}(t))S(t)$ and to compartment I_2 at rate $\beta P_{S\to I_2}(\bar{a},\bar{p},\bar{I}(t))S(t)$. Again due to purchases, the consumers move to compartment I_{12} at rate $\beta P_{I_1\to I_{12}}(\bar{a},\bar{p},\bar{I}(t))I_1(t)$ from compartment I_1 and at rate $\beta P_{I_2\to I_{12}}(\bar{a},\bar{p},\bar{I}(t))I_2(t)$ from compartment I_2. At the same time the products are breaking down at intensity λ. Then due to product breakdowns, consumers move from compartments I_1 and I_2 to compartment S at rates $\lambda I_1(t)$ and $\lambda I_2(t)$ respectively. Additionally, since the consumers in compartment I_{12} own both products, they move from this compartment at rate $2\lambda I_{12}(t)$ to compartments I_1 and I_2.

As with the Stackelberg duopoly model, the formulation of initial value problem (6.3) implies that both firms' products have the same expected lifetime as the breakdown intensity λ is the same for both firms. I again interpret the parameter λ as reflecting the "state of technology" in production, with both firms choosing the best available technology to provide the most durable product. As the same technology is available to both firms, they both make equally durable products leading to the same breakdown intensity λ.

Finally, the population size stays constant over time as $d/dt(S(t) + I_1(t) + I_2(t) + I_{12}(t)) = \dot{S}(t) + \dot{I}_1(t) + \dot{I}_2(t) + \dot{I}_{12}(t) = 0$. This leads to identity (6.5):

$$S(t) + I_1(t) + I_2(t) + I_{12}(t) = N \tag{6.5}$$

The identity (6.5) may be used to eliminate one of the differential equations in the initial value problem (6.3) as we may, for example, write $S(t) = N - I_1(t) - I_2(t) - I_{12}(t)$. This assumption is again used in the numerical solution of initial value problem (6.3).

6.3 Firm behavior

Firm behavior in the model is ultimately determined by the SPNE of the interaction game. However, before describing the game I formulate the duopolist's problem. While the duopolist's problem does not explicitly enter into the game, I hope that it will illustrate the incentives behind the behavior of a single firm and highlight the similarities between this model and those presented in the preceeding chapters. Moreover, describing the duopolist's problem tells us how to calculate the duopolist's profits, which are then used as payoffs in the interaction game.

6.3.1 Duopolist's Problem

The duopolist's problem is constructed in the same way as in the preceeding chapters, i.e. the duopolist's objective function, the decision variables and constraints are explicit defined and combined. As the model's firms are symmetrical, let us consider firm $i \in \{1, 2\}$ as our duopolist and firm $j \in \{1, 2\}, j \neq i$ as its competitor. Firm i naturally seeks to maximize its profits, giving us the problem's objective function, by choosing the quality a_i and the price p_i of its product and whether to give out free samples q_i, giving us the decision variables of the problem.

Firm i's profits are determined by the demand of its product. The term $\beta P_{S \to I_i}(\bar{a}, \bar{p}, \bar{I}(t))S(t)$ describes the demand for firm i's product at time t from those consumers who own neither product. The term $\beta P_{I_j \to I_{12}}(\bar{a}, \bar{p}, \bar{I}(t))I_j(t)$ describes the demand for firm i's product at time t from those consumers who own firm j's product. Firm i's total sales at time t are then $\beta P_{S \to I_i}(\bar{a}, \bar{p}, \bar{I}(t))S(t) + \beta P_{I_j \to I_{12}}(\bar{a}, \bar{p}, \bar{I}(t))I_j(t)$. The per-unit discounted profit at time t are given by the term $\exp(-rt)(p_i - C(a_i))$. We may then integrate the product $\exp(-rt)(p_i - C(a_i))(\beta P_{S \to I_i} S(t) + \beta P_{I_j \to I_{12}} I_j(t))$ over the planning horizon T and account for the free sample costs $C(a_i)q_i$ to calculate firm i's profit. Firm i's profits $\pi(a_i, q_i, p_i \mid a_j, q_j, p_j)$ are given by equation (6.6):

$$\pi_i(a_i, q_i, p_i \mid a_j, q_j, p_j) = \int_0^T \exp(-rt)(p_i - C(a_i))(\beta P_{S \to I_i}(\bar{a}, \bar{p}, \bar{I}(t))S(t) + \beta P_{I_j \to I_{12}}(\bar{a}, \bar{p}, \bar{I}(t))I_j(t))dt - C(a_i)q_i \tag{6.6}$$

The constraints of firm i's problem are again simple. Firm i cannot choose a negative quality or a price nor give out a negative amount of free samples, giving us the natural constraints $a_i, q_i, p_i \geq 0$ for the decision variables. Firm i naturally operates within the described market structure, and we therefore require that the functions $S(t)$, $I_1(t)$, $I_2(t)$ and $I_{12}(t)$ are solutions of the initial value problem (6.3).

Firm i's problem then, given the choices (a_j, q_j, p_j) of firm j, takes the form (6.7):

$$\begin{aligned} \max_{a_i, q_i, p_i} \quad & \pi_i(a_i, q_i, p_i \mid a_j, q_j, p_j) \quad \text{s.t.} \\ & S(t), I_1(t), I_2(t), I_{12}(t) \text{ solutions of initial value problem (6.3) on } [0, T] \\ & a_i, q_i, p_i \geq 0 \end{aligned} \tag{6.7}$$

The duopolist's problem (6.7) illustrates what firm i seeks to do given the choices of firm j. We may note the similarity between problem (6.7) and the problems, especially the Follower's problem (5.10), presented in the preceeding chapters. The similarity of the problems reflects the similarity of incentives for firms across the models. Since the firms make their choices simultaneously, solving the duopolist's problem is impossible as we would need to know the choices of firm j to find the optimal choices for firm i. We circumvent this problem by modeling the interaction between the firms as a game.

6.3.2 Proposed interaction game

To model the interaction between the firms I propose a multi-stage (multi-stage in the sense of Fudenberg and Tirole (1991, p. 70-2)) game with three stages: First, the firms choose their qualities simultaneously and observe each other's choices. Next, the firms simultaneously choose how many free samples they give out and observe each other's choices. Finally, the firms simultaneously choose prices, enter the market simultaneously and receive the profits (6.6) dependent on the choices of both firms. Figure 6.2 illustrates the game structure.

I offer the following rationalization for choosing this game specification: The firms choose their quality first as it is the most difficult variable to change, as, for example, the firms may have built quality-specific manufacturing machines. The firms choose their free samples next. If we interpret choosing quality as building the appropriate machines, it is natural that the firm may not give out any free products before setting up the appropriate infrastructure. Finally, the firms choose their prices last as the prices are clearly the easiest variables to change (firms may change their prices simply by announcing a price change).

I assume that the choice sets and the way to calculate profits are common knowledge to both firms. Moreover, I assume that when faced with situation that can be represented as a normal-form game the both firms are equally capable at calculating NE strategies and deducing that the other firm will be able to do the same. The play in each normal-form game should then converge to a NE by assumption.

More precisely speaking, the proposed interaction game is an extensive-form game. An extensive-form game consists of a set of players, the order of moves (i.e. who moves when), the players' payoffs as a function of the moves, what the players' choices are when they move, what each player knows when he makes his choices and the probability distributions over any exogenous events (Fudenberg and Tirole (1991, p. 77)). For the proposed game, the set of players consists of the two firms, the players' payoffs are determined by the profits (6.6). As there are no exogenous events, there is no need to define any probability distributions for them. A structure called the game tree provides the rest of the required components. The game tree describes the order in which the players make their choices. Unfortunately, the game tree for the proposed game is too large to be presented in its entirety. Figure 6.3 illustrates a pruned game tree of the proposed game. In the complete game tree of the proposed game each of the pruned branch nodes expands to a whole branch that has the same structure as the presented branch at that node level.

Each node of the game tree is a decision point for a player. The game starts at the highest node, denoted by an empty circle around the node, and moves down the game tree in nodes denoted by filled circles. The nodes of figure 6.3 are named either 1 or 2 to indicate whether firm 1 or 2 is making its choices at that particular node. Each branch stemming from a node describes an action that the player may choose. The action sets are naturally the choice sets (3.8) at the appropriate nodes, and the choice order naturally conforms to figure 6.2.

While the game tree in figure 6.3 specifically describes the order in which a specific player makes a choice, it may still be used to model simultaneous choice. Simultaneous choice in the game tree is modeled by using information sets. An information set describes what a player knows when it is choosing its action. The nontrivial information sets in the game tree in figure 6.3 are marked with dashed lines and the trivial information sets with only a single member are naturally left unmarked. When the game arrives to a node which is a member of a nontrivial information set, the player who is choosing its action at that node is unaware in which of the information set's member nodes the game actually is.

To emphasize that the use of information sets indeed allows us to model simultaneous choice, I offer the following informal example: Let us consider Rock-Paper-Scissors as our game. The traditional way of playing Rock-Paper-Scissors is the players making their choices of rock, paper or scissor at the same time. This approach can naturally be modeled with a normal-form game in which rock, paper and scissor form the row and column choices of the players. The same game may also be played in a way where the choices are not made at the same time. Suppose that the first player writes his choice on a paper that is sealed in an envelope. The second player then writes down his choice, opens the envelope and the game concludes according the rules of Rock-Paper-Scissors and the choices of both players. This second way can be modeled with an extensive-form game with a single information set. When the second player receives the sealed envelope, the extensive-form game has arrived to some unknown node in the information set as the second player knows that the first player has made a choice but does not know what that choice is. These two approaches are clearly equivalent as they model the same game.

With the Rock-Paper-Scissors example in mind, we may interpret the first two node levels of the game tree as the first stage, the third and the fourth node levels as the second stage and the fifth and sixth node levels as the third stage of the game structure in figure 6.2. For future reference, let us name the nodes in the first two node levels as the "first stage nodes", nodes in the third and the fourth node levels as the "second stage nodes" and finally the nodes in the fifth and sixth node levels as the "third stage nodes" of firms 1 and 2. Each of the "stages" can be likened to a normal-form game as in the Rock-Paper-Scissors example, and this observation will be used to construct the SPNE for the whole game later on.

The play in an extensive-form game is determined by the strategies of both players. The strategy of a player is a set of instructions that describes how the player should choose his actions at each decision node. These instructions are written before the play starts, and the players cannot change their strategies midplay. The instructions are written on basis of the past actions and the information of the player at the decision node. The instructions may instruct the player to randomize his choices over the actions, as there is no guarantee that the decision node has a "self-evidently best" action. For example, when playing Rock-Paper-Scissors if the second player knows the first player's choice, he may always choose the winning action. Without this information the second player needs to randomize his choices (for example by putting equal probability weights to each of his actions). Finally, the instructions cover all possible contingencies, i.e. a strategy will provide instructions for all decision nodes of the player.

We may now give a more formal definition of a strategy for the proposed game. The strategies described here can be likened to behavior strategies (see e.g. Fudenberg and Tirole (1991, p. 83-5)). First, let us denote the number of possible qualities as $\#A$, i.e. the set of quality choices A has $\#A$ different members, the number of possible free samples as $\#Q$ and the number of possible prices as $\#P$. For the sets (3.8), we have $\#A = \#Q = \#P = 4$. The strategy of firm i is a structure (l_i, n_i, h_i) such that $l_i \in \mathbb{R}^{\#A}$, $n_i : A \times A \to \mathbb{R}^{\#Q}$ and $h_i : A \times A \times Q \times Q \to \mathbb{R}^{\#P}$. Vector l_i is a probability distribution over the actions in its first stage node, i.e. the firm choose action j with probability $(l_i)_j$ in its first stage node. The functions $n_i(a_1, a_2)$ and $h_i(a_1, a_2, q_1, q_2)$ are rules that firm i uses in choosing its strategies given the game's history. The vector $x = n_i(a_1, a_2) \in \mathbb{R}^{\#Q}$ is a probability distribution over the actions of firm i in its second stage node given that firm 1 has chosen quality a_1 and firm 2 has chosen quality a_2, i.e. the history (a_1, a_2), in their first stage nodes. Correspondingly, the vector $y = h_i(a_1, a_2, q_1, q_2) \in \mathbb{R}^{\#P}$ is a probability distribution over the actions of firm i in its third stage node given the history (a_1, a_2, q_1, q_2). As they are probability distributions, each component of vectors l_i, x and y is non-negative and the sum of the components of l_i, x and y is 1.

After giving the definition of a strategy (l_i, n_i, h_i), I think it appropriate to also give an interpretation. The strategy may be thought of as a book that is is written before the start of the game and that gives the firm instructions on how to choose an action at every possible decision node. The instructions describe how a firm should weight a dice such that the result of the dice roll determines the firm's action. As there are three parts in the strategy (l_i, n_i, h_i), we may consider the book to have three chapters. The first chapter, which may be thought to consists of the information in l_i, describes how the firm should weight its dice in the first stage. The second chapter, consisting of the information in n_i, describes how the firm should weight its dice in the second stage for all possible choice combinations from the first stage. Finally the third chapter, consisting of the information in h_i, describes how the firm should weight its dice in the third stage for all possible choice combinations from the first and second stages.

Why choose such a complicated definition of a strategy instead of, for example, assuming that firms choose their actions deterministically? Allowing the firms to randomize over their actions allows us the invoke Nash's theorem (see e.g. Fudenberg and Tirole (1991, p. 29)) which guarantees the existence of a NE in the stage normal-form games as each of the normal-form games has a finite number of players and actions. When each of the possible normal-form games has a NE, we may construct the SPNE for the whole proposed game using the found NEa.

6.3.3 Subgame-perfect Nash equilibrium

As the proposed game is an extensive-form game, I naturally use the SPNE as my equilibrium concept. The SPNE is, as the name suggests, a refinement of the classical NE concept. For sake of completeness, let us retrace the steps through which we arrive at the SPNE from the classical NE concept for the proposed game.

The classical NE for an arbitrary two player game may be defined as follows. Let $s_i, i = 1,2$ denote player i's strategy, and $\pi_i(s_1, s_2)$ player i's (expected) payoff for the strategy pair (s_1, s_2). Strategy pair (s_1^*, s_2^*) is the game's NE if the following conditions (6.8) hold:

$$\begin{aligned} \pi_1(s_1^*, s_2^*) &\geq \pi_1(s_1', s_2^*) \text{ for all } s_1' \\ \pi_2(s_1^*, s_2^*) &\geq \pi_2(s_1^*, s_2') \text{ for all } s_2' \end{aligned} \tag{6.8}$$

Conditions (6.8) state that neither player can improve his payoff by unilaterally changing its strategy. Since we assume that the players are rational, we may reasonably expect the players to have optimized their own strategies to the point where there are no unilateral payoff improvements. The play with rational players will therefore converge to a NE. In order to be able to use the conditions (6.8) for the proposed game, we need to determine the payoffs for the firms.

The construction of the payoffs for the proposed game requires some care. As we have allowed the firms to randomize over their choices, we cannot use the profits (6.6) as the payoffs as such. Rather, the payoffs for the game are the expected profits from a chosen strategy pair. To define the expected profits, it is necessary to present compact notation for the possible choice combinations and the probability for reaching a particular choice combination.

First let us denote a particular choice combination as $c = (a_1, a_2, q_1, q_2, p_1, p_2)$ where naturally $a_i \in A, q_i \in Q, p_i \in P, i = 1,2$. Next let set C denote the set of all choice combinations, i.e. $C = \{(a_1, a_2, q_1, q_2, p_1, p_2) \mid a_i \in A, q_i \in Q, p_i \in P, i = 1,2\}$. Now given any choice combination c, we may write the profits (6.6) of both firms 1 and 2 simply as $\pi_i(c) = \pi_i(a_i, q_i, p_i \mid a_j, q_j, p_j)$.

The probability of reaching a particular choice combination is determined by the strategies of both firms. Let $s_i = (l_i, n_i, h_i)$ denote firm i's strategy and let $c = (a_1, a_2, q_1, q_2, p_1, p_2)$ denote some choice combination. Let us now suppose that the choice a_1 is the kth and the choice a_2 the jth choice in the set A. Now the probability of firm 1 choosing a_1 in the first stage is naturally given by $(l_1)_k$ (the kth component of vector l_1), and correspondingly the probability of firm 2 choosing a_2 in the first stage is given by $(l_2)_j$. Naturally then the probability of firm 1 choosing a_1 and firm 2 choosing a_2 at the first stage is the product $(l_1)_k(l_2)_j$. Next, let us suppose that the choices q_1 is the k'th and the choice q_2 the j'th choice in set Q. Now given the first stage choices (a_1, a_2), firm 1 chooses q_1 with the probability $(n_1(a_1, a_2))_{k'}$ and firm 2 chooses q_2 with the probability $(n_2(a_1, a_2))_{j'}$. Naturally then the probability of firm 1 choosing q_1 and firm 2 choosing q_2 together at the second stage is the product $(n_1(a_1, a_2))_{k'}(n_2(a_1, a_2))_{j'}$. After applying this reasoning to determine the probabilities for the third stage choices (where p_1 is chosen with the probability given by the k''th component of $h_1(a_1, a_2, q_1, q_2)$ etc.), we may write the probability $P(c)$ for the choice combination c in equation (6.9):

$$P(c) = (l_1)_k(l_2)_j(n_1(a_1, a_2))_{k'}(n_2(a_1, a_2))_{j'}(h_1(a_1, a_2, q_1, q_2))_{k''}(h_2(a_1, a_2, q_1, q_2))_{j''} \tag{6.9}$$

We may now determine the payoff functions for both firms. Let us abuse the notation even more and define functions $\pi_i(s_1, s_2), i = 1,2$ as our payoff functions given strategies s_1 and s_2. The payoff of firm i is the expected value of firm i's profits given the strategies (s_1, s_2). Accounting for all possible choice combinations (i.e. summing through the set of all possible combinations C), the payoff functions $\pi_i(s_1, s_2)$ are then written as in equation (6.10):

$$\pi_i(s_1, s_2) = \sum_{c \in C} P(c)\pi_i(c) \tag{6.10}$$

While using the payoffs (6.10) together with the strategies $s_i = (l_i, n_i, h_i)$ allows us to write the NE conditions (6.8) for the proposed game, solving the NE strategies (s_1^*, s_2^*) is difficult due to the complicated definition of a strategy. There is no requirement for the NE to be unique as there may exists many pairs $(s_1^*, s_2^*), (s_1^*, s_2^*)', \ldots$ that satisfy the conditions (6.8). As the proposed game is an extensive-form game, the conditions (6.8) also allow for equilibria that are in a sense unbelievable. Since the firms commit to their strategies before the game starts, it is possible for the firms to "threaten" each other by committing to make detrimental choices. For example, firm 1 may construct its strategy s_1 such that it always chooses price $p_1 = 0$ in the third stage if it observes firm 2 choosing a high quality (i.e. $a_2 = 2$ or $a_2 = 3$) in the first stage to undercut firm 2's profits. These types of threats are unbelievable since when the game has reached the third stage, it is never optimal for firm 1 to choose price $p_1 = 0$ as it would always lead to a negative profit. The example threat is thus an empty one. Nevertheless, empty threats may form a part of firm 1's strategy to influence the choices of firm 2. The use of the SPNE concept allows us to avoid these types of equilibria.

Defining the SPNE concept requires first defining what is meant by a proper subgame. Roughly speaking (an exact definition may be found in Fudenberg and Tirole (1991, p. 94)) a proper subgame is an extensive-form game that forms a part of the whole extensive-form game and shares the same information sets. We may construct proper subgames from figure 6.3 by taking a single node and all its successor nodes. If the first node is a member of an information set, we also have to include all the nodes of the information set and all their successor nodes.

Fudenberg and Tirole (1991, p. 95) defines the SPNE as follows: A behavior-strategy profile σ of an extensive-form game is a SPNE if the restriction of σ to G is a NE of G for every proper subgame G. Let us interpret this definition in terms of the proposed game. I have likened the strategies $s_i = (l_i, n_i, h_i)$ to behavioral strategies. I therefore further liken the behavior-strategy profile σ to a pair of strategies (s_1, s_2). A restriction of σ to a proper subgame is then the parts of σ that describe play in the parts of the whole extensive-form game that form the proper subgame. For example, suppose that we have some strategy pair (s_1, s_2) and wish to see what form one of its restriction takes. Suppose further that firm 1 has chosen quality $a_1 = 1$ and firm 2 has chosen quality $a_2 = 1$. Then a restriction of the strategy pair (s_1, s_2) to the subgame takes the form $(s_i \mid a_1 = 1, a_2 = 1) = (n_i(1,1), f)$ where $f(x, y) = h(1, 1, x, y), x, y \in Q$. The form of the restriction naturally depends on which node the proper subgame is created.

Why does the subgame-perfection requirement rule out empty threats in the equilibrium strategies in the proposed game? This property comes from the requirement that the restrictions of a SPNE strategies form NEa in all of its subgames. Suppose that the firms have already chosen their qualities and free samples and the game has entered the third stage. In a SPNE, the restricted strategies in the third stage will then have to form a NE (i.e. satisfy conditions (6.8)) for this subgame. This precludes choosing a suboptimal price and thereby fulfilling a possible threat. We may use the same argument to preclude threats of high amounts of free samples in the second stage. Moreover, as each extensive-form game is a proper subgame of itself, a SPNE is necessarily also a NE (Fudenberg and Tirole (1991, p. 96)).

After presenting the definition and the properties of the SPNE, let us consider what conditions should the SPNE strategies fulfill. It is now useful to recall that each stage of the proposed game may be seen as a normal-form game. Starting from the last stage, we may note that the third stage is equivalent to a normal-form game where the firms are simultaneously choosing their prices with the other profit-determining variables having already been set. Then the SPNE requirement states that the third stage restrictions of the strategies need to form the NEa for all possible third stage normal-form games. Next, as now we know that each third stage normal-form game has a NE that the play adheres to, we may associate the NE profits from the third stage normal-form games to each second stage choice pair. The situation the firms then face in the second stage is equivalent to a normal-form game, as the qualities have already been set and the prices will be set by the NE strategies of the third stage normal-form games. As with the third stage normal-form game, we may solve the NEa of these new second stage normal-form games and associate their NE profits to the first stage choices that lead to each particular second stage. Finally, this procedure then gives us a first stage normal-form game. We may then solve the NEa for the first stage normal-form game. The NE profits for the first stage normal-form game now correspond to the expected profits of the proposed game. Moreover, the NE strategies for the third, second and first stages may be combined to form the SPNE strategies for the whole proposed game.

Let us now try to state the preceeding reasoning in a more precise manner. We wish to consider firm i's strategy $s_i = (l_i, n_i, h_i)$ for the proposed game, and construct the conditions for the parts l_i, n_i and h_i such that the strategy s_i satisfies the SPNE requirements. These conditions are next constructed stage-by-stage.

Let us first consider the third stage conditions. Suppose that the proposed game has entered the third stage with history (a_1, a_2, q_1, q_2). Let $h_i = h_i(a_1, a_2, q_1, q_2)$ denote firm i's strategy, and $\pi_i^{\text{Third Stage}}(h_1, h_2)$ firm i's (expected) profits for strategies (h_1, h_2) at the third stage normal-form game. Each of the third stage normal-form games then has the form of the normal-form game in figure 6.4. Then using conditions (6.8), we may write conditions (6.11) to guarantee that the strategies in the third stage form a NE:

$$\begin{aligned}
&\text{For all } a_i \in A, q_i \in Q, i = 1, 2 \\
&\qquad \pi_1^{\text{Third Stage}}(h_1^*, h_2^*) \geq \pi_1^{\text{Third Stage}}(h_1', h_2^*) \text{ for all } h_1' \\
&\qquad \pi_2^{\text{Third Stage}}(h_1^*, h_2^*) \geq \pi_2^{\text{Third Stage}}(h_1^*, h_2') \text{ for all } h_2'
\end{aligned} \tag{6.11}$$

When the NE conditions (6.11) for the third stage normal-form game hold, we note that with the second stage choices (q_1, q_2) with history (a_1, a_2), firm i will have the expected profits $\pi_i^{\text{Third Stage}}(h_1^*, h_2^*)$. We therefore associate the third stage expected NE profits as the (induced) expected profits for choices (q_1, q_2). Let $\pi_i^{\text{Second Stage}}(n_1, n_2)$, where $n_i = n_i(a_1, a_2)$, denote firm i's induced profits in the second stage normal-form game. Each of the second stage normal-form games now has the form of the normal-form game in figure 6.6. Again using conditions (6.8), we may write the conditions (6.12) to guarantee that the strategies in the second stage form a NE:

$$\begin{aligned}
&\text{For all } a_i \in A, i = 1, 2 \\
&\qquad \pi_1^{\text{Second Stage}}(n_1^*, n_2^*) \geq \pi_1^{\text{Second Stage}}(n_1', n_2^*) \text{ for all } n_1' \\
&\qquad \pi_2^{\text{Second Stage}}(n_1^*, n_2^*) \geq \pi_2^{\text{Second Stage}}(n_1^*, n_2') \text{ for all } n_2'
\end{aligned} \tag{6.12}$$

When the NE conditions (6.12) and (6.11) hold, firm i will have the expected profits $\pi_i^{\text{Second Stage}}(n_1^*, n_2^*), i = 1, 2$ for first stage choices (a_1, a_2). We therefore again associate the second stage (induced) expected NE profits as the (induced) expected profits for the choices (a_1, a_2). Let $\pi_i^{\text{First Stage}}(l_1, l_2)$ denote firm i's induced profits in the first stage normal-form game with strategies (l_1, l_2). The first stage normal-form game then has the form of the normal-form game in figure 6.8. Once again using conditions (6.8), we may write the conditions (6.13) to guarantee that the strategies in the first stage form a NE:

$$\pi_1^{\text{First Stage}}(l_1^*, l_2^*) \geq \pi_1^{\text{First Stage}}(l_1', l_2^*) \text{ for all } l_1'$$
$$\pi_2^{\text{First Stage}}(l_1^*, l_2^*) \geq \pi_2^{\text{First Stage}}(l_1^*, l_2') \text{ for all } l_2' \tag{6.13}$$

When the strategy pair (s_1, s_2) satisfies conditions (6.11), (6.12) and (6.13), it will form a SPNE for the proposed game. The construction of the presented conditions also provides us now with the method for actually solving the SPNE strategies.

6.3.4 Solving the SPNE

As the initial value problem (6.3) is (almost) equivalent to the post-entry initial value problem (5.6), it too has no easy analytical solutions. I therefore use numerical methods as solving the initial value problem is necessary to determine the profits, and thereby the behavior, of the two firms.

The approach for solving the firm behavior is a process very much similar to backward induction in games of perfect information. Whereas in the standard applications of backward induction we only need to consider the actions of a single player at a time, in this instance we consider the behavior of two firms in a single normal-form game at a time. We study the proposed game stage by stage starting from the third stage normal-form games. We first solve a NE for each of the third stage normal-form games, use the NE strategies to construct the functions $h_i(a_1, a_2, q_1, q_2), i = 1, 2$ and the NE profits as the induced profits for the second stage normal-form games. We then solve a NE for each second stage normal-form games with induced profits, use the NE strategies to construct the functions $n_i(a_1, a_2), i = 1, 2$ and the NE profits as the induced profits for the first stage normal-form game. Finally, we solve a NE for the first stage normal-form game and use the NE strategies as the strategies $l_i, i = 1, 2$ and the NE profits as the expected profits of the whole proposed game. The strategies $s_i = (l_i, n_i, h_i), i = 1, 2$ then form a SPNE for the whole proposed game and determine the range of possible firm behavior we may expect to observe.

NEa in the normal-form games for each stage are solved with the support enumeration algorithm (algorithm 1 of Avis et al. (2010, p. 13)) which finds all NEa of a given nondegenerate bimatrix normal-form game. The algorithm iterates through all combinations of possible supports of strategies (a support of a strategy is the set of actions to which the strategy assigns a positive probability) and tries to solve a system of linear equations determined by the supports. If the solution of the system of linear equations satisfies the best response conditions and is non-negative, the solution forms a NE for the game.

For each subgame, I arbitrarily use the first NE that is found. This choice bypasses the questions of equilibrium choice that could be raised, as considering them would lead to new complications in an already overcomplicated model. I acknowledge the possibility that multiple NEa in the second or third stage games may now hide possible NEa in earlier stages. For example, let us suppose that there are multiple NEa in a third stage normal-form game. Since the approach chooses the first found NE and uses it to calculate the induced profits for all preceeding stages leading to this third stage normal-form game, the choice affects all subsequent normal-form games through profits. This choice may then mask other possible NEa in earlier stages.

For sake of clarity, let us walk through the solving approach step by step. We start by considering all possible third stage normal-form games. Suppose that the game has entered the third stage corresponding to the fifth and the sixth decision nodes in figure 6.3 with history (a_1, a_2, q_1, q_2). The third stage with the given history is then equivalent to the normal-form game in figure 6.4. We now solve the NE strategies and (expected) profits of the equivalent normal-form game. Let us denote firm i's ($i = 1, 2$) NE strategy by h_i^* and firm i's expected profits as $\pi_i^{\text{Third Stage}}(h_1^*, h_2^*)$. Given the history (a_1, a_2, q_1, q_2) we then simply set $h_i(a_1, a_2, q_1, q_2) = h_i^*$. Going through all possible (a_1, a_2, q_1, q_2)-combinations then completes the construction of functions $h_i(a_1, a_2, q_1, q_2)$.

Next, solving the NEa of all possible third stage normal-form games we may eliminate the third stage nodes in figure 6.3 and study instead a truncated game tree in figure 6.5. In this new game tree the second stage nodes lead to the previously calculated third stage NE profits $\pi_i^{\text{Third Stage}}(h_1^*, h_2^*)$. We now apply the same procedure to the second stage nodes as to the third stage nodes. Given the history (a_1, a_2) the second stage is equivalent to a normal-form game given in figure 6.6. We again solve the NE strategies and the expected NE profits of the equivalent normal-form game. Let now n_i^* denote firm i's ($i = 1, 2$) NE strategy and $\pi_i^{\text{Second Stage}}(n_1^*, n_2^*)$ firm i's expected NE profits. We may then again set $n_i(a_1, a_2) = n_i^*$ to construct the function $n_i(a_1, a_2)$. Going through all possible (a_1, a_2)-combinations then completes the construction of functions $n_i(a_1, a_2)$.

Finally, solving the NEa of all possible second stage normal-form games we may eliminate the second stage nodes in figure 6.5 and study instead an even more truncated game tree in figure 6.7. In this new game tree the first stage nodes now lead to the second stage NE profits $\pi_i^{\text{Second Stage}}(n_1^*, n_2^*)$. Again, the new game tree is equivalent to the normal-form game in figure 6.8. We again solve the NE strategies and the expected NE profits of the equivalent normal-form game. Let l_i^* denote firm i's ($i = 1, 2$) NE strategy and $\pi_i^{\text{First Stage}}(l_1^*, l_2^*)$ firm i's expected NE profits. These components complete the construction of the SPNE strategies for the proposed game, as we may now set $l_i = l_i^*$ completing the strategy $s_i = (l_i, n_i, h_i)$. Moreover, the expected NE profits $\pi_i^{\text{First Stage}}(l_1^*, l_2^*)$ for the (first stage) normal-form game give us the expected NE profits of the whole proposed game.

Figure 6.9 describes the structure of the implemented script. The structure of the script is a close reflection to the way the SPNE strategies are constructed. The script first generates numerically the payoff matrices for all of the third stage normal-form games and solves the NE strategies and profits for the normal-form games. Next, the script generates the payoff matrices of the second stage normal-form games from the NE profits of the third stage normal-form games and solves the NE strategies and profits for the newly-constructed second stage normal-form games. Finally, the script generates the payoff matrices of the first stage normal-form game from the NE profits of the second stage normal-form games and solves the NE strategies and profits of the final first stage normal-form game. From the solved NE strategies from the third, second and first stage normal-form games the script generates SPNE strategies for the proposed game. See appendix E for the implemented script.

The subroutines `FirstStageNEQualities`, `SecondStageNESamples` and `ThirdStageNEPrices-Parallel` in figure 6.9 construct the SPNE strategy components from the first, second and third stage normal-form games respectively. For each of the normal-form games they create the appropriate payoff matrices, solve and store the NE strategies. The payoff matrices for third stage normal-form games are generated by calculating profits (6.6), and the payoff matrices for the second and first stages from the NE profits from following stages. Solving the NE strategies for a given normal-form game is done with the subroutine `NashEquilibriaBySupportEnume-ration`.

The subroutine `NashEquilibriaBySupportEnumeration` is an implementation of the support enumeration algorithm (Avis et al. (2010, p. 13)). The algorithm requires an input of a nondegenerate bimatrix game, which are provided by the subroutines `FirstStageNEQualities`, `SecondStageNESamples` and `ThirdStage-NEPricesParallel` and gives as an output all NEa of the given game. This implementation requires that the payoff matrices are $n \times n$-square matrices. This requirement stems from the implementation itself while the algorithm could handle $n \times m$-matrices.

The payoff matrices for the third stage normal-form games are calculated parallely in order to save computing time. The subroutines `parallelWrapperProfitsFirm1` and `parallelWrapperProfitsFirm2` are "wrappers" for implementing parallel computing with MATLAB's *parfor*-command. The subroutines `profitsFirm1` and `profitsFirm2` calculate the actual profits of firm 1 and 2 given choices $(a_1, a_2, q_1, q_2, p_1, p_2)$. The profits are again calculated using the same methods as in the previous two models: The subroutines first solve the initial value problem (6.3) using *ode45*. They then generate the appropriate revenue vectors by using subroutines `revenueFirm1` and `revenueFirm2` and use *trapz* to calculate the integral in equation (6.6) numerically. The profits for the firms are then determined by accounting for the free samples costs from the calculated integrals. The subroutines `revenueFirm1` and `revenueFirm2` generate the revenue vectors required for numerical integration. The subroutine `duopolyDES` is the (reduced) differential equation system of initial value problem (6.3) required by *ode45*.

6.4 Outcomes

We wish to determine how the firms will behave in order to consider the research questions. The firm behavior in the model is determined by the SPNE strategies. However, the nature of the SPNE strategies creates difficulties in presenting them in their entirety. First, let us recall that the functions $n_i(a_1, a_2)$ and $h_i(a_1, a_2, q_1, q_2)$ for firm i's strategy s_i are constructed by associating a probability distribution to each member of sets $A\times A$ and $A\times A\times Q\times Q$. If we were then to present the probability distribution associated to a particular member on a single line, we would need in total $\#A \cdot \#A$ lines to describe the function $n_i(a_1, a_2)$ and $\#A \cdot \#A \cdot \#Q \cdot \#Q$ lines to describe the function $h_i(a_1, a_2, q_1, q_2)$ for firm i's strategy, where $\#A$ denotes the number of members in set A and $\#Q$ the number of members in set Q. For the sets (3.8), we have $\#A \cdot \#A = 16$ and $\#A \cdot \#A \cdot \#Q \cdot \#Q = 256$. Since we are interested in studying firm behavior in fifteen different combinations of trend type and strength, planning horizon length and firm patience, completely describing the SPNE strategies of even a single firm for all considered cases would require prohibitively many lines. I therefore choose to present here only the parts of both firms' SPNE strategies that directly affect firm behavior. I acknowledge that the complete strategies for the proposed game would be required to verify that the strategies indeed form a SPNE. I invite the reader interested in checking the subgame-perfection claim to run the supplied script to generate the appropriate information. See appendix B for an example pair of complete SPNE strategies.

6.4.1 SPNE strategies

First stage SPNE strategies l_i

Tables 6.1-6.3 present the first stage normal-form game strategies of the SPNE that determine the quality choices. Since each vector in the tables is a probability distribution over the quality choice set A of sets (3.8), we may describe how the quality choices are made by studying the components of the vectors. First component of vector l_i gives the probability that firm i chooses quality $a_i = 0$, the second the probability that firm i chooses quality $a_i = 1$ etc.

Table 6.1 describes how the quality choices are made in the SPNE for patient firms with discount rate $r = 0$. We note that in most cases the quality decisions are made deterministically, as in most cases the vectors take the form $(0, 0, 1, 0)^T$ or $(0, 0, 0, 1)^T$. These vectors correspond to firm i choosing the quality $a_i = 2$ or $a_i = 3$ for certain. We also note that quality choices in cases with parabel trend with strength $d = 2$ and planning horizons $T = 10$ and $T = 100$ are nondeterministic. With planning horizon $T = 10$, we have $l_i = (0, 0.085328, 0.46174, 0.45293)^T$ which leads to firm i choosing quality $a_i = 1$, $a_i = 2$ or $a_i = 3$. With planning horizon $T = 100$, we have $l_i = (0, 0.28001, 0, 0.71999)^T$ which leads to firm i choosing either quality $a_i = 1$ or $a_i = 3$.

Table 6.2 describes how the quality choices are made in the SPNE for normal firms with discount rate $r = 0.05$. Again we note that in most cases the quality choices are made deterministically, as in most cases the vectors take the form $(0, 0, 1, 0)^T$ or $(0, 0, 0, 1)^T$. These vectors correspond to firm i choosing quality $a_i = 2$ or $a_i = 3$ for certain. We also note that quality choices in cases with parabel trend with strength $d = 2$ and planning horizons $T = 10$ and $T = 100$ are nondeterministic. With planning horizon $T = 10$, we have $l_i = (0, 0.0155, 0.58875, 0.39575)^T$ which leads to firm i choosing quality $a_i = 1$, $a_i = 2$ or $a_i = 3$. With planning horizon $T = 100$, we have $l_i = (0, 0.25943, 0.14396, 0.59661)^T$ which leads to firm i choosing quality $a_i = 1$, $a_i = 2$ or $a_i = 3$.

Planning horizon	Trend type	l_1	l_2
$T = 10$	no trend	$(0, 0, 1, 0)^T$	$(0, 0, 1, 0)^T$
	linear, $d = 2$	$(0, 0, 1, 0)^T$	$(0, 0, 1, 0)^T$
	parabel, $d = 2$	$(0, 0.085328, 0.46174, 0.45293)^T$	$(0, 0.085328, 0.46174, 0.45293)^T$
	linear, $d = 8$	$(0, 0, 0, 1)^T$	$(0, 0, 0, 1)^T$
	parabel, $d = 8$	$(0, 0, 1, 0)^T$	$(0, 0, 1, 0)^T$
$T = 100$	no trend	$(0, 0, 1, 0)^T$	$(0, 0, 1, 0)^T$
	linear, $d = 2$	$(0, 0, 1, 0)^T$	$(0, 0, 1, 0)^T$
	parabel, $d = 2$	$(0, 0.28001, 0, 0.71999)^T$	$(0, 0.28001, 0, 0.71999)^T$
	linear, $d = 8$	$(0, 0, 0, 1)^T$	$(0, 0, 0, 1)^T$
	parabel, $d = 8$	$(0, 0, 1, 0)^T$	$(0, 0, 1, 0)^T$

Table 6.1: First stage normal-form game SPNE strategies, patient firms, $r = 0$

Planning horizon	Trend type	l_1	l_2
$T = 10$	no trend	$(0, 0, 1, 0)^T$	$(0, 0, 1, 0)^T$
	linear, $d = 2$	$(0, 0, 1, 0)^T$	$(0, 0, 1, 0)^T$
	parabel, $d = 2$	$(0, 0.0155, 0.58875, 0.39575)^T$	$(0, 0.0155, 0.58875, 0.39575)^T$
	linear, $d = 8$	$(0, 0, 0, 1)^T$	$(0, 0, 0, 1)^T$
	parabel, $d = 8$	$(0, 0, 1, 0)^T$	$(0, 0, 1, 0)^T$
$T = 100$	no trend	$(0, 0, 1, 0)^T$	$(0, 0, 1, 0)^T$
	linear, $d = 2$	$(0, 1, 0, 0)^T$	$(0, 0, 1, 0)^T$
	parabel, $d = 2$	$(0, 0.25943, 0.14396, 0.59661)^T$	$(0, 0.25943, 0.14396, 0.59661)^T$
	linear, $d = 8$	$(0, 0, 0, 1)^T$	$(0, 0, 0, 1)^T$
	parabel, $d = 8$	$(0, 0, 1, 0)^T$	$(0, 0, 1, 0)^T$

Table 6.2: First stage normal-form game SPNE strategies, normal firms, $r = 0.05$

Planning horizon	Trend type	l_1	l_2
$T = 10$	no trend	$(0,0,1,0)^T$	$(0,0,1,0)^T$
	linear, $d = 2$	$(0,0,1,0)^T$	$(0,0,1,0)^T$
	parabel, $d = 2$	$(0,0,1,0)^T$	$(0,0,1,0)^T$
	linear, $d = 8$	$(0,0,1,0)^T$	$(0,0,1,0)^T$
	parabel, $d = 8$	$(0,0,1,0)^T$	$(0,1,0,0)^T$
$T = 100$	no trend	$(0,0,1,0)^T$	$(0,0,1,0)^T$
	linear, $d = 2$	$(0,0,1,0)^T$	$(0,0,1,0)^T$
	parabel, $d = 2$	$(0,0,1,0)^T$	$(0,0,1,0)^T$
	linear, $d = 8$	$(0,0,1,0)^T$	$(0,0,1,0)^T$
	parabel, $d = 8$	$(0,0,1,0)^T$	$(0,1,0,0)^T$

Table 6.3: First stage normal-form game SPNE strategies, impatient firms, $r = 10$

Table 6.3 describes how the quality choices are made in the SPNE for impatient firms with discount rate $r = 10$. We note that the quality choices are made deterministically in every case. With the vector $l_i = (0,0,1,0)^T$ firm i chooses quality $a_i = 2$ for certain, and with the vector $l_i = (0,1,0,0)^T$ firm i chooses quality $a_i = 1$ for certain.

Second stage SPNE strategies $n_i(a_1, a_2)$

The free samples choices in the second stage depend on the quality choices in the first stage that are determined by tables 6.1-6.3. Tables 6.4-6.6 give the second stage normal-form game strategies of the SPNE that determine the free samples choices. We describe how the free samples choices are made by determining the vectors $n_i(a_1, a_2)$ that are dependent on the quality choices (a_1, a_2). The vectors $n_i(a_1, a_2)$ are probability distributions over the free samples set Q of sets (3.8). Again, the first component of vector $n_i(a_1, a_2)$ is the probability that firm i chooses free samples $q_i = 0$, the second component the probability that firm i chooses free samples $q_i = 200\ 000$ etc.

Table 6.4 describes how the free samples choices are made in the SPNE for patient firms with discount rate $r = 0$. We again note that in most cases the free samples choices are deterministic: Firm i chooses free samples $q_i = 0$ when $n_i(a_1, a_2) = (1,0,0,0)^T$, free samples $q_i = 200\ 000$ when $n_i(a_1, a_2) = (0,1,0,0)^T$, free samples $q_i = 400\ 000$ when $n_i(a_1, a_2) = (0,0,1,0)^T$ and free samples $q_i = 600\ 000$ when $n_i(a_1, a_2) = (0,0,0,1)^T$ for certain. We also note that free samples choices in the case with linear trend with strength $d = 2$ and planning horizons $T = 10$ are nondeterministic. As firm i's free samples choice strategy is in this case $n_i(2,2) = (0.34668, 0.21864, 0.43468, 0)^T$, firm i chooses free samples $q_i = 0$, $q_i = 200\ 000$ or $q_i = 400\ 000$.

Table 6.5 describes how the free samples choices are made in the SPNE for normal firms with discount rate $r = 0.05$. We again note that in most cases the free samples choices are deterministic: Firm i chooses free samples $q_i = 0$ when $n_i(a_1, a_2) = (1,0,0,0)^T$, free samples $q_i = 200\ 000$ when $n_i(a_1, a_2) = (0,1,0,0)^T$ and free samples $q_i = 400\ 000$ when $n_i(a_1, a_2) = (0,0,1,0)^T$ and free samples $q_i = 600\ 000$ when $n_i(a_1, a_2) = (0,0,0,1)^T$ for certain. We also again note that free samples choices in the case with linear trend with strength $d = 2$ and planning horizons $T = 10$ is nondeterministic. As firm i's free samples choice strategy is in this case $n_i(2,2) = (0.20827, 0.56213, 0.2296, 0)^T$, firm i chooses free samples $q_i = 0$, $q_i = 200\ 000$ or $q_i = 400\ 000$.

Table 6.6 describes how free samples choices are made in the SPNE for impatient firms with discount rate $r = 10$. We note that free samples choices are deterministic in every case. With the vector $n_i(a_1, a_2) = (1,0,0,0)^T$ firm i chooses free samples $q_i = 0$ for certain, and with the vector $n_i(a_1, a_2) = (0,1,0,0)^T$ firm i chooses free samples $q_i = 200\ 000$ for certain.

Planning horizon	Trend type	(a_1, a_2)	$n_1(a_1, a_2)$	$n_2(a_1, a_2)$
$T = 10$	no trend	$(2, 2)$	$(1, 0, 0, 0)^T$	$(1, 0, 0, 0)^T$
	linear, $d = 2$	$(2, 2)$	$(0.34668, 0.21864, 0.43468, 0)^T$	$(0.34668, 0.21864, 0.43468, 0)^T$
	parabel, $d = 2$	$(1, 1)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(1, 2)$	$(1, 0, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(1, 3)$	$(0, 1, 0, 0)^T$	$(0, 0, 1, 0)^T$
		$(2, 1)$	$(0, 1, 0, 0)^T$	$(1, 0, 0, 0)^T$
		$(2, 2)$	$(1, 0, 0, 0)^T$	$(1, 0, 0, 0)^T$
		$(2, 3)$	$(1, 0, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(3, 1)$	$(0, 0, 1, 0)^T$	$(0, 1, 0, 0)^T$
		$(3, 2)$	$(0, 1, 0, 0)^T$	$(1, 0, 0, 0)^T$
		$(3, 3)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	linear, $d = 8$	$(3, 3)$	$(0, 0, 0, 1)^T$	$(0, 0, 0, 1)^T$
	parabel, $d = 8$	$(2, 2)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
$T = 100$	no trend	$(2, 2)$	$(1, 0, 0, 0)^T$	$(1, 0, 0, 0)^T$
	linear, $d = 2$	$(2, 2)$	$(0, 0, 1, 0)^T$	$(0, 0, 1, 0)^T$
	parabel, $d = 2$	$(1, 1)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(1, 3)$	$(0, 1, 0, 0)^T$	$(0, 0, 1, 0)^T$
		$(3, 1)$	$(0, 0, 1, 0)^T$	$(0, 1, 0, 0)^T$
		$(3, 3)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	linear, $d = 8$	$(3, 3)$	$(0, 0, 0, 1)^T$	$(0, 0, 0, 1)^T$
	parabel, $d = 8$	$(2, 2)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$

Table 6.4: Second stage normal-form game SPNE strategies, patient firms, $r = 0$

Planning horizon	Trend type	(a_1, a_2)	$n_1(a_1, a_2)$	$n_2(a_1, a_2)$
$T = 10$	no trend	$(2, 2)$	$(1, 0, 0, 0)^T$	$(1, 0, 0, 0)^T$
	linear, $d = 2$	$(2, 2)$	$(0.20827, 0.56213, 0.2296, 0)^T$	$(0.20827, 0.56213, 0.2296, 0)^T$
	parabel, $d = 2$	$(1, 1)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(1, 2)$	$(1, 0, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(1, 3)$	$(0, 1, 0, 0)^T$	$(0, 0, 1, 0)^T$
		$(2, 1)$	$(0, 1, 0, 0)^T$	$(1, 0, 0, 0)^T$
		$(2, 2)$	$(1, 0, 0, 0)^T$	$(1, 0, 0, 0)^T$
		$(2, 3)$	$(1, 0, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(3, 1)$	$(0, 0, 1, 0)^T$	$(0, 1, 0, 0)^T$
		$(3, 2)$	$(0, 1, 0, 0)^T$	$(1, 0, 0, 0)^T$
		$(3, 3)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	linear, $d = 8$	$(3, 3)$	$(0, 0, 1, 0)^T$	$(0, 0, 1, 0)^T$
	parabel, $d = 8$	$(2, 2)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
$T = 100$	no trend	$(2, 2)$	$(1, 0, 0, 0)^T$	$(1, 0, 0, 0)^T$
	linear, $d = 2$	$(1, 2)$	$(1, 0, 0, 0)^T$	$(1, 0, 0, 0)^T$
	parabel, $d = 2$	$(1, 1)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(1, 2)$	$(1, 0, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(1, 3)$	$(0, 1, 0, 0)^T$	$(0, 0, 1, 0)^T$
		$(2, 1)$	$(0, 1, 0, 0)^T$	$(1, 0, 0, 0)^T$
		$(2, 2)$	$(1, 0, 0, 0)^T$	$(1, 0, 0, 0)^T$
		$(2, 3)$	$(1, 0, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(3, 1)$	$(0, 0, 1, 0)^T$	$(0, 1, 0, 0)^T$
		$(3, 2)$	$(0, 1, 0, 0)^T$	$(1, 0, 0, 0)^T$
		$(3, 3)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	linear, $d = 8$	$(3, 3)$	$(0, 0, 0, 1)^T$	$(0, 0, 0, 1)^T$
	parabel, $d = 8$	$(2, 2)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$

Table 6.5: Second stage normal-form game SPNE strategies, normal firms, $r = 0.05$

Planning horizon	Trend type	(a_1, a_2)	$n_1(a_1, a_2)$	$n_2(a_1, a_2)$
$T = 10$	no trend	$(2, 2)$	$(1, 0, 0, 0)^T$	$(1, 0, 0, 0)^T$
	linear, $d = 2$	$(2, 2)$	$(1, 0, 0, 0)^T$	$(1, 0, 0, 0)^T$
	parabel, $d = 2$	$(2, 2)$	$(1, 0, 0, 0)^T$	$(1, 0, 0, 0)^T$
	linear, $d = 8$	$(2, 2)$	$(1, 0, 0, 0)^T$	$(1, 0, 0, 0)^T$
	parabel, $d = 8$	$(2, 1)$	$(1, 0, 0, 0)^T$	$(0, 1, 0, 0)^T$
$T = 100$	no trend	$(2, 2)$	$(1, 0, 0, 0)^T$	$(1, 0, 0, 0)^T$
	linear, $d = 2$	$(2, 2)$	$(1, 0, 0, 0)^T$	$(1, 0, 0, 0)^T$
	parabel, $d = 2$	$(2, 2)$	$(1, 0, 0, 0)^T$	$(1, 0, 0, 0)^T$
	linear, $d = 8$	$(2, 2)$	$(1, 0, 0, 0)^T$	$(1, 0, 0, 0)^T$
	parabel, $d = 8$	$(2, 1)$	$(1, 0, 0, 0)^T$	$(0, 1, 0, 0)^T$

Table 6.6: Second stage normal-form game SPNE strategies, impatient firms, $r = 10$

Third stage SPNE strategies $h_i(a_1, a_2, q_1, q_2)$

The price choices in the third stage depend on the quality choices in the first stage (determined by tables 6.1-6.3) and the free samples choices in the second stage (determined by tables 6.4-6.6). Tables 6.7-6.9 give the third stage normal-form game strategies of the SPNE that determine the price choices. We describe how the price choices are made by determining the vectors $h_i(a_1, a_2, q_1, q_2)$ that are dependent on the previous quality and free samples choices (a_1, a_2, q_1, q_2). The vectors $h_i(a_1, a_2, q_1, q_2)$ are probability distributions over the price set P of sets (3.8). Again, the first component of vector $h_i(a_1, a_2, q_1, q_2)$ is the probability that firm i chooses price $p_i = 0$, the second component the probability that firm i chooses price $p_i = 3$ etc.

We note that the price choices in the SPNE of all considered discount rates and cases are deterministic. With the vector $h_i(a_1, a_2, q_1, q_2) = (0, 1, 0, 0)^T$ firm i chooses price $p_i = 3$ for certain. With the vector $h_i(a_1, a_2, q_1, q_2) = (0, 0, 1, 0)^T$ firm i chooses price $p_i = 6$ for certain. With the vector $h_i(a_1, a_2, q_1, q_2) = (0, 0, 0, 1)^T$ firm i chooses price $p_i = 9$ for certain.

6.4.2 Observable firm behavior

As firm behavior in the SPNE may be nondeterministic, we are forced to consider a range of possible firm behavior profiles instead of a single pair of deterministic firm behavior profiles. Tables 6.10-6.12 give the possible firm behavior profiles and table 6.13 the expected SPNE profits.

The results again do not offer us a reason to doubt the internal validity of the model. We note that both firms will always set a positive price to guarantee nonnegative expected SPNE profits. Again, the SPNE behavior in each of the no trend cases for both firms is $(a_i, q_i, p_i) = (2, 0, 3), i = 1, 2$. This allows us to consider the effects of the planning horizon and the discount rate to the expected SPNE profits. We note that with longer planning horizons the expected SPNE profits are larger, and with larger discount rates the expected SPNE profits are correspondingly smaller.

Let us next consider the research questions: Different trend types again lead to observable differences in firm behavior but the specific type and strength of the trend may not be inferred purely from firm behavior. For example, we note that both (patient) firms with short planning horizons may end up choosing $(a_i, q_i, p_i) = (2, 0, 3)$ in cases with no trend, a linear trend with strength $d = 2$ or a parabel trend with strength $d = 2$ (see table 6.10, $T = 10$). Thus if the observer happened to know the discount rate and the planning horizons of both firms

Planning horizon	Trend type	(a_1, a_2, q_1, q_2)	$h_1(a_1, a_2, q_1, q_2)$	$h_2(a_1, a_2, q_1, q_2)$
$T = 10$	no trend	$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	linear, $d = 2$	$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 0, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 0, 400\,000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 200\,000, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 200\,000, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 200\,000, 400\,000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 400\,000, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 400\,000, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 400\,000, 400\,000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	parabel, $d = 2$	$(1, 1, 200\,000, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(1, 2, 0, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 0, 1, 0)^T$
		$(1, 3, 200\,000, 400\,000)$	$(0, 1, 0, 0)^T$	$(0, 0, 0, 1)^T$
		$(2, 1, 200\,000, 0)$	$(0, 0, 1, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 3, 0, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 0, 1, 0)^T$
		$(3, 1, 400\,000, 200\,000)$	$(0, 0, 0, 1)^T$	$(0, 1, 0, 0)^T$
		$(3, 2, 200\,000, 0)$	$(0, 0, 1, 0)^T$	$(0, 1, 0, 0)^T$
		$(3, 3, 200\,000, 200\,000)$	$(0, 0, 1, 0)^T$	$(0, 0, 1, 0)^T$
	linear, $d = 8$	$(3, 3, 600\,000, 600\,000)$	$(0, 0, 0, 1)^T$	$(0, 0, 0, 1)^T$
	parabel, $d = 8$	$(2, 2, 200\,000, 200\,000)$	$(0, 0, 0, 1)^T$	$(0, 0, 0, 1)^T$
$T = 100$	no trend	$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	linear, $d = 2$	$(2, 2, 400\,000, 400\,000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	parabel, $d = 2$	$(1, 1, 200\,000, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(1, 3, 200\,000, 400\,000)$	$(0, 1, 0, 0)^T$	$(0, 0, 0, 1)^T$
		$(3, 1, 400\,000, 200\,000)$	$(0, 0, 0, 1)^T$	$(0, 1, 0, 0)^T$
		$(3, 3, 200\,000, 200\,000)$	$(0, 0, 1, 0)^T$	$(0, 0, 1, 0)^T$
	linear, $d = 8$	$(3, 3, 600\,000, 600\,000)$	$(0, 0, 0, 1)^T$	$(0, 0, 0, 1)^T$
	parabel, $d = 8$	$(2, 2, 200\,000, 200\,000)$	$(0, 0, 0, 1)^T$	$(0, 0, 0, 1)^T$

Table 6.7: Third stage normal-form game SPNE strategies, patient firms, $r = 0$

Planning horizon	Trend type	(a_1, a_2, q_1, q_2)	$h_1(a_1, a_2, q_1, q_2)$	$h_2(a_1, a_2, q_1, q_2)$
$T = 10$	no trend	$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	linear, $d = 2$	$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 0, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 0, 400\,000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 200\,000, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 200\,000, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 200\,000, 400\,000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 400\,000, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 400\,000, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 400\,000, 400\,000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	parabel, $d = 2$	$(1, 1, 200\,000, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(1, 2, 0, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 0, 1, 0)^T$
		$(1, 3, 200\,000, 400\,000)$	$(0, 1, 0, 0)^T$	$(0, 0, 0, 1)^T$
		$(2, 1, 200\,000, 0)$	$(0, 0, 1, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 3, 0, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 0, 1, 0)^T$
		$(3, 1, 400\,000, 200\,000)$	$(0, 0, 0, 1)^T$	$(0, 1, 0, 0)^T$
		$(3, 2, 200\,000, 0)$	$(0, 0, 1, 0)^T$	$(0, 1, 0, 0)^T$
		$(3, 3, 200\,000, 200\,000)$	$(0, 0, 1, 0)^T$	$(0, 0, 1, 0)^T$
	linear, $d = 8$	$(3, 3, 400\,000, 400\,000)$	$(0, 0, 0, 1)^T$	$(0, 0, 0, 1)^T$
	parabel, $d = 8$	$(2, 2, 200\,000, 200\,000)$	$(0, 0, 0, 1)^T$	$(0, 0, 0, 1)^T$
$T = 100$	no trend	$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	linear, $d = 2$	$(1, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	parabel, $d = 2$	$(1, 1, 200\,000, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(1, 2, 0, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 0, 1, 0)^T$
		$(1, 3, 200\,000, 400\,000)$	$(0, 1, 0, 0)^T$	$(0, 0, 0, 1)^T$
		$(2, 1, 200\,000, 0)$	$(0, 0, 1, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
		$(2, 3, 0, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 0, 1, 0)^T$
		$(3, 1, 400\,000, 200\,000)$	$(0, 0, 0, 1)^T$	$(0, 1, 0, 0)^T$
		$(3, 2, 200\,000, 0)$	$(0, 0, 1, 0)^T$	$(0, 1, 0, 0)^T$
		$(3, 3, 200\,000, 200\,000)$	$(0, 0, 1, 0)^T$	$(0, 0, 1, 0)^T$
	linear, $d = 8$	$(3, 3, 600\,000, 600\,000)$	$(0, 0, 0, 1)^T$	$(0, 0, 0, 1)^T$
	parabel, $d = 8$	$(2, 2, 200\,000, 200\,000)$	$(0, 0, 0, 1)^T$	$(0, 0, 0, 1)^T$

Table 6.8: Third stage normal-form game SPNE strategies, normal firms, $r = 0.05$

Planning horizon	Trend type	(a_1, a_2, q_1, q_2)	$h_1(a_1, a_2, q_1, q_2)$	$h_2(a_1, a_2, q_1, q_2)$
$T = 10$	no trend	$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	linear, $d = 2$	$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	parabel, $d = 2$	$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	linear, $d = 8$	$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	parabel, $d = 8$	$(2, 1, 0, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 0, 1, 0)^T$
$T = 100$	no trend	$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	linear, $d = 2$	$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	parabel, $d = 2$	$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	linear, $d = 8$	$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 0)^T$
	parabel, $d = 8$	$(2, 1, 0, 200\,000)$	$(0, 1, 0, 0)^T$	$(0, 0, 1, 0)^T$

Table 6.9: Third stage normal-form game SPNE strategies, impatient firms, $r = 10$

(clearly an unrealistic assumption), the trend type could still not be inferred from observing both firms choosing $(a_i, q_i, p_i) = (2, 0, 3), i = 1, 2$.

We again note that there exists cases with trends such that in the SPNE one or both firms give out free products as samples. As with the two preceeding models, we may again present an existence result for the trends: If we observe either firm giving out free products, we may infer that a trend exists. Again, the result does not run in the other direction, i.e. we may not infer the nonexistence of a trend from observing that neither firm gives out free samples.

Even though the firms are symmetric, we note that the SPNE strategies are not necessarily symmetric. The nonsymmetric SPNE strategies lead to nonsymmetric expected SPNE profits. We find three cases of this type in table 6.13: See the cases with impatient firms with discount rate $r = 10$ and parabel trends with strength $d = 8$ or the case with normal firms with discount rate $r = 0.05$ and planning horizon $T = 100$ and linear trend with strength $d = 2$. Appendix A reveals that for the last case there are three NEa in the first stage normal-form game. For the first two cases there are three NEa in a few of the second and third stage normal-form games. We may then raise the question whether we could find a SPNE that leads to symmetric expected SPNE profits with different equilibrium choices. Additionally, we note that the nonsymmetric SPNE strategies guarantee that the observable firm behavior is nonsymmetric, although the symmetric SPNE strategies also allow the observable firm behavior to be nonsymmetric.

We may note that firms randomize differently depending on the type and strength of the trend. The firms randomize over their free samples choices when there is a linear trend (see tables 6.4 and 6.5), and over quality choices when there is a parabel trend (see tables 6.1 and 6.2). This behavior is also only evident when there is a weak trend with strength $d = 2$ and disappears in cases with strong trends with strength $d = 8$. Moreover, the planning horizon seems to have an effect on the randomization. We first note that the randomization over the free samples choices disappears as planning horizon lenghtens from $T = 10$ to $T = 100$. Next, if we interpret that a large amount of observable firm behavior profiles signifies “strong” randomization, we note that randomization is stronger in cases with shorter planning horizons. This last observation is based solely on tables 6.10 and 6.11 and may be of little use when studying trends in more general settings.

Planning horizon	Trend type	(a_1, q_1, p_1)	(a_2, q_2, p_2)
$T = 10$	no trend	$(2, 0, 3)$	$(2, 0, 3)$
	linear, $d = 2$	$(2, 0, 3)$	$(2, 0, 3)$
		$(2, 0, 3)$	$(2, 200\,000, 3)$
		$(2, 0, 3)$	$(2, 400\,000, 3)$
		$(2, 200\,000, 3)$	$(2, 0, 3)$
		$(2, 200\,000, 3)$	$(2, 200\,000, 3)$
		$(2, 200\,000, 3)$	$(2, 400\,000, 3)$
		$(2, 400\,000, 3)$	$(2, 0, 3)$
		$(2, 400\,000, 3)$	$(2, 200\,000, 3)$
		$(2, 400\,000, 3)$	$(2, 400\,000, 3)$
	parabel, $d = 2$	$(1, 200\,000, 3)$	$(1, 200\,000, 3)$
		$(1, 0, 3)$	$(2, 200\,000, 6)$
		$(1, 200\,000, 3)$	$(3, 400\,000, 9)$
		$(2, 200\,000, 6)$	$(1, 0, 3)$
		$(2, 0, 3)$	$(2, 0, 3)$
		$(2, 0, 3)$	$(3, 200\,000, 6)$
		$(3, 400\,000, 9)$	$(1, 200\,000, 3)$
		$(3, 200\,000, 6)$	$(2, 0, 3)$
		$(3, 200\,000, 6)$	$(3, 200\,000, 6)$
	linear, $d = 8$	$(3, 600\,000, 9)$	$(3, 600\,000, 9)$
	parabel, $d = 8$	$(2, 200\,000, 9)$	$(2, 200\,000, 9)$
$T = 100$	no trend	$(2, 0, 3)$	$(2, 0, 3)$
	linear, $d = 2$	$(2, 400\,000, 3)$	$(2, 400\,000, 3)$
	parabel, $d = 2$	$(1, 200\,000, 3)$	$(1, 200\,000, 3)$
		$(1, 200\,000, 3)$	$(3, 400\,000, 9)$
		$(3, 400\,000, 9)$	$(1, 200\,000, 3)$
		$(3, 200\,000, 6)$	$(3, 200\,000, 6)$
	linear, $d = 8$	$(3, 600\,000, 9)$	$(3, 600\,000, 9)$
	parabel, $d = 8$	$(2, 200\,000, 9)$	$(2, 200\,000, 9)$

Table 6.10: SPNE firm behavior, patient firms, $r = 0$

Planning horizon	Trend type	(a_1, q_1, p_1)	(a_2, q_2, p_2)
$T = 10$	no trend	$(2, 0, 3)$	$(2, 0, 3)$
	linear, $d = 2$	$(2, 0, 3)$	$(2, 0, 3)$
		$(2, 0, 3)$	$(2, 200\,000, 3)$
		$(2, 0, 3)$	$(2, 400\,000, 3)$
		$(2, 200\,000, 3)$	$(2, 0, 3)$
		$(2, 200\,000, 3)$	$(2, 200\,000, 3)$
		$(2, 200\,000, 3)$	$(2, 400\,000, 3)$
		$(2, 400\,000, 3)$	$(2, 0, 3)$
		$(2, 400\,000, 3)$	$(2, 200\,000, 3)$
		$(2, 400\,000, 3)$	$(2, 400\,000, 3)$
	parabel, $d = 2$	$(1, 200\,000, 3)$	$(1, 200\,000, 3)$
		$(1, 0, 3)$	$(2, 200\,000, 6)$
		$(1, 200\,000, 3)$	$(3, 400\,000, 9)$
		$(2, 200\,000, 6)$	$(1, 0, 3)$
		$(2, 0, 3)$	$(2, 0, 3)$
		$(2, 0, 3)$	$(3, 200\,000, 6)$
		$(3, 400\,000, 9)$	$(1, 200\,000, 3)$
		$(3, 200\,000, 6)$	$(2, 0, 3)$
		$(3, 200\,000, 6)$	$(3, 200\,000, 6)$
	linear, $d = 8$	$(3, 400\,000, 9)$	$(3, 400\,000, 9)$
	parabel, $d = 8$	$(2, 200\,000, 9)$	$(2, 200\,000, 9)$
$T = 100$	no trend	$(2, 0, 3)$	$(2, 0, 3)$
	linear, $d = 2$	$(1, 0, 3)$	$(2, 0, 3)$
	parabel, $d = 2$	$(1, 200\,000, 3)$	$(1, 200\,000, 3)$
		$(1, 0, 3)$	$(2, 200\,000, 6)$
		$(1, 200\,000, 3)$	$(3, 400\,000, 9)$
		$(2, 200\,000, 6)$	$(1, 0, 3)$
		$(2, 0, 3)$	$(2, 0, 3)$
		$(2, 0, 3)$	$(3, 200\,000, 6)$
		$(3, 400\,000, 9)$	$(1, 200\,000, 3)$
		$(3, 200\,000, 6)$	$(2, 0, 3)$
		$(3, 200\,000, 6)$	$(3, 200\,000, 6)$
	linear, $d = 8$	$(3, 600\,000, 9)$	$(3, 600\,000, 9)$
	parabel, $d = 8$	$(2, 200\,000, 9)$	$(2, 200\,000, 9)$

Table 6.11: SPNE firm behavior, normal firms, $r = 0.05$

Planning horizon	Trend type	(a_1, q_1, p_1)	(a_2, q_2, p_2)
$T = 10$	no trend	$(2, 0, 3)$	$(2, 0, 3)$
	linear, $d = 2$	$(2, 0, 3)$	$(2, 0, 3)$
	parabel, $d = 2$	$(2, 0, 3)$	$(2, 0, 3)$
	linear, $d = 8$	$(2, 0, 3)$	$(2, 0, 3)$
	parabel, $d = 8$	$(2, 0, 3)$	$(1, 200\,000, 6)$
$T = 100$	no trend	$(2, 0, 3)$	$(2, 0, 3)$
	linear, $d = 2$	$(2, 0, 3)$	$(2, 0, 3)$
	parabel, $d = 2$	$(2, 0, 3)$	$(2, 0, 3)$
	linear, $d = 8$	$(2, 0, 3)$	$(2, 0, 3)$
	parabel, $d = 8$	$(2, 0, 3)$	$(1, 200\,000, 6)$

Table 6.12: SPNE firm behavior, impatient firms, $r = 10$

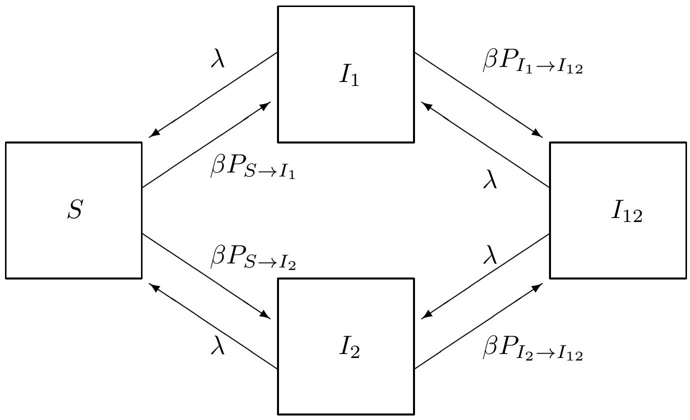


Figure 6.1: Compartmental Graph, Simultaneous-choice duopoly

-

Discount rate	Planning horizon	Trend type	π_1	π_2
$r = 0$	$T = 10$	no trend	397150	397150
		linear, $d = 2$	295220	295220
		parabel, $d = 2$	$1.76699 \cdot 10^6$	$1.76699 \cdot 10^6$
		linear, $d = 8$	$9.0159 \cdot 10^6$	$9.0159 \cdot 10^6$
		parabel, $d = 8$	$1.8715 \cdot 10^7$	$1.8715 \cdot 10^7$
	$T = 100$	no trend	$3.7722 \cdot 10^6$	$3.7722 \cdot 10^6$
		linear, $d = 2$	$6.3709 \cdot 10^6$	$6.3709 \cdot 10^6$
		parabel, $d = 2$	$3.29188 \cdot 10^7$	$3.29188 \cdot 10^7$
		linear, $d = 8$	$1.1702 \cdot 10^8$	$1.1702 \cdot 10^8$
		parabel, $d = 8$	$1.7863 \cdot 10^8$	$1.7863 \cdot 10^8$
$r = 0.05$	$T = 10$	no trend	315800	315800
		linear, $d = 2$	155230	155230
		parabel, $d = 2$	$1.07646 \cdot 10^6$	$1.07646 \cdot 10^6$
		linear, $d = 8$	$7.8276 \cdot 10^6$	$7.8276 \cdot 10^6$
		parabel, $d = 8$	$1.4874 \cdot 10^7$	$1.4874 \cdot 10^7$
	$T = 100$	no trend	765670	765670
		linear, $d = 2$	111450	$3.2737 \cdot 10^6$
		parabel, $d = 2$	$5.52684 \cdot 10^6$	$5.52684 \cdot 10^6$
		linear, $d = 8$	$2.0854 \cdot 10^7$	$2.0854 \cdot 10^7$
		parabel, $d = 8$	$3.6188 \cdot 10^7$	$3.6188 \cdot 10^7$
$r = 10$	$T = 10$	no trend	5225.2	5225.2
		linear, $d = 2$	5574.9	5574.9
		parabel, $d = 2$	8516.8	8516.8
		linear, $d = 8$	7232.4	7232.4
		parabel, $d = 8$	371.03	165390
	$T = 100$	no trend	5225.2	5225.2
		linear, $d = 2$	5574.9	5574.9
		parabel, $d = 2$	8516.8	8516.8
		linear, $d = 8$	7232.4	7232.4
		parabel, $d = 8$	371.03	165390

Table 6.13: SPNE expected profits

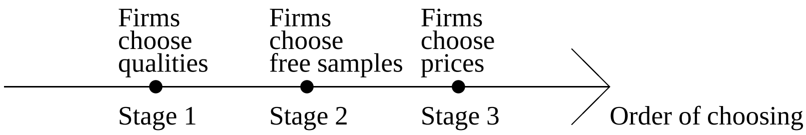
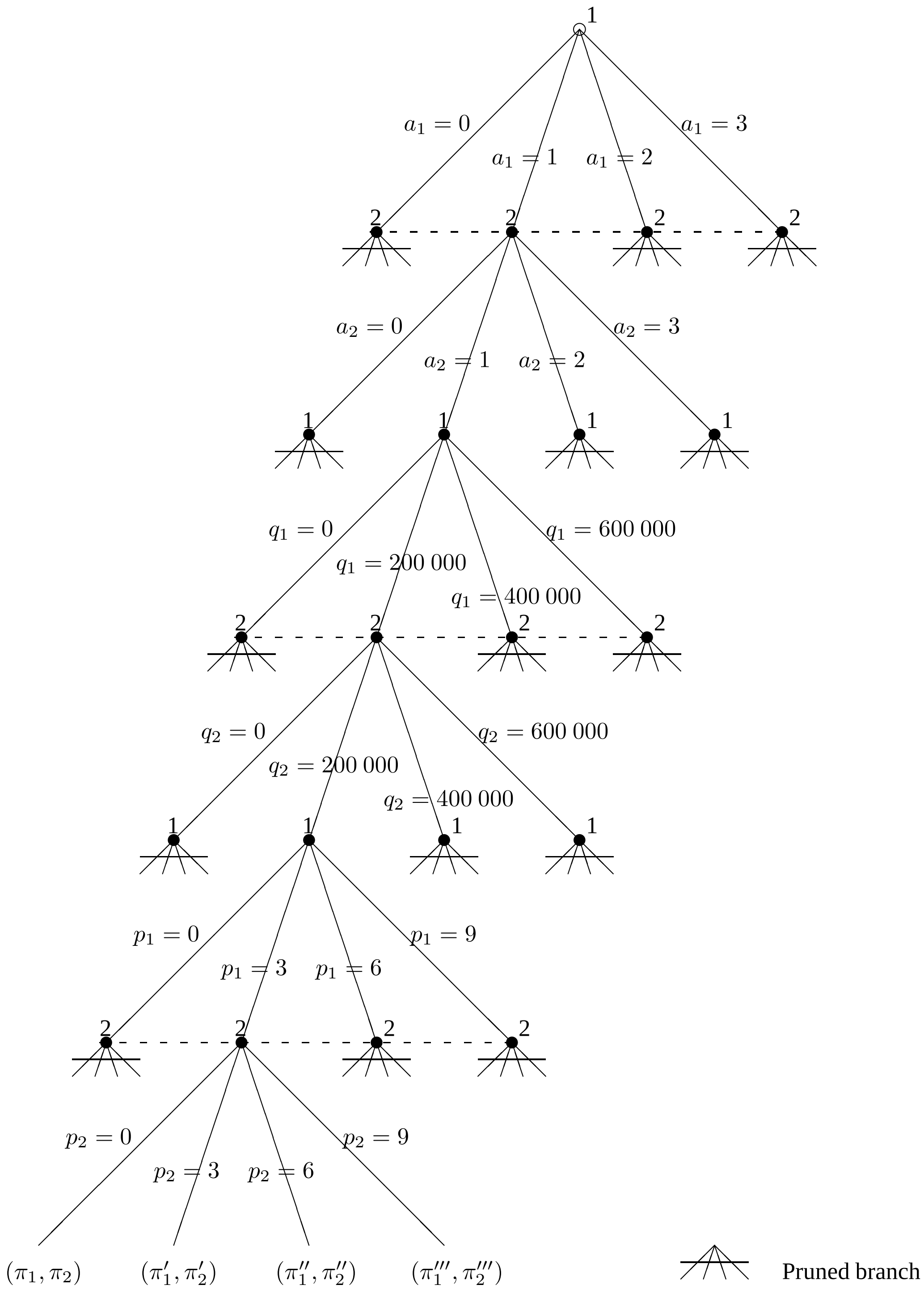


Figure 6.2: Duopoly Game



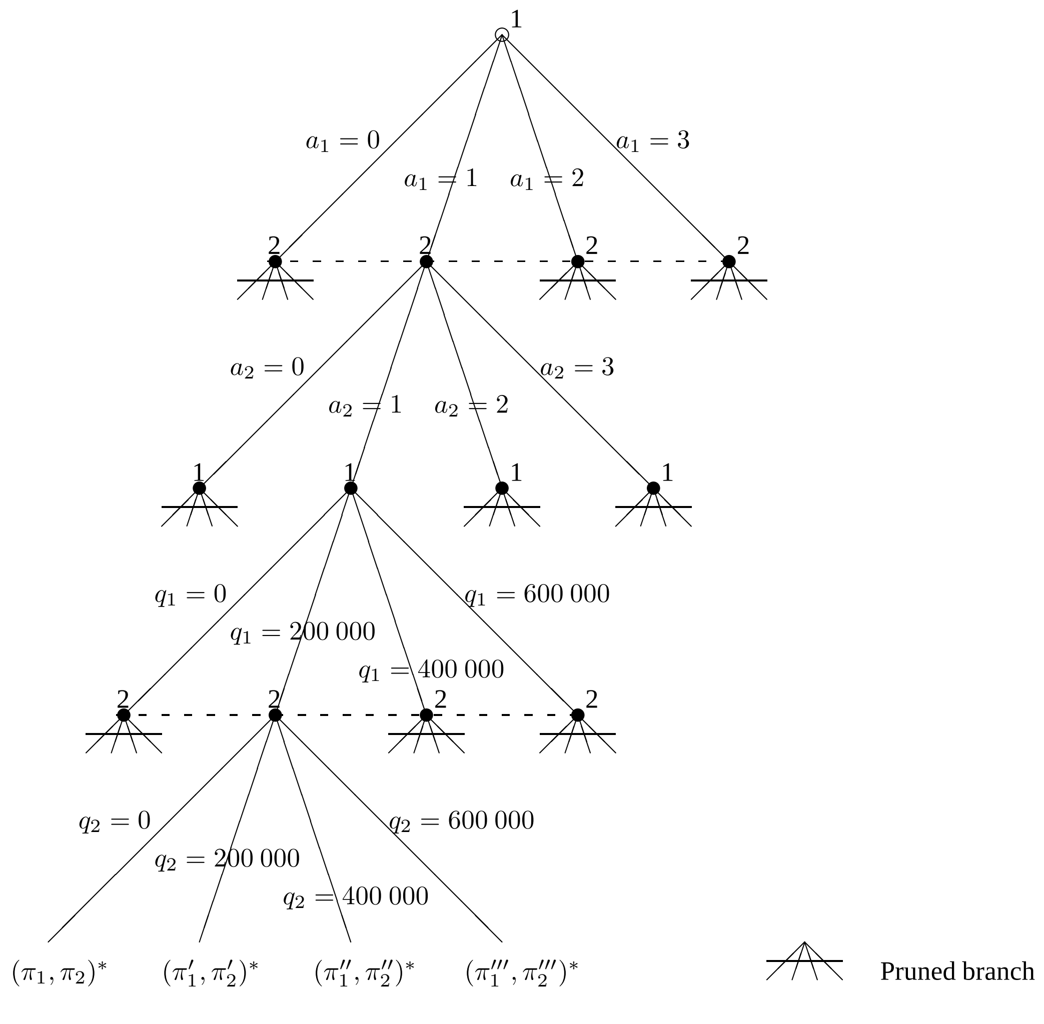
where $\pi_1''' = \pi_1(a_1 = 1, q_1 = 200\,000, p_1 = 3 \mid a_2 = 1, q_2 = 200\,000, p_2 = 9)$ etc.

Figure 6.3: Pruned Game Tree

		p_2			
		0	3	6	9
p_1	0	$\pi_1(0,0), \pi_2(0,0)$	$\pi_1(0,3), \pi_2(0,3)$	$\pi_1(0,6), \pi_2(0,6)$	$\pi_1(0,9), \pi_2(0,9)$
	3	$\pi_1(3,0), \pi_2(3,0)$	$\pi_1(3,3), \pi_2(3,3)$	$\pi_1(3,6), \pi_2(3,6)$	$\pi_1(3,9), \pi_2(3,9)$
	6	$\pi_1(6,0), \pi_2(6,0)$	$\pi_1(6,3), \pi_2(6,3)$	$\pi_1(6,6), \pi_2(6,6)$	$\pi_1(6,9), \pi_2(6,9)$
	9	$\pi_1(9,0), \pi_2(9,0)$	$\pi_1(9,3), \pi_2(9,3)$	$\pi_1(9,6), \pi_2(9,6)$	$\pi_1(9,9), \pi_2(9,9)$

where $\pi_i(p_1, p_2) = \pi_i(a_i, q_i, p_i \mid a_j, q_j, p_j)$ (equation 6.6)

Figure 6.4: Third stage normal-form game with history (a_1, a_2, q_1, q_2)



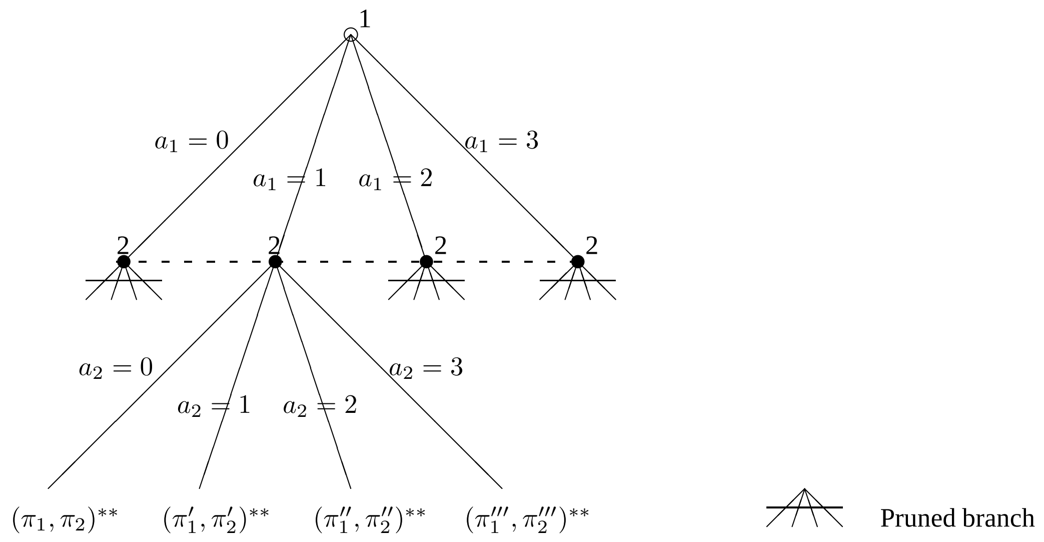
where π_1''' is firm 1's expected NE profits from third stage normal-form game with history $(a_1, a_2, q_1, q_2) = (1, 1, 200\,000, 600\,000)$ etc.

Figure 6.5: Truncated Game Tree, Second stage

		q_2			
		Q_1	Q_2	Q_3	Q_4
q_1	Q_1	$\pi_1(Q_1,Q_1),\pi_2(Q_1,Q_1)$	$\pi_1(Q_1,Q_2),\pi_2(Q_1,Q_2)$	$\pi_1(Q_1,Q_3),\pi_2(Q_1,Q_3)$	$\pi_1(Q_1,Q_4),\pi_2(Q_1,Q_4)$
	Q_2	$\pi_1(Q_2,Q_1),\pi_2(Q_2,Q_1)$	$\pi_1(Q_2,Q_2),\pi_2(Q_2,Q_2)$	$\pi_1(Q_2,Q_3),\pi_2(Q_2,Q_3)$	$\pi_1(Q_2,Q_4),\pi_2(Q_2,Q_4)$
	Q_3	$\pi_1(Q_3,Q_1),\pi_2(Q_3,Q_1)$	$\pi_1(Q_3,Q_2),\pi_2(Q_3,Q_2)$	$\pi_1(Q_3,Q_3),\pi_2(Q_3,Q_3)$	$\pi_1(Q_3,Q_4),\pi_2(Q_3,Q_4)$
	Q_4	$\pi_1(Q_4,Q_1),\pi_2(Q_4,Q_1)$	$\pi_1(Q_4,Q_2),\pi_2(Q_4,Q_2)$	$\pi_1(Q_4,Q_3),\pi_2(Q_4,Q_3)$	$\pi_1(Q_4,Q_4),\pi_2(Q_4,Q_4)$

where $Q_1 = 0, Q_2 = 200\,000, Q_3 = 400\,000, Q_4 = 600\,000$, $\pi_i(q_1, q_2) = \pi_i^{\text{Third Stage}}(h_1^*, h_2^*)$ with history (a_1, a_2, q_1, q_2)

Figure 6.6: Second stage normal-form game with history (a_1, a_2)



where π_1''' is firm 1's expected NE profits from second stage normal-form game with history $(a_1, a_2) = (1, 3)$

Figure 6.7: Truncated Game Tree, First stage

		a_2			
		0	1	2	3
a_1	0	$\pi_1(0,0), \pi_2(0,0)$	$\pi_1(0,1), \pi_2(0,1)$	$\pi_1(0,2), \pi_2(0,2)$	$\pi_1(0,3), \pi_2(0,3)$
	1	$\pi_1(1,0), \pi_2(1,0)$	$\pi_1(1,1), \pi_2(1,1)$	$\pi_1(1,2), \pi_2(1,2)$	$\pi_1(1,3), \pi_2(1,3)$
	2	$\pi_1(2,0), \pi_2(2,0)$	$\pi_1(2,1), \pi_2(2,1)$	$\pi_1(2,2), \pi_2(2,2)$	$\pi_1(2,3), \pi_2(2,3)$
	3	$\pi_1(3,0), \pi_2(3,0)$	$\pi_1(3,1), \pi_2(3,1)$	$\pi_1(3,2), \pi_2(3,2)$	$\pi_1(3,3), \pi_2(3,3)$

where $\pi_i(a_1, a_2) = \pi_i^{\text{Second Stage}}(n_1^*, n_2^*)$ with history (a_1, a_2)

Figure 6.8: First stage normal-form game

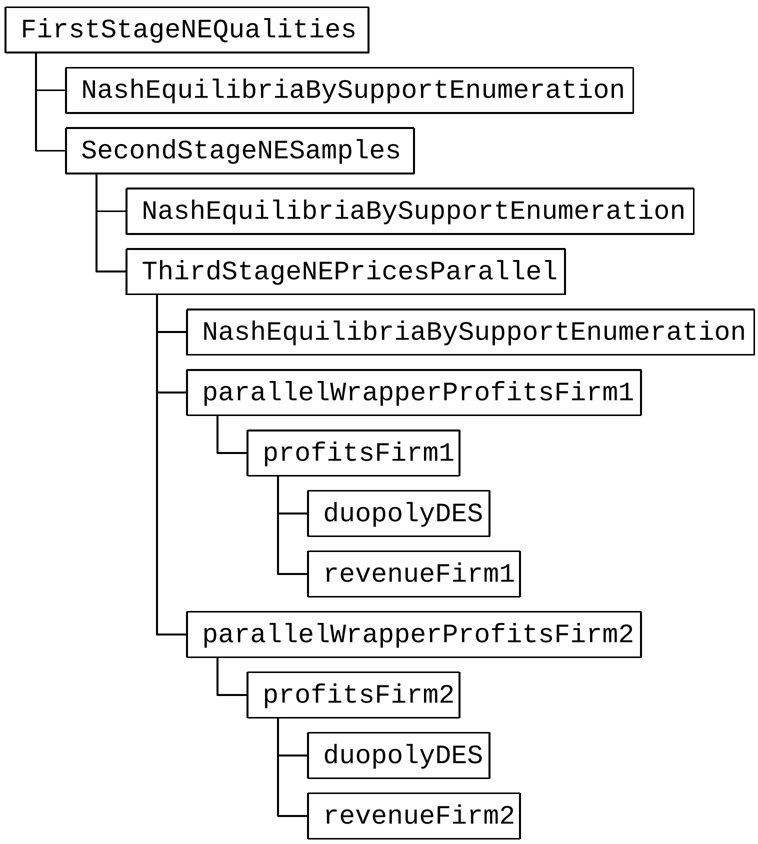


Figure 6.9: Script structure, Simultaneous-choice duopoly

7

Conclusions

7.1 Research summary

7.1.1 The definition of a trend

A trend is defined in this thesis as an effect to the consumer's indirect utility that is dependent solely on the number of products already in the market. The effect of the trend may be either positive or negative depending on the way the effect has been defined and the prevailing situation in the market. Indirect utility is used to determine the consumer's buying probabilities for a given product. This definition was chosen because it fits well one of the conceptions that arises from the word ”trend:“ If a product is trendy, then the probability that a particular consumer buys it is most likely dependent on how many others already have the product.

We may also imagine other conceptions for a trend. For example, we could require that when a product is trendy, its popularity should tail off after an initial boom. This would mean that the trend effects should be time-dependent and fading. Arguing further, we might require that when a product is trendy, it should disappear from the market after the trend has run its course. This would mean that the buying probabilities fade to zero as time goes on. The trend effects presented in this thesis can easily be modified to account for also these types of requirements.

In order to determine firm behavior in each of the presented models, it is necessary to give concrete functional forms for the trend effects. These functional forms are the ”no trend”, the linear and the parabel trend effect, and which seek to model the archetypal ideas of a trend. The ”no trend” effect is a constant zero to model the nonexistence of a trend. This gives us a benchmark for firm behavior by determining the behavior of firms in an environment lacking a trend. Next, the linear trend effect increases linearly as the number of the products already in the market increases. This arguably models markets where the product is a network good, i.e. the utility from the product increases as the number of other people owning the products increases. Finally, the parabel trend effect increases up to some saturation point after which it starts to decrease as the number of others already owning the product increases. This arguably models markets where the consumption of the product is affected by a ”club effect,“ i.e. the consumption of the product is increasingly desirable when it is limited to a small number of people, but when it becomes too widespread the consumption becomes less and less appealing.

7.1.2 Model specifications

The three models presented in this thesis are models of imperfect competition. In each of the models, a number of firms are selling their products to a large population. The demand for the products is determined by initial value problems which are modified versions of the initial value problems of the SIS model and incorporate the trend effects explicitly. The firms determine the forms of the initial value problems through their choice of quality, price and possible free samples of their products. The firms may choose to give out free samples in order to create demand through a possible trend. These choices are made once and for all from finite choice sets.

The first of the presented models is the monopoly model. In this model a single firm chooses its quality, free samples and price to maximize its profits. The monopoly model is very simple, and serves as the basis for building the next two models.

The second of the presented models is the Stackelberg duopoly model. In this model there are two firms, Leader and Follower, which have different market entry times with Leader entering the market first. As in the monopoly models, both of the firms choose the quality, free samples and price of their products to maximize their profits. Profit maximization is now complicated by the different market entry times. It is assumed that both firms know that behavior of the other firm is based on profit maximization. As Leader enters the market first and knows that Follower maximizes its profits, Leader incorporates this knowledge to its maximization problem and makes its choices accordingly.

The third and final of the presented models is the simultaneous-choice duopoly model. In this model there are two firms, firm 1 and 2, which enter the market at the same time. The firms choose the quality, free samples and price of their products in course of a game with three stages that the firms play prior to their entry to the market. The choices are made on the basis of strategies that the firms formulate at the start of the game to maximize their expected profits. The strategies are instructions on how to make choices at each possible stage of the game and may require the firms to randomize their choices at some stages. The randomization of the choices means that we have to consider a range of possible firm behavior rather than the deterministic firm behavior of the previous two models.

7.2 Results

The results in this thesis serve two functions: They provide us a way to consider the internal validity of the models and perhaps allow us to very minutely increase the overall knowledge of economics concerning trends.

By internal validity of the models I mean checking whether the results of the models are blatantly false. Examples for this kind of blatant falseness would be firms setting zero prices or receiving negative profits. On the whole, none of the results is blatantly wrong which means that we need not consider the models worthless out of hand.

Based on the results we note that firms behave similarly across a multitude of different trend types and strengths. We cannot therefore deduce the type or the strength of a trend by purely studying firm behavior. We may, however, formulate an existence result for a trend in each of the three models. We note that we may infer the existence of a trend in each model if we observe at least one firm giving out free samples. Additionally, we note that the result does not run in the other direction in any of the models, i.e. we may not infer the nonexistence of a trend if we observe no firms giving out free samples.

7.3 Criticism

When we construct models with simplistic assumptions, the assumptions are wisely chosen when everything within the model may be determined analytically. While the assumptions behind the three presented models in this thesis are simplistic, we note that analytical solutions were unattainable. The models can be characterized as forced marriages between the SIS model and the multinomial logit model for the buying decisions. As such, there are too many exogenous variables to be able to offer reasonable interpretations for the cumbersome formulas and results.

Future work would be needed in choosing new assumptions to guarantee analytical solutions while maintaining the essence of the presented models.

In the end, the validity of a model is determined by empirical tests. The goodness of a model is then derived from the possibility of constructing such tests. When we apply this criterion to the three proposed models, we note that the models are ”not even wrong,“ i.e. it is difficult to construct empirical tests to either validate or invalidate the results of the models. The models as such are therefore of little scientific merit. However, I hope that the models may serve as starting points for constructing empirically testable models of trendy goods.

Finally, we note that some of the stage games in the simultaneous-choice duopoly model have multiple Nash equilibria. This forces us to make difficult equilibrium choices and leads us thereby to question whether we might alter the game specification to avoid this. Perhaps a better game specification could be found with a more in depth knowledge of how real firms interact in duopoly situations.

7.4 Extensions

7.4.1 Epidemiological viewpoints

There are further possibilities for utilizing epidemiological models for modeling trends. I would like to highlight two possible approaches for economics applications from the wealth of epidemiological literature: the reproduction rate R_0 and the SIRS model.

From the SIS model and other epidemiological model it is possible to calculate the so-called reproduction rate R_0 (see e.g. Brauer and Castillo-Chávez (2001, p. 353, 412)). This number describes the mean number of secondary infections per a single primary infection. The number has also an interesting property regarding the outbreak of an epidemic: If the reproduction rate R_0 is greater than one, there will be an epidemic in the population, and if the reproduction rate is smaller or equal to one, the disease will die out in the population. If we were able to calculate an equivalent number in models of trendy goods to describe the mean number of secondary purchases stemming from a single primary purchase, we would have a natural measure of trendiness. The higher the hypothetical ”reproduction rate for purchases“ would be, the more trendy we could call the product.

Next, the Susceptible-Infective-Resistant-Susceptible model (Brauer and Castillo-Chávez (2001, p. 427)) could have interesting applications in economics. In this model the cured individuals move to a new compartment R, denoting that they have a temporary immunity to and do not spread the disease. The immunity disappears after a period of time and the individuals move to the compartment S, denoting that they are again susceptible to the disease. If we would apply this compartmental structure to the models of trendy goods, we could interpret the time period that the individual is in compartment R as a period of indignation stemming from a product breakdown when the individual refuses to consider buying any products. This approach, combined with product lifetimes that have been endogenized to firms’ choices, could be used to study whether firms should produce flimsy products on purpose to guarantee resales or whether firms should maximize the lifetime of their products to maximize their profits.

7.4.2 Economical viewpoints

I would like to highlight two major extension to the presented models that perhaps could be done using the standard tools of economics. These are the use of intervals as choice sets and optimal control.

When we use finite sets as the firms’ choice sets, we implicitly define the range of possible endogenous behavior beforehand. This is especially problematic if we would wish to fit our results to empirical observations as we would need to define the finite choice sets with supernatural care. It is therefore desirable to get rid of this assumption and rather study maximization problems where the firms choose the quality, price and free samples of their products from intervals such as the positive real numbers. It would then be necessary to use more sophisticated optimization methods to solve the firms' maximization problems. Applying the new optimization methods to the duopoly models would then provide us with two interesting problems: Firstly, we would need to handle the best-responding in the Stackelberg duopoly model in a feasible manner. Secondly, when the action sets in the proposed game in the simultaneous-choice duopoly model are intervals rather than finite sets, we cannot rely on Nash's existence theorem to guarantee the existence of Nash equilibrium in each of the stage games. Solving these problems seems to be very much nontrivial.

Real firms vary the price of their products all the time. It is therefore desirable to allow the firms vary their prices also in the proposed models. When the market evolution is described with price-dependent differential equations and the firms are allowed to vary their prices over time, we may use optimal control for the monopoly model or differential games for the duopoly models to study the behavior of the firms. Rather than choosing a constant price, the firms would now choose a price schedule that describes the price of the product as a function of time. We could then study markets where the price of a product is determined by a price schedule and the trend effects of the product are decreasing in time. If the optimal price schedules would also be decreasing in time, we might ask ourselves could we explain the existence of clearance sales with the existence of decreasing trend effects in, for example, markets for clothes.

Bibliography

Simon Anderson, André de Palma, and Jacques-François Thisse. *Discrete Choice Theory of Product Differentation*. The MIT Press, 1992.

David Avis, Gabriel D. Rosenberg, Rahul Savani, and Bernhard von Stengel. Enumeration of Nash equilibria for two-player games. *Economic Theory*, 42(1):9–37, 2010.

Fred Brauer and Carlos Castillo-Chávez. *Mathematical Models in Population Biology and Epidemiology*. Springer-Verlag New York, 2nd edition, 2001.

Steven C. Chapra. *Applied Numerical Methods with MATLAB for Engineers and Scientists*. McGraw-Hill, 2nd edition, 2005.

Richard M. Feldman and Ciriaco Valdez-Flores. *Applied Probability and Stochastic Processes*. Springer, 2nd edition, 2010.

Drew Fudenberg and Jean Tirole. *Game Theory*. The MIT Press, 1991.

S.R.K. Iyengar and R.K. Jain. *Numerical Methods*. New Age International (P) Ltd., 2009.

R. Kent Nagle, Edward B. Saff, and Arthur David Snider. *Fundamentals of Differential Equations*. Addison-Wesley, 8th edition, 2011.

Appendix A

Count of Nash equilibria in stage normal-form games

As mentioned before the Nash equilibrium strategies at any particular normal-form game may not be unique. In the calculations for this thesis each normal-form game will either have a single unique Nash equilibrium, or three separate Nash equilibria. Tables A.1, A.2 and A.3 give the counts of Nash equilibria at each stage of the proposed game.

Discount rate	Planning horizon	Trend type	Games with one Nash equilibrium	Games with three Nash equilibria
$r = 0$	$T = 10$	no trend	1	0
		linear $d = 2$	1	0
		parabel $d = 2$	1	0
		linear $d = 8$	1	0
		parabel $d = 8$	1	0
	$T = 100$	no trend	1	0
		linear $d = 2$	1	0
		parabel $d = 2$	0	1
		linear $d = 8$	1	0
		parabel $d = 8$	1	0
$r = 0.05$	$T = 10$	no trend	1	0
		linear $d = 2$	1	0
		parabel $d = 2$	1	0
		linear $d = 8$	1	0
		parabel $d = 8$	1	0
	$T = 100$	no trend	1	0
		linear $d = 2$	0	1
		parabel $d = 2$	1	0
		linear $d = 8$	1	0
		parabel $d = 8$	1	0
$r = 10$	$T = 10$	no trend	1	0
		linear $d = 2$	1	0
		parabel $d = 2$	1	0
		linear $d = 8$	1	0
		parabel $d = 8$	1	0
	$T = 100$	no trend	1	0
		linear $d = 2$	1	0
		parabel $d = 2$	1	0
		linear $d = 8$	1	0
		parabel $d = 8$	1	0

Table A.1: Count of first stage normal-form game Nash equilibria

Discount rate	Planning horizon	Trend type	Games with one Nash equilibrium	Games with three Nash equilibria
$r = 0$	$T = 10$	no trend	16	0
		linear $d = 2$	15	1
		parabel $d = 2$	14	2
		linear $d = 8$	14	2
		parabel $d = 8$	16	0
	$T = 100$	no trend	16	0
		linear $d = 2$	15	1
		parabel $d = 2$	14	2
		linear $d = 8$	14	2
		parabel $d = 8$	16	0
$r = 0.05$	$T = 10$	no trend	16	0
		linear $d = 2$	16	0
		parabel $d = 2$	14	2
		linear $d = 8$	14	2
		parabel $d = 8$	16	0
	$T = 100$	no trend	16	0
		linear $d = 2$	15	1
		parabel $d = 2$	14	2
		linear $d = 8$	14	2
		parabel $d = 8$	16	0
$r = 10$	$T = 10$	no trend	16	0
		linear $d = 2$	16	0
		parabel $d = 2$	16	0
		linear $d = 8$	16	0
		parabel $d = 8$	15	1
	$T = 100$	no trend	16	0
		linear $d = 2$	16	0
		parabel $d = 2$	16	0
		linear $d = 8$	16	0
		parabel $d = 8$	15	1

Table A.2: Count of second stage normal-form game Nash equilibria

Discount rate	Planning horizon	Trend type	Games with one Nash equilibrium	Games with three Nash equilibria
$r = 0$	$T = 10$	no trend	256	0
		linear $d = 2$	256	0
		parabel $d = 2$	243	13
		linear $d = 8$	256	0
		parabel $d = 8$	256	0
	$T = 100$	no trend	256	0
		linear $d = 2$	256	0
		parabel $d = 2$	253	3
		linear $d = 8$	256	0
		parabel $d = 8$	256	0
$r = 0.05$	$T = 10$	no trend	256	0
		linear $d = 2$	256	0
		parabel $d = 2$	243	13
		linear $d = 8$	256	0
		parabel $d = 8$	256	0
	$T = 100$	no trend	256	0
		linear $d = 2$	256	0
		parabel $d = 2$	251	5
		linear $d = 8$	256	0
		parabel $d = 8$	256	0
$r = 10$	$T = 10$	no trend	256	0
		linear $d = 2$	256	0
		parabel $d = 2$	253	3
		linear $d = 8$	255	1
		parabel $d = 8$	254	2
	$T = 100$	no trend	256	0
		linear $d = 2$	256	0
		parabel $d = 2$	253	3
		linear $d = 8$	255	1
		parabel $d = 8$	254	2

Table A.3: Count of third stage normal-form game Nash equilibria

Appendix B

A complete subgame-perfect Nash equilibrium strategy pair

This appendix gives the subgame-perfect Nash equilibrium strategies for both firms for the parabel trend case with parameters $d = 2$, $r = 0.05$ and $T = 10$. Firm 1 is the row player and firm 2 the column player in the normal-form games. First stage strategy for firm 1 is $l_1 = (0, 0.0155, 0.58875, 0.39575)^T$ and for firm 2 $l_2 = (0, 0.0155, 0.58875, 0.39575)^T$.

(a_1, a_2)	$n_1(a_1, a_2)$	$n_2(a_1, a_2)$
$(0,0)$	$(1,0,0,0)^T$	$(1,0,0,0)^T$
$(0,1)$	$(1,0,0,0)^T$	$(0,1,0,0)^T$
$(0,2)$	$(1,0,0,0)^T$	$(0,1,0,0)^T$
$(0,3)$	$(1,0,0,0)^T$	$(0,1,0,0)^T$
$(1,0)$	$(0,1,0,0)^T$	$(1,0,0,0)^T$
$(1,1)$	$(0,1,0,0)^T$	$(0,1,0,0)^T$
$(1,2)$	$(1,0,0,0)^T$	$(0,1,0,0)^T$
$(1,3)$	$(0,1,0,0)^T$	$(0,0,1,0)^T$
$(2,0)$	$(0,1,0,0)^T$	$(1,0,0,0)^T$
$(2,1)$	$(0,1,0,0)^T$	$(1,0,0,0)^T$
$(2,2)$	$(1,0,0,0)^T$	$(1,0,0,0)^T$
$(2,3)$	$(1,0,0,0)^T$	$(0,1,0,0)^T$
$(3,0)$	$(0,1,0,0)^T$	$(1,0,0,0)^T$
$(3,1)$	$(0,0,1,0)^T$	$(0,1,0,0)^T$
$(3,2)$	$(0,1,0,0)^T$	$(1,0,0,0)^T$
$(3,3)$	$(0,1,0,0)^T$	$(0,1,0,0)^T$

Table B.1: Firms' 1 and 2 functions $n_1(a_1, a_2)$ and $n_1(a_1, a_2)$

(a_1, a_2, q_1, q_2)	$h_1(a_1, a_2, q_1, q_2)$	(a_1, a_2, q_1, q_2)	$h_1(a_1, a_2, q_1, q_2)$
$(0, 0, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 0, 0, 200000)$	$(0, 1, 0, 0)^T$
$(0, 0, 0, 400000)$	$(0, 1, 0, 0)^T$	$(0, 0, 0, 600000)$	$(0, 1, 0, 0)^T$
$(0, 0, 200000, 0)$	$(0, 1, 0, 0)^T$	$(0, 0, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 0, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 0, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(0, 0, 400000, 0)$	$(0, 1, 0, 0)^T$	$(0, 0, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 0, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 0, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(0, 0, 600000, 0)$	$(0, 1, 0, 0)^T$	$(0, 0, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 0, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 0, 600000, 600000)$	$(0, 1, 0, 0)^T$
$(0, 1, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 200000)$	$(0, 1, 0, 0)^T$
$(0, 1, 0, 400000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 600000)$	$(0, 1, 0, 0)^T$
$(0, 1, 200000, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 1, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 1, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(0, 1, 400000, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 1, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 1, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(0, 1, 600000, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 1, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 1, 600000, 600000)$	$(0, 1, 0, 0)^T$
$(0, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 2, 0, 200000)$	$(0, 1, 0, 0)^T$
$(0, 2, 0, 400000)$	$(0, 1, 0, 0)^T$	$(0, 2, 0, 600000)$	$(0, 1, 0, 0)^T$
$(0, 2, 200000, 0)$	$(0, 1, 0, 0)^T$	$(0, 2, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 2, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 2, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(0, 2, 400000, 0)$	$(0, 1, 0, 0)^T$	$(0, 2, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 2, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 2, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(0, 2, 600000, 0)$	$(0, 1, 0, 0)^T$	$(0, 2, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 2, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 2, 600000, 600000)$	$(0, 1, 0, 0)^T$
$(0, 3, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 3, 0, 200000)$	$(0, 1, 0, 0)^T$
$(0, 3, 0, 400000)$	$(0, 1, 0, 0)^T$	$(0, 3, 0, 600000)$	$(0, 1, 0, 0)^T$
$(0, 3, 200000, 0)$	$(0, 1, 0, 0)^T$	$(0, 3, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 3, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 3, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(0, 3, 400000, 0)$	$(0, 1, 0, 0)^T$	$(0, 3, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 3, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 3, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(0, 3, 600000, 0)$	$(0, 1, 0, 0)^T$	$(0, 3, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 3, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 3, 600000, 600000)$	$(0, 1, 0, 0)^T$

Table B.2: Firm 1's function h_1, partial

(a_1, a_2, q_1, q_2)	$h_1(a_1, a_2, q_1, q_2)$	(a_1, a_2, q_1, q_2)	$h_1(a_1, a_2, q_1, q_2)$
$(1, 0, 0, 0)$	$(0, 1, 0, 0)^T$	$(1, 0, 0, 200000)$	$(0, 1, 0, 0)^T$
$(1, 0, 0, 400000)$	$(0, 1, 0, 0)^T$	$(1, 0, 0, 600000)$	$(0, 1, 0, 0)^T$
$(1, 0, 200000, 0)$	$(0, 1, 0, 0)^T$	$(1, 0, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 0, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 0, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(1, 0, 400000, 0)$	$(0, 1, 0, 0)^T$	$(1, 0, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 0, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 0, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(1, 0, 600000, 0)$	$(0, 1, 0, 0)^T$	$(1, 0, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 0, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 0, 600000, 600000)$	$(0, 1, 0, 0)^T$
$(1, 1, 0, 0)$	$(0, 1, 0, 0)^T$	$(1, 1, 0, 200000)$	$(0, 1, 0, 0)^T$
$(1, 1, 0, 400000)$	$(0, 1, 0, 0)^T$	$(1, 1, 0, 600000)$	$(0, 1, 0, 0)^T$
$(1, 1, 200000, 0)$	$(0, 1, 0, 0)^T$	$(1, 1, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 1, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 1, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(1, 1, 400000, 0)$	$(0, 1, 0, 0)^T$	$(1, 1, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 1, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 1, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(1, 1, 600000, 0)$	$(0, 1, 0, 0)^T$	$(1, 1, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 1, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 1, 600000, 600000)$	$(0, 1, 0, 0)^T$
$(1, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(1, 2, 0, 200000)$	$(0, 1, 0, 0)^T$
$(1, 2, 0, 400000)$	$(0, 1, 0, 0)^T$	$(1, 2, 0, 600000)$	$(0, 1, 0, 0)^T$
$(1, 2, 200000, 0)$	$(0, 1, 0, 0)^T$	$(1, 2, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 2, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 2, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(1, 2, 400000, 0)$	$(0, 1, 0, 0)^T$	$(1, 2, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 2, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 2, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(1, 2, 600000, 0)$	$(0, 1, 0, 0)^T$	$(1, 2, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 2, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 2, 600000, 600000)$	$(0, 1, 0, 0)^T$
$(1, 3, 0, 0)$	$(0, 1, 0, 0)^T$	$(1, 3, 0, 200000)$	$(0, 1, 0, 0)^T$
$(1, 3, 0, 400000)$	$(0, 1, 0, 0)^T$	$(1, 3, 0, 600000)$	$(0, 1, 0, 0)^T$
$(1, 3, 200000, 0)$	$(0, 1, 0, 0)^T$	$(1, 3, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 3, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 3, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(1, 3, 400000, 0)$	$(0, 1, 0, 0)^T$	$(1, 3, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 3, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 3, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(1, 3, 600000, 0)$	$(0, 1, 0, 0)^T$	$(1, 3, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 3, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 3, 600000, 600000)$	$(0, 1, 0, 0)^T$

Table B.3: Firm 1's function h_1, continued

(a_1, a_2, q_1, q_2)	$h_1(a_1, a_2, q_1, q_2)$	(a_1, a_2, q_1, q_2)	$h_1(a_1, a_2, q_1, q_2)$
$(2, 0, 0, 0)$	$(0, 1, 0, 0)^T$	$(2, 0, 0, 200000)$	$(0, 1, 0, 0)^T$
$(2, 0, 0, 400000)$	$(0, 1, 0, 0)^T$	$(2, 0, 0, 600000)$	$(0, 1, 0, 0)^T$
$(2, 0, 200000, 0)$	$(0, 0, 1, 0)^T$	$(2, 0, 200000, 200000)$	$(0, 0, 1, 0)^T$
$(2, 0, 200000, 400000)$	$(0, 0, 1, 0)^T$	$(2, 0, 200000, 600000)$	$(0, 0, 1, 0)^T$
$(2, 0, 400000, 0)$	$(0, 0, 1, 0)^T$	$(2, 0, 400000, 200000)$	$(0, 0, 1, 0)^T$
$(2, 0, 400000, 400000)$	$(0, 0, 1, 0)^T$	$(2, 0, 400000, 600000)$	$(0, 0, 1, 0)^T$
$(2, 0, 600000, 0)$	$(0, 0, 1, 0)^T$	$(2, 0, 600000, 200000)$	$(0, 0, 1, 0)^T$
$(2, 0, 600000, 400000)$	$(0, 0, 1, 0)^T$	$(2, 0, 600000, 600000)$	$(0, 0, 1, 0)^T$
$(2, 1, 0, 0)$	$(0, 1, 0, 0)^T$	$(2, 1, 0, 200000)$	$(0, 1, 0, 0)^T$
$(2, 1, 0, 400000)$	$(0, 1, 0, 0)^T$	$(2, 1, 0, 600000)$	$(0, 1, 0, 0)^T$
$(2, 1, 200000, 0)$	$(0, 0, 1, 0)^T$	$(2, 1, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(2, 1, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(2, 1, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(2, 1, 400000, 0)$	$(0, 0, 1, 0)^T$	$(2, 1, 400000, 200000)$	$(0, 0, 1, 0)^T$
$(2, 1, 400000, 400000)$	$(0, 0, 1, 0)^T$	$(2, 1, 400000, 600000)$	$(0, 0, 1, 0)^T$
$(2, 1, 600000, 0)$	$(0, 0, 1, 0)^T$	$(2, 1, 600000, 200000)$	$(0, 0, 1, 0)^T$
$(2, 1, 600000, 400000)$	$(0, 0, 1, 0)^T$	$(2, 1, 600000, 600000)$	$(0, 0, 1, 0)^T$
$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(2, 2, 0, 200000)$	$(0, 1, 0, 0)^T$
$(2, 2, 0, 400000)$	$(0, 1, 0, 0)^T$	$(2, 2, 0, 600000)$	$(0, 1, 0, 0)^T$
$(2, 2, 200000, 0)$	$(0, 1, 0, 0)^T$	$(2, 2, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(2, 2, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(2, 2, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(2, 2, 400000, 0)$	$(0, 1, 0, 0)^T$	$(2, 2, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(2, 2, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(2, 2, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(2, 2, 600000, 0)$	$(0, 1, 0, 0)^T$	$(2, 2, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(2, 2, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(2, 2, 600000, 600000)$	$(0, 1, 0, 0)^T$
$(2, 3, 0, 0)$	$(0, 1, 0, 0)^T$	$(2, 3, 0, 200000)$	$(0, 1, 0, 0)^T$
$(2, 3, 0, 400000)$	$(0, 1, 0, 0)^T$	$(2, 3, 0, 600000)$	$(0, 1, 0, 0)^T$
$(2, 3, 200000, 0)$	$(0, 0, 1, 0)^T$	$(2, 3, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(2, 3, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(2, 3, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(2, 3, 400000, 0)$	$(0, 0, 1, 0)^T$	$(2, 3, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(2, 3, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(2, 3, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(2, 3, 600000, 0)$	$(0, 0, 1, 0)^T$	$(2, 3, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(2, 3, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(2, 3, 600000, 600000)$	$(0, 1, 0, 0)^T$

Table B.4: Firm 1's function h_1, continued

(a_1, a_2, q_1, q_2)	$h_1(a_1, a_2, q_1, q_2)$	(a_1, a_2, q_1, q_2)	$h_1(a_1, a_2, q_1, q_2)$
$(3, 0, 0, 0)$	$(0, 0, 1, 0)^T$	$(3, 0, 0, 200000)$	$(0, 0, 1, 0)^T$
$(3, 0, 0, 400000)$	$(0, 0, 1, 0)^T$	$(3, 0, 0, 600000)$	$(0, 0, 1, 0)^T$
$(3, 0, 200000, 0)$	$(0, 0, 0, 1)^T$	$(3, 0, 200000, 200000)$	$(0, 0, 0, 1)^T$
$(3, 0, 200000, 400000)$	$(0, 0, 0, 1)^T$	$(3, 0, 200000, 600000)$	$(0, 0, 0, 1)^T$
$(3, 0, 400000, 0)$	$(0, 0, 0, 1)^T$	$(3, 0, 400000, 200000)$	$(0, 0, 0, 1)^T$
$(3, 0, 400000, 400000)$	$(0, 0, 0, 1)^T$	$(3, 0, 400000, 600000)$	$(0, 0, 0, 1)^T$
$(3, 0, 600000, 0)$	$(0, 0, 0, 1)^T$	$(3, 0, 600000, 200000)$	$(0, 0, 0, 1)^T$
$(3, 0, 600000, 400000)$	$(0, 0, 0, 1)^T$	$(3, 0, 600000, 600000)$	$(0, 0, 0, 1)^T$
$(3, 1, 0, 0)$	$(0, 0, 1, 0)^T$	$(3, 1, 0, 200000)$	$(0, 0, 1, 0)^T$
$(3, 1, 0, 400000)$	$(0, 0, 1, 0)^T$	$(3, 1, 0, 600000)$	$(0, 0, 1, 0)^T$
$(3, 1, 200000, 0)$	$(0, 0, 1, 0)^T$	$(3, 1, 200000, 200000)$	$(0, 0, 1, 0)^T$
$(3, 1, 200000, 400000)$	$(0, 0, 1, 0)^T$	$(3, 1, 200000, 600000)$	$(0, 0, 1, 0)^T$
$(3, 1, 400000, 0)$	$(0, 0, 0, 1)^T$	$(3, 1, 400000, 200000)$	$(0, 0, 0, 1)^T$
$(3, 1, 400000, 400000)$	$(0, 0, 0, 1)^T$	$(3, 1, 400000, 600000)$	$(0, 0, 0, 1)^T$
$(3, 1, 600000, 0)$	$(0, 0, 0, 1)^T$	$(3, 1, 600000, 200000)$	$(0, 0, 0, 1)^T$
$(3, 1, 600000, 400000)$	$(0, 0, 0, 1)^T$	$(3, 1, 600000, 600000)$	$(0, 0, 0, 1)^T$
$(3, 2, 0, 0)$	$(0, 0, 1, 0)^T$	$(3, 2, 0, 200000)$	$(0, 0, 1, 0)^T$
$(3, 2, 0, 400000)$	$(0, 0, 1, 0)^T$	$(3, 2, 0, 600000)$	$(0, 0, 1, 0)^T$
$(3, 2, 200000, 0)$	$(0, 0, 1, 0)^T$	$(3, 2, 200000, 200000)$	$(0, 0, 1, 0)^T$
$(3, 2, 200000, 400000)$	$(0, 0, 1, 0)^T$	$(3, 2, 200000, 600000)$	$(0, 0, 1, 0)^T$
$(3, 2, 400000, 0)$	$(0, 0, 1, 0)^T$	$(3, 2, 400000, 200000)$	$(0, 0, 1, 0)^T$
$(3, 2, 400000, 400000)$	$(0, 0, 1, 0)^T$	$(3, 2, 400000, 600000)$	$(0, 0, 1, 0)^T$
$(3, 2, 600000, 0)$	$(0, 0, 1, 0)^T$	$(3, 2, 600000, 200000)$	$(0, 0, 1, 0)^T$
$(3, 2, 600000, 400000)$	$(0, 0, 1, 0)^T$	$(3, 2, 600000, 600000)$	$(0, 0, 1, 0)^T$
$(3, 3, 0, 0)$	$(0, 0, 1, 0)^T$	$(3, 3, 0, 200000)$	$(0, 0, 1, 0)^T$
$(3, 3, 0, 400000)$	$(0, 0, 1, 0)^T$	$(3, 3, 0, 600000)$	$(0, 0, 1, 0)^T$
$(3, 3, 200000, 0)$	$(0, 0, 0, 1)^T$	$(3, 3, 200000, 200000)$	$(0, 0, 1, 0)^T$
$(3, 3, 200000, 400000)$	$(0, 0, 1, 0)^T$	$(3, 3, 200000, 600000)$	$(0, 0, 1, 0)^T$
$(3, 3, 400000, 0)$	$(0, 0, 0, 1)^T$	$(3, 3, 400000, 200000)$	$(0, 0, 1, 0)^T$
$(3, 3, 400000, 400000)$	$(0, 0, 1, 0)^T$	$(3, 3, 400000, 600000)$	$(0, 0, 1, 0)^T$
$(3, 3, 600000, 0)$	$(0, 0, 0, 1)^T$	$(3, 3, 600000, 200000)$	$(0, 0, 1, 0)^T$
$(3, 3, 600000, 400000)$	$(0, 0, 1, 0)^T$	$(3, 3, 600000, 600000)$	$(0, 0, 1, 0)^T$

Table B.5: Firm 1's function h_1, continued and concluded

(a_1, a_2, q_1, q_2)	$h_2(a_1, a_2, q_1, q_2)$	(a_1, a_2, q_1, q_2)	$h_2(a_1, a_2, q_1, q_2)$
$(0, 0, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 0, 0, 200000)$	$(0, 1, 0, 0)^T$
$(0, 0, 0, 400000)$	$(0, 1, 0, 0)^T$	$(0, 0, 0, 600000)$	$(0, 1, 0, 0)^T$
$(0, 0, 200000, 0)$	$(0, 1, 0, 0)^T$	$(0, 0, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 0, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 0, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(0, 0, 400000, 0)$	$(0, 1, 0, 0)^T$	$(0, 0, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 0, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 0, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(0, 0, 600000, 0)$	$(0, 1, 0, 0)^T$	$(0, 0, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 0, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 0, 600000, 600000)$	$(0, 1, 0, 0)^T$
$(0, 1, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 200000)$	$(0, 1, 0, 0)^T$
$(0, 1, 0, 400000)$	$(0, 1, 0, 0)^T$	$(0, 1, 0, 600000)$	$(0, 1, 0, 0)^T$
$(0, 1, 200000, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 1, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 1, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(0, 1, 400000, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 1, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 1, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(0, 1, 600000, 0)$	$(0, 1, 0, 0)^T$	$(0, 1, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(0, 1, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(0, 1, 600000, 600000)$	$(0, 1, 0, 0)^T$
$(0, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(0, 2, 0, 200000)$	$(0, 0, 1, 0)^T$
$(0, 2, 0, 400000)$	$(0, 0, 1, 0)^T$	$(0, 2, 0, 600000)$	$(0, 0, 1, 0)^T$
$(0, 2, 200000, 0)$	$(0, 1, 0, 0)^T$	$(0, 2, 200000, 200000)$	$(0, 0, 1, 0)^T$
$(0, 2, 200000, 400000)$	$(0, 0, 1, 0)^T$	$(0, 2, 200000, 600000)$	$(0, 0, 1, 0)^T$
$(0, 2, 400000, 0)$	$(0, 1, 0, 0)^T$	$(0, 2, 400000, 200000)$	$(0, 0, 1, 0)^T$
$(0, 2, 400000, 400000)$	$(0, 0, 1, 0)^T$	$(0, 2, 400000, 600000)$	$(0, 0, 1, 0)^T$
$(0, 2, 600000, 0)$	$(0, 1, 0, 0)^T$	$(0, 2, 600000, 200000)$	$(0, 0, 1, 0)^T$
$(0, 2, 600000, 400000)$	$(0, 0, 1, 0)^T$	$(0, 2, 600000, 600000)$	$(0, 0, 1, 0)^T$
$(0, 3, 0, 0)$	$(0, 0, 1, 0)^T$	$(0, 3, 0, 200000)$	$(0, 0, 0, 1)^T$
$(0, 3, 0, 400000)$	$(0, 0, 0, 1)^T$	$(0, 3, 0, 600000)$	$(0, 0, 0, 1)^T$
$(0, 3, 200000, 0)$	$(0, 0, 1, 0)^T$	$(0, 3, 200000, 200000)$	$(0, 0, 0, 1)^T$
$(0, 3, 200000, 400000)$	$(0, 0, 0, 1)^T$	$(0, 3, 200000, 600000)$	$(0, 0, 0, 1)^T$
$(0, 3, 400000, 0)$	$(0, 0, 1, 0)^T$	$(0, 3, 400000, 200000)$	$(0, 0, 0, 1)^T$
$(0, 3, 400000, 400000)$	$(0, 0, 0, 1)^T$	$(0, 3, 400000, 600000)$	$(0, 0, 0, 1)^T$
$(0, 3, 600000, 0)$	$(0, 0, 1, 0)^T$	$(0, 3, 600000, 200000)$	$(0, 0, 0, 1)^T$
$(0, 3, 600000, 400000)$	$(0, 0, 0, 1)^T$	$(0, 3, 600000, 600000)$	$(0, 0, 0, 1)^T$

Table B.6: Firm 2's function h_2, partial

(a_1, a_2, q_1, q_2)	$h_2(a_1, a_2, q_1, q_2)$	(a_1, a_2, q_1, q_2)	$h_2(a_1, a_2, q_1, q_2)$
$(1, 0, 0, 0)$	$(0, 1, 0, 0)^T$	$(1, 0, 0, 200000)$	$(0, 1, 0, 0)^T$
$(1, 0, 0, 400000)$	$(0, 1, 0, 0)^T$	$(1, 0, 0, 600000)$	$(0, 1, 0, 0)^T$
$(1, 0, 200000, 0)$	$(0, 1, 0, 0)^T$	$(1, 0, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 0, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 0, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(1, 0, 400000, 0)$	$(0, 1, 0, 0)^T$	$(1, 0, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 0, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 0, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(1, 0, 600000, 0)$	$(0, 1, 0, 0)^T$	$(1, 0, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 0, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 0, 600000, 600000)$	$(0, 1, 0, 0)^T$
$(1, 1, 0, 0)$	$(0, 1, 0, 0)^T$	$(1, 1, 0, 200000)$	$(0, 1, 0, 0)^T$
$(1, 1, 0, 400000)$	$(0, 1, 0, 0)^T$	$(1, 1, 0, 600000)$	$(0, 1, 0, 0)^T$
$(1, 1, 200000, 0)$	$(0, 1, 0, 0)^T$	$(1, 1, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 1, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 1, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(1, 1, 400000, 0)$	$(0, 1, 0, 0)^T$	$(1, 1, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 1, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 1, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(1, 1, 600000, 0)$	$(0, 1, 0, 0)^T$	$(1, 1, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 1, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(1, 1, 600000, 600000)$	$(0, 1, 0, 0)^T$
$(1, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(1, 2, 0, 200000)$	$(0, 0, 1, 0)^T$
$(1, 2, 0, 400000)$	$(0, 0, 1, 0)^T$	$(1, 2, 0, 600000)$	$(0, 0, 1, 0)^T$
$(1, 2, 200000, 0)$	$(0, 1, 0, 0)^T$	$(1, 2, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 2, 200000, 400000)$	$(0, 0, 1, 0)^T$	$(1, 2, 200000, 600000)$	$(0, 0, 1, 0)^T$
$(1, 2, 400000, 0)$	$(0, 1, 0, 0)^T$	$(1, 2, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 2, 400000, 400000)$	$(0, 0, 1, 0)^T$	$(1, 2, 400000, 600000)$	$(0, 0, 1, 0)^T$
$(1, 2, 600000, 0)$	$(0, 1, 0, 0)^T$	$(1, 2, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(1, 2, 600000, 400000)$	$(0, 0, 1, 0)^T$	$(1, 2, 600000, 600000)$	$(0, 0, 1, 0)^T$
$(1, 3, 0, 0)$	$(0, 0, 1, 0)^T$	$(1, 3, 0, 200000)$	$(0, 0, 1, 0)^T$
$(1, 3, 0, 400000)$	$(0, 0, 0, 1)^T$	$(1, 3, 0, 600000)$	$(0, 0, 0, 1)^T$
$(1, 3, 200000, 0)$	$(0, 0, 1, 0)^T$	$(1, 3, 200000, 200000)$	$(0, 0, 1, 0)^T$
$(1, 3, 200000, 400000)$	$(0, 0, 0, 1)^T$	$(1, 3, 200000, 600000)$	$(0, 0, 0, 1)^T$
$(1, 3, 400000, 0)$	$(0, 0, 1, 0)^T$	$(1, 3, 400000, 200000)$	$(0, 0, 1, 0)^T$
$(1, 3, 400000, 400000)$	$(0, 0, 0, 1)^T$	$(1, 3, 400000, 600000)$	$(0, 0, 0, 1)^T$
$(1, 3, 600000, 0)$	$(0, 0, 1, 0)^T$	$(1, 3, 600000, 200000)$	$(0, 0, 1, 0)^T$
$(1, 3, 600000, 400000)$	$(0, 0, 0, 1)^T$	$(1, 3, 600000, 600000)$	$(0, 0, 0, 1)^T$

Table B.7: Firm 2's function h_2, continued

(a_1, a_2, q_1, q_2)	$h_2(a_1, a_2, q_1, q_2)$	(a_1, a_2, q_1, q_2)	$h_2(a_1, a_2, q_1, q_2)$
$(2, 0, 0, 0)$	$(0, 1, 0, 0)^T$	$(2, 0, 0, 200000)$	$(0, 1, 0, 0)^T$
$(2, 0, 0, 400000)$	$(0, 1, 0, 0)^T$	$(2, 0, 0, 600000)$	$(0, 1, 0, 0)^T$
$(2, 0, 200000, 0)$	$(0, 1, 0, 0)^T$	$(2, 0, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(2, 0, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(2, 0, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(2, 0, 400000, 0)$	$(0, 1, 0, 0)^T$	$(2, 0, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(2, 0, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(2, 0, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(2, 0, 600000, 0)$	$(0, 1, 0, 0)^T$	$(2, 0, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(2, 0, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(2, 0, 600000, 600000)$	$(0, 1, 0, 0)^T$
$(2, 1, 0, 0)$	$(0, 1, 0, 0)^T$	$(2, 1, 0, 200000)$	$(0, 1, 0, 0)^T$
$(2, 1, 0, 400000)$	$(0, 1, 0, 0)^T$	$(2, 1, 0, 600000)$	$(0, 1, 0, 0)^T$
$(2, 1, 200000, 0)$	$(0, 1, 0, 0)^T$	$(2, 1, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(2, 1, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(2, 1, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(2, 1, 400000, 0)$	$(0, 1, 0, 0)^T$	$(2, 1, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(2, 1, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(2, 1, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(2, 1, 600000, 0)$	$(0, 1, 0, 0)^T$	$(2, 1, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(2, 1, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(2, 1, 600000, 600000)$	$(0, 1, 0, 0)^T$
$(2, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(2, 2, 0, 200000)$	$(0, 1, 0, 0)^T$
$(2, 2, 0, 400000)$	$(0, 1, 0, 0)^T$	$(2, 2, 0, 600000)$	$(0, 1, 0, 0)^T$
$(2, 2, 200000, 0)$	$(0, 1, 0, 0)^T$	$(2, 2, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(2, 2, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(2, 2, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(2, 2, 400000, 0)$	$(0, 1, 0, 0)^T$	$(2, 2, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(2, 2, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(2, 2, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(2, 2, 600000, 0)$	$(0, 1, 0, 0)^T$	$(2, 2, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(2, 2, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(2, 2, 600000, 600000)$	$(0, 1, 0, 0)^T$
$(2, 3, 0, 0)$	$(0, 0, 1, 0)^T$	$(2, 3, 0, 200000)$	$(0, 0, 1, 0)^T$
$(2, 3, 0, 400000)$	$(0, 0, 1, 0)^T$	$(2, 3, 0, 600000)$	$(0, 0, 1, 0)^T$
$(2, 3, 200000, 0)$	$(0, 0, 1, 0)^T$	$(2, 3, 200000, 200000)$	$(0, 0, 1, 0)^T$
$(2, 3, 200000, 400000)$	$(0, 0, 1, 0)^T$	$(2, 3, 200000, 600000)$	$(0, 0, 1, 0)^T$
$(2, 3, 400000, 0)$	$(0, 0, 1, 0)^T$	$(2, 3, 400000, 200000)$	$(0, 0, 1, 0)^T$
$(2, 3, 400000, 400000)$	$(0, 0, 1, 0)^T$	$(2, 3, 400000, 600000)$	$(0, 0, 1, 0)^T$
$(2, 3, 600000, 0)$	$(0, 0, 1, 0)^T$	$(2, 3, 600000, 200000)$	$(0, 0, 1, 0)^T$
$(2, 3, 600000, 400000)$	$(0, 0, 1, 0)^T$	$(2, 3, 600000, 600000)$	$(0, 0, 1, 0)^T$

Table B.8: Firm 2's function h_2, continued

(a_1, a_2, q_1, q_2)	$h_2(a_1, a_2, q_1, q_2)$	(a_1, a_2, q_1, q_2)	$h_2(a_1, a_2, q_1, q_2)$
$(3, 0, 0, 0)$	$(0, 1, 0, 0)^T$	$(3, 0, 0, 200000)$	$(0, 1, 0, 0)^T$
$(3, 0, 0, 400000)$	$(0, 1, 0, 0)^T$	$(3, 0, 0, 600000)$	$(0, 1, 0, 0)^T$
$(3, 0, 200000, 0)$	$(0, 1, 0, 0)^T$	$(3, 0, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(3, 0, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(3, 0, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(3, 0, 400000, 0)$	$(0, 1, 0, 0)^T$	$(3, 0, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(3, 0, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(3, 0, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(3, 0, 600000, 0)$	$(0, 1, 0, 0)^T$	$(3, 0, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(3, 0, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(3, 0, 600000, 600000)$	$(0, 1, 0, 0)^T$
$(3, 1, 0, 0)$	$(0, 1, 0, 0)^T$	$(3, 1, 0, 200000)$	$(0, 1, 0, 0)^T$
$(3, 1, 0, 400000)$	$(0, 1, 0, 0)^T$	$(3, 1, 0, 600000)$	$(0, 1, 0, 0)^T$
$(3, 1, 200000, 0)$	$(0, 1, 0, 0)^T$	$(3, 1, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(3, 1, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(3, 1, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(3, 1, 400000, 0)$	$(0, 1, 0, 0)^T$	$(3, 1, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(3, 1, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(3, 1, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(3, 1, 600000, 0)$	$(0, 1, 0, 0)^T$	$(3, 1, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(3, 1, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(3, 1, 600000, 600000)$	$(0, 1, 0, 0)^T$
$(3, 2, 0, 0)$	$(0, 1, 0, 0)^T$	$(3, 2, 0, 200000)$	$(0, 0, 1, 0)^T$
$(3, 2, 0, 400000)$	$(0, 0, 1, 0)^T$	$(3, 2, 0, 600000)$	$(0, 0, 1, 0)^T$
$(3, 2, 200000, 0)$	$(0, 1, 0, 0)^T$	$(3, 2, 200000, 200000)$	$(0, 1, 0, 0)^T$
$(3, 2, 200000, 400000)$	$(0, 1, 0, 0)^T$	$(3, 2, 200000, 600000)$	$(0, 1, 0, 0)^T$
$(3, 2, 400000, 0)$	$(0, 1, 0, 0)^T$	$(3, 2, 400000, 200000)$	$(0, 1, 0, 0)^T$
$(3, 2, 400000, 400000)$	$(0, 1, 0, 0)^T$	$(3, 2, 400000, 600000)$	$(0, 1, 0, 0)^T$
$(3, 2, 600000, 0)$	$(0, 1, 0, 0)^T$	$(3, 2, 600000, 200000)$	$(0, 1, 0, 0)^T$
$(3, 2, 600000, 400000)$	$(0, 1, 0, 0)^T$	$(3, 2, 600000, 600000)$	$(0, 1, 0, 0)^T$
$(3, 3, 0, 0)$	$(0, 0, 1, 0)^T$	$(3, 3, 0, 200000)$	$(0, 0, 0, 1)^T$
$(3, 3, 0, 400000)$	$(0, 0, 0, 1)^T$	$(3, 3, 0, 600000)$	$(0, 0, 0, 1)^T$
$(3, 3, 200000, 0)$	$(0, 0, 1, 0)^T$	$(3, 3, 200000, 200000)$	$(0, 0, 1, 0)^T$
$(3, 3, 200000, 400000)$	$(0, 0, 1, 0)^T$	$(3, 3, 200000, 600000)$	$(0, 0, 1, 0)^T$
$(3, 3, 400000, 0)$	$(0, 0, 1, 0)^T$	$(3, 3, 400000, 200000)$	$(0, 0, 1, 0)^T$
$(3, 3, 400000, 400000)$	$(0, 0, 1, 0)^T$	$(3, 3, 400000, 600000)$	$(0, 0, 1, 0)^T$
$(3, 3, 600000, 0)$	$(0, 0, 1, 0)^T$	$(3, 3, 600000, 200000)$	$(0, 0, 1, 0)^T$
$(3, 3, 600000, 400000)$	$(0, 0, 1, 0)^T$	$(3, 3, 600000, 600000)$	$(0, 0, 1, 0)^T$

Table B.9: Firm 2's function h_2, continued and concluded

Appendix C

MATLAB scripts for the monopoly model

The script requires four different structures in order to solve the model. These are the vectors *tspan* and *parameters* and the structure arrays *trend* and *latticeInfo*.

The vector *tspan* is a row vector with two elements. The elements describe the starting and the ending points of the time interval corresponding to the monopoly's planning horizon. In this thesis the vector *tspan* then is either $(0, 10)$ corresponding to the time interval $[0, 10]$ or $(0, 100)$ corresponding to the time interval $[0, 100]$

The vector *parameters* is a row vector with seven elements. These elements correspond to the parameter choices described in chapter 3. Table C.1 describes the composition of vector *parameters*.

Element in *parameters*	Parameter name
$parameters_1$	N
$parameters_2$	μ
$parameters_3$	r
$parameters_4$	β
$parameters_5$	λ
$parameters_6$	c_1
$parameters_7$	c_2

Table C.1: Composition of vector *parameters*

The structure array *trend* describes the trend parameters. The double *trend.d* is the strength of the trend, i.e. the parameter d. The double *trend.type* determines the type of the trend, with *trend.type* $= 1$ standing for a linear trend and *trend.type* $= 2$ standing for a parabel trend. Finally, the double *trend.c* is the saturation point of the parabel trend, i.e. the parameter m.

The stucture array *latticeInfo* describes the choice sets A, Q and P (the sets (3.8)). The script generates each set from two values, the count and the interval values. As there are three different sets in total, there are six different values in total in *latticeInfo*: the values *latticeInfo.QualityInterval* and *latticeInfo.QualityCount* describing the quality set A, the values *latticeInfo.SamplesInterval* and *latticeInfo.SamplesCount* describing the free samples set Q, and finally the values *latticeInfo.PriceInterval* and *latticeInfo.PriceCount* describing the price set P. The script assumes that the 0 will always be a member of each choice set and will generate each sets starting from it. The count value gives the number of elements in each set. The interval value gives the interval between the sequential members of the set.

For example, let us consider how to generate the quality set $A = \{0, 4, 8, 12, 16, 20\}$. First, we note that the interval value for the set is $4 = 4 - 0 = 8 - 4 = \ldots = 20 - 16$, and therefore we set *latticeInfo.QualityInterval* $= 4$. Next, we note that there are six members in the set we wish to generate, and therefore we set *latticeInfo.QualityCount* $=$ 6. Generating the free samples and price sets follows the same logic.

C.1 Example runfile for the monopoly model

The following runfile uses the scripts to solve the monopoly model with parameter choices described in chapter 3 and choices $r = 0$ and $T = 10$.

```
% all of the script files should be in the same directory

% defining the path for the script files

functionPath = cd();
addpath(functionPath);

tspan = [0 10];

parameters = [1000000 0.5 0.00 1 0.5 0.5 1];

latticeInfo.PriceInterval = 3;
latticeInfo.PriceCount = 4;

latticeInfo.SamplesInterval = 200000;
latticeInfo.SamplesCount = 4;

latticeInfo.QualityInterval = 1;
latticeInfo.QualityCount = 4;

trend.type = 1;
trend.d = 0;

% unnecessary for the no trend case, add for sake of defining all
% variables
trend.c = 1*parameters(1)/3;

[firmChoices maxprofits] = MonopolyOptimizes(latticeInfo, tspan, ...
      parameters, trend);
```

The script generates a structural array *firmChoices* and a double *maxprofits*. The structural array *firmChoices* contains the optimal behavior of the monopoly, and the double *maxprofits* the optimal profits of the monopoly.

C.2 MonopolyOptimizes.m

```
%%%%%%%%%%%%%%%%%%%%%% ubnofqczsqgjpejlcqsidv %%%%%%%%%%%%%%%%%%%%%%%%%%%
% If the reader actually wishes to play around with the scripts, please %
% send a message to ville.ka.makinen@gmail.com for the script files     %
%%%%%%%%%%%%%%%%%%%% ijkwtsczyicrtemmgntuzzpfkspkt  %%%%%%%%%%%%%%%%%%%%%%

function [MonopolyOptimalChoices Profits] = MonopolyOptimizes ...
                                    (latticeInfo, tspan, parameters, trend)

    MonopolyOptimalChoices.Quality  = 0;
    MonopolyOptimalChoices.Samples = 0;
    MonopolyOptimalChoices.Price = 0;

    Profits = -10^7;

    for i = 1:latticeInfo.QualityCount
        for j = 1:latticeInfo.SamplesCount
            for k = 1:latticeInfo.PriceCount

                firmChoices.Quality = (i-1)*latticeInfo.QualityInterval;
                firmChoices.Samples = (j-1)*latticeInfo.SamplesInterval;
                firmChoices.Price = (k-1)*latticeInfo.PriceInterval;

                CalculatedMonopolyProfits = MonopolyProfits(tspan, ...
                                        parameters, trend, firmChoices);

                if CalculatedMonopolyProfits > Profits
                    Profits = CalculatedMonopolyProfits;

                    MonopolyOptimalChoices.Quality = firmChoices.Quality;
                    MonopolyOptimalChoices.Samples = firmChoices.Samples;
                    MonopolyOptimalChoices.Price  = firmChoices.Price;

                end
            end
        end
    end
end
```

C.3 MonopolyProfits.m

```
function retVal = MonopolyProfits(tspan, parameters, trend, firmChoices)

    t = linspace(tspan(1), tspan(2),tspan(2)*100);
    revenue = zeros(1,length(t));
```

```
    initialState = firmChoices.Samples;

    ratk = ode45(@monopolyDE,tspan,initialState,[], parameters, ...
                trend, firmChoices);

    for i = 1:length(t)
        revenue(i) = MonopolyRevenue(t(i), ratk, parameters, ...
                        trend, firmChoices);
    end

    netrevenue = trapz(revenue)*(t(2)-t(1));

    retVal = netrevenue - firmChoices.Samples*parameters(6)*...
                            (firmChoices.Quality^2+parameters(7));
end
```

C.4 monopolyDE.m

```
function retVal = monopolyDE(t, x, parameters,trend,firmChoices)

    N = parameters(1);
    mu = parameters(2);
    beta = parameters(4);
    lambda = parameters(5);


    a = firmChoices.Quality;
    p = firmChoices.Price;

    if trend.type == 1 % linear trend
        effectOfTrend = trend.d*a*x/N;
    elseif trend.type == 2 % parabeloid trend
        A = -trend.d/(trend.c^2);
        B = 2*trend.d/trend.c;

        effectOfTrend = a*(A*x^2 + B*x);
    end

    retVal = beta*(1+exp(-(a-p+effectOfTrend)*mu^-1))^-1*(N-x)-lambda*x;
end
```

C.5 MonopolyRevenue.m

```
function retVal = MonopolyRevenue(t,ratk, parameters, trend, firmChoices)
```

```
    x = deval(ratk,t);

    N = parameters(1);
    mu = parameters(2);
    beta = parameters(4);

    r = parameters(3);

    c1 = parameters(6);
    c2 = parameters(7);


    a = firmChoices.Quality;
    p = firmChoices.Price;

    if trend.type == 1 % linear trend
        effectOfTrend = trend.d*a*x/N;

    elseif trend.type == 2 % parabeloid trend
        A = -trend.d/(trend.c^2);
        B = 2*trend.d/trend.c;

        effectOfTrend = a*(A*x^2 + B*x);
    end

    retVal = exp(-r*t)*(p-c1*(a^2+c2))*beta*...
                (1/(1+exp(-(a-p+effectOfTrend)*(1/mu))))*(N-x);
end
```

Appendix D

MATLAB scripts for the Stackelberg duopoly model

The script requires five different structures in order to solve the model. These are the vectors *monopolytspan*, *duopolytspan* and *parameters* and the structure arrays *trend* and *latticeInfo*.

The vectors *monopolytspan* and *duopolytspan* are row vectors with two elements. The elements describe the starting and the ending points of the time intervals in the model. First, the vector *monopolytspan* describes the time interval when the Stackelberg leader operates alone in the market. In this thesis the vector *monopolytspan* then is $(0, 1)$ corresponding to the time interval $[0, 1]$. Next, the vector *duopolytspan* describes the time interval when the Stackelberg leader and the Stackelberg follower operate together in the market. The time interval corresponds to the planning horizon of both firms. In this thesis the vector *duopolytspan* then is either $(0, 10)$ corresponding to the duopoly time interval $[0, 10]$ or $(0, 100)$ corresponding to the duopoly time interval $[0, 100]$

The vector *parameters* is a row vector with seven elements. These elements correspond to the parameter choices described in chapter 3. Table D.1 describes the composition of vector *parameters*.

Element in *parameters*	Parameter name
parameters$_1$	N
parameters$_2$	μ
parameters$_3$	r
parameters$_4$	β
parameters$_5$	λ
parameters$_6$	c_1
parameters$_7$	c_2

Table D.1: Composition of vector *parameters*

The structure array *trend* describes the trend parameters. The double *trend.d* is the strength of the trend, i.e. the parameter d. The double *trend.type* determines the type of the trend, with *trend.type* $= 1$ standing for a linear trend and *trend.type* $= 2$ standing for a parabel trend. Finally, the double *trend.c* is the saturation point of the parabel trend, i.e. the parameter m.

The stucture array *latticeInfo* describes the choice sets A, Q and P (the sets (3.8)). The script generates each set from two values, the count and the interval values. As there are three different sets in total, there are six different values in total in *latticeInfo*: the values *latticeInfo.QualityInterval* and *latticeInfo.QualityCount* describing the quality set A, the values *latticeInfo.SamplesInterval* and *latticeInfo.SamplesCount* describing the free samples set Q, and finally the values *latticeInfo.PriceInterval* and *latticeInfo.PriceCount* describing the price set P. The script assumes that the 0 will always be a member of each choice set and will generate each sets starting from it. The count value gives the number of elements in each set. The interval value gives the interval between the sequential members of the set.

For example, let us consider how to generate quality set $\{0, 4, 8, 12, 16, 20\}$. First, we note that the interval value for the set is $4 = 4 - 0 = 8 - 4 = \ldots = 20 - 16$, and therefore we set *latticeInfo.QualityInterval* $= 4$. Next, we note that there are six members in the set we wish to generate, and therefore we set *latticeInfo.QualityCount* $= 6$. Generating the free samples and price sets follows the same logic.

D.1 Example runfile for the Stackelberg duopoly model

The following runfile uses the scripts to solve the Stackelberg duopoly model with parameter choices described in chapter 3 and choices $r = 0$ and $T = 10$.

```
% all of the script files should be in the same directory

% defining the path for the script files

functionPath = cd();
addpath(functionPath);


monopolyTimeLength = 1;

duopolytspan = [0 10];
monopolytspan = [0 monopolyTimeLength];

parameters = [1000000 0.5 0.00 1 0.5 0.5 1];

latticeInfo.PriceInterval = 3;
latticeInfo.PriceCount = 4;

latticeInfo.SamplesInterval = 200000;
latticeInfo.SamplesCount = 4;

latticeInfo.QualityInterval = 1;
latticeInfo.QualityCount = 4;

trend.type = 1;
trend.d = 0;

% unnecessary for the no trend case, add for sake of defining all
% variables
```

```
trend.c = 1*parameters(1)/3;

[LeaderOptimalChoices FollowerOptimalChoices LeaderOptimalProfits ...
FollowerOptimalProfits] = LeaderOptimizes(latticeInfo, monopolytspan, ...
duopolytspan, parameters, trend);
```

This script generates two structural arrays and two doubles. The two structure arrays *LeaderOptimalChoices* and *FollowerOptimalChoices* contain the optimal behavior of both firms. The two doubles *LeaderOptimalProfits* and *FollowerOptimalProfits* contain the optimal profits determined by firm behavior.

D.2 LeaderOptimizes.m

```
%%%%%%%%%%%%%%%%%%%%%% ubnofqczsqgjpejlcqsidv %%%%%%%%%%%%%%%%%%%%%%%%%%%%
% If the reader actually wishes to play around with the scripts, please %
% send a message to ville.ka.makinen@gmail.com for the script files     %
%%%%%%%%%%%%%%%%%%% ijkwtsczyicrtemmgntuzzpfkspkt  %%%%%%%%%%%%%%%%%%%%%%

function [LeaderOptimalChoices FollowerOptimalChoices ...
            LeaderOptimalProfits FollowerOptimalProfits] = ...
            LeaderOptimizes(latticeInfo, monopolytspan, duopolytspan, ...
            parameters, trend)

    FollowerOptimalChoices.Quality  = 0;
    FollowerOptimalChoices.Samples = 0;
    FollowerOptimalChoices.Price = 0;

    FollowerOptimalProfits = -10^7;

    % optimal choices defined and formatted
    LeaderOptimalChoices = FollowerOptimalChoices;
    LeaderOptimalProfits = FollowerOptimalProfits;

    for i = 1:latticeInfo.QualityCount
        for j = 1:latticeInfo.SamplesCount
            for k = 1:latticeInfo.PriceCount
                LeaderChoices.Quality = (i-1)*latticeInfo.QualityInterval;
                LeaderChoices.Samples = (j-1)*latticeInfo.SamplesInterval;
                LeaderChoices.Price = (k-1)*latticeInfo.PriceInterval;

                % market evolves as it was a monopoly

                [netLeaderMonopolyRevenue AmountOfLeadersProductsInMarket] = ...
                    preentryProfitsLeader(latticeInfo, monopolytspan, ...
                    duopolytspan, parameters, trend, LeaderChoices);
```

```
                % The Follower enters the market and optimizes its behavior
                % w.r.t. the market situation/the Leaders choices

                [FollowerChoices FollowerProfits] = ...
                        FollowerOptimizes(latticeInfo, monopolytspan, ...
                        duopolytspan, parameters, trend, LeaderChoices, ...
                        AmountOfLeadersProductsInMarket);

                % FollowerChoices now gives the follower choices in the
                % duopoly stage, thus enabling us to calculate the leader’s
                % profits

                netLeaderDuopolyProfits = profitsLeader(duopolytspan, ...
                    parameters, trend, LeaderChoices, FollowerChoices, ...
                    AmountOfLeadersProductsInMarket, monopolytspan);

                CalculatedLeaderProfits = netLeaderDuopolyProfits + ...
                    netLeaderMonopolyRevenue - ...
                    LeaderChoices.Samples.*parameters(6).*...
                    (LeaderChoices.Quality.^2+parameters(7));

                % LeaderOptimalChoices checked and updated if approriate

                if CalculatedLeaderProfits > LeaderOptimalProfits
                    LeaderOptimalProfits = CalculatedLeaderProfits;
                    LeaderOptimalChoices = LeaderChoices;
                    FollowerOptimalChoices = FollowerChoices;
                    FollowerOptimalProfits = FollowerProfits;
                end
            end
        end
    end
end
```

D.3 preentryProfitsLeader.m

```
function [netLeaderMonopolyRevenue AmountOfLeadersProductsInMarket] ...
= preentryProfitsLeader(latticeInfo, monopolytspan, ...
 duopolytspan, parameters, trend, LeaderChoices)

        MonopolyT = linspace(monopolytspan(1), monopolytspan(2), ...
monopolytspan(2)*100);
        LeaderMonopolyRevenue = zeros(1,length(MonopolyT));

        MonopolyInitialState = LeaderChoices.Samples;
```

```
        ratk = ode45(@monopolyDE,monopolytspan, ...
                        MonopolyInitialState,[], parameters, ...
                        trend, LeaderChoices);

        for l = 1:length(MonopolyT)
        LeaderMonopolyRevenue(l) = MonopolyRevenue(MonopolyT(l), ...
ratk, parameters, trend, LeaderChoices);
        end

        netLeaderMonopolyRevenue = trapz(LeaderMonopolyRevenue)*...
                                    (MonopolyT(2)-MonopolyT(1));

        AmountOfLeadersProductsInMarket = deval(ratk, monopolytspan(2));

end
```

D.4 monopolyDE.m

```
function retVal = monopolyDE(t, x, parameters,trend,firmChoices)

    N = parameters(1);
    mu = parameters(2);
    beta = parameters(4);
    lambda = parameters(5);


    a = firmChoices.Quality;
    p = firmChoices.Price;

    if trend.type == 1 % linear trend
        effectOfTrend = trend.d*a*x/N;

    elseif trend.type == 2 % parabeloid trend
        A = -trend.d/(trend.c^2);
        B = 2*trend.d/trend.c;

        effectOfTrend = a*(A*x^2 + B*x);
    end

    retVal = beta*(1+exp(-(a-p+effectOfTrend)*mu^-1))^-1*(N-x)-lambda*x;

end
```

D.5 MonopolyRevenue.m

```
function retVal = MonopolyRevenue(t,ratk, parameters, trend, firmChoices)

    x = deval(ratk,t);

    N = parameters(1);
    mu = parameters(2);
    beta = parameters(4);

    r = parameters(3);

    c1 = parameters(6);
    c2 = parameters(7);


    a = firmChoices.Quality;
    p = firmChoices.Price;

    if trend.type == 1 % linear trend
        effectOfTrend = trend.d*a*x/N;

    elseif trend.type == 2 % parabeloid trend
        A = -trend.d/(trend.c^2);
        B = 2*trend.d/trend.c;

        effectOfTrend = a*(A*x^2 + B*x);
    end

    retVal = exp(-r*t)*(p-c1*(a^2+c2))*beta*...
        (1/(1+exp(-(a-p+effectOfTrend)*(1/mu))))*(N-x);


end
```

D.6 FollowerOptimizes.m

```
% By assumption the Leader may not alter its choices during the market
% evolution.
% The Follower can then optimize its own profits w.r.t. to the Leaders
% choices

function [FollowerOptimalChoices FollowerOptimalProfits] = ...
            FollowerOptimizes(latticeInfo, monopolytspan, duopolytspan, ...
            parameters, trend, LeaderChoices, ...
            AmountOfLeadersProductsInMarket)
```

```
    FollowerOptimalChoices.Quality = 0;
    FollowerOptimalChoices.Samples = 0;
    FollowerOptimalChoices.Price = 0;

    FollowerOptimalProfits = -10^7;

    for i = 1:latticeInfo.QualityCount
        for j = 1:latticeInfo.SamplesCount
            for k = 1:latticeInfo.PriceCount

                FollowerChoices.Quality = (i-1)*latticeInfo.QualityInterval;
                FollowerChoices.Samples = (j-1)*latticeInfo.SamplesInterval;
                FollowerChoices.Price = (k-1)*latticeInfo.PriceInterval;

                CalculatedFollowerProfits = profitsFollower(duopolytspan,...
                    parameters, trend, LeaderChoices, FollowerChoices, ...
                    AmountOfLeadersProductsInMarket);

                if CalculatedFollowerProfits > FollowerOptimalProfits
                    FollowerOptimalChoices = FollowerChoices;
                    FollowerOptimalProfits = CalculatedFollowerProfits;
                end

            end
        end
    end
end
```

D.7 profitsFollower.m

```
function retVal = profitsFollower(tspan, parameters, trend, ...
        LeaderStrategy, FollowerStrategy, AmountOfLeadersProductsInMarket)

    t = linspace(tspan(1),tspan(2),tspan(2)*100);
    revenue = zeros(1,length(t));

    N  = parameters(1);

    initialState = [0,0,0];

    % each consumer has the same probability to get a free sample, and the
    % probabilities are independent between firms

    initialState(1) = (1-FollowerStrategy.Samples/N)*...
                            AmountOfLeadersProductsInMarket;
```

```
    initialState(2) = (FollowerStrategy.Samples/N)*...
                            (N-AmountOfLeadersProductsInMarket);
    initialState(3) = (FollowerStrategy.Samples/N)*...
                            AmountOfLeadersProductsInMarket;

    ratk = ode45(@duopolyDES,tspan,initialState,[],parameters,trend,...
                    LeaderStrategy,FollowerStrategy);

    for i = 1:length(t)
        revenue(i) = revenueFollower(t(i),ratk, parameters,trend,...
            LeaderStrategy, FollowerStrategy);
    end

    netrevenue = trapz(revenue)*(t(2)-t(1));

    retVal = netrevenue - FollowerStrategy.Samples.*parameters(6).*...
        (FollowerStrategy.Quality.^2+parameters(7));
end
```

D.8 duopolyDES.m

```
function retval = duopolyDES(t,x, parameters,trend, firm1strategy, ...
                    firm2strategy)

    N = parameters(1);
    mu = parameters(2);
    beta = parameters(4);
    lambda = parameters(5);

    a1 = firm1strategy.Quality;
    p1 = firm1strategy.Price;

    a2 = firm2strategy.Quality;
    p2 = firm2strategy.Price;

    q1 = x(1)+x(3); q2 = x(2)+x(3);

    if trend.type == 1 % linear trend
        effectOfTrendProduct1 = trend.d*a1*q1/N;
        effectOfTrendProduct2 = trend.d*a2*q2/N;
    elseif trend.type == 2 % parabeloid trend
        A = -trend.d/(trend.c^2);
        B = 2*trend.d/trend.c;

        effectOfTrendProduct1 = a1*(A*q1^2 + B*q1);
        effectOfTrendProduct2 = a2*(A*q2^2 + B*q2);
```

```
    end

    denomS = 1 + exp((1/mu)*(a1-p1+effectOfTrendProduct1))+ ...
                exp((1/mu)*(a2-p2+effectOfTrendProduct2));
    denomI1 = 1 + exp((1/mu)*(max(a2-a1,0)-p2+effectOfTrendProduct2));
    denomI2 = 1 + exp((1/mu)*(max(a1-a2,0)-p1+effectOfTrendProduct1));

    PSI1 = exp((1/mu)*(a1-p1+effectOfTrendProduct1))/denomS;
    PSI2 = exp((1/mu)*(a2-p2+effectOfTrendProduct2))/denomS;
    PI1I12 = exp((1/mu)*(max(a2-a1,0)-p2+effectOfTrendProduct2))/denomI1;
    PI2I12 = exp((1/mu)*(max(a1-a2,0)-p1+effectOfTrendProduct1))/denomI2;

    I1prime = beta*PSI1*(N-x(1)-x(2)-x(3))-beta*PI1I12*x(1)+ ...
                lambda*x(3)-lambda*x(1);
    I2prime = beta*PSI2*(N-x(1)-x(2)-x(3))-beta*PI2I12*x(2)+ ...
                lambda*x(3)-lambda*x(2);

    I12prime = beta*PI1I12*x(1)+beta*PI2I12*x(2)-2*lambda*x(3);

    retval = [ I1prime; I2prime; I12prime];
end
```

D.9 revenueFollower.m

```
function retVal = revenueFollower(t,ratk,parameters,trend,firm1strategy,...
                    firm2strategy)

    % t refers to time

    state = deval(ratk,t);

    N = parameters(1);
    mu = parameters(2);
    r = parameters(3);
    beta = parameters(4);

    c1 = parameters(6);
    c2 = parameters(7);

    a1 = firm1strategy.Quality;
    p1 = firm1strategy.Price;

    a2 = firm2strategy.Quality;
    p2 = firm2strategy.Price;

    q1 = state(1)+state(3); q2 = state(2)+state(3);
```

```
    if trend.type == 1 % linear trend
        effectOfTrendProduct1 = trend.d*a1*q1/N;
        effectOfTrendProduct2 = trend.d*a2*q2/N;
    elseif trend.type == 2 % parabeloid trend
        A = -trend.d/(trend.c^2);
        B = 2*trend.d/trend.c;

        effectOfTrendProduct1 = a1*(A*q1^2 + B*q1);
        effectOfTrendProduct2 = a2*(A*q2^2 + B*q2);
    end
    % denomI2, PSI1, PI2I12 commented as unnecessary for firm 1 revenue

    denomS = 1 + exp((1/mu)*(a1-p1+effectOfTrendProduct1))+exp((1/mu)*...
                (a2-p2+effectOfTrendProduct2));
    denomI1 = 1 + exp((1/mu)*(max(a2-a1,0)-p2+effectOfTrendProduct2));
    %denomI2 = 1 + exp((1/mu)*(max(a1-a2,0)-p1+effectOfTrendProduct1));

    %PSI1 = exp((1/mu)*(a1-p1+effectOfTrendProduct1))/denomS;
    PSI2 = exp((1/mu)*(a2-p2+effectOfTrendProduct2))/denomS;
    PI1I12 = exp((1/mu)*(max(a2-a1,0)-p2+effectOfTrendProduct2))/denomI1;
    %PI2I12 = exp((1/mu)*(max(a1-a2,0)-p1+effectOfTrendProduct1))/denomI2;

    salesSI2 = beta*PSI2*(N-state(1)-state(2)-state(3));
    salesI1I12 = beta*PI1I12*state(1);

    retVal = exp(-r.*t)*(p2-c1.*(a2.^2+c2))*(salesSI2+salesI1I12);
end
```

D.10 profitsLeader.m

```
function retVal = profitsLeader(duopolytspan, parameters, trend, ...
    LeaderChoices, FollowerChoices,AmountOfLeadersProductsInMarket,...
    monopolytspan)

    t = linspace(duopolytspan(1),duopolytspan(2),duopolytspan(2)*100);
    revenue = zeros(1,length(t));

    N  = parameters(1);

    initialState = [0,0,0];

    initialState(1) = (1-FollowerChoices.Samples/N)*...
                        AmountOfLeadersProductsInMarket;
    initialState(2) = (FollowerChoices.Samples/N)*...
                        (N-AmountOfLeadersProductsInMarket);
```

```
    initialState(3) = (FollowerChoices.Samples/N)*...
                        AmountOfLeadersProductsInMarket;

    ratk = ode45(@duopolyDES,duopolytspan,initialState,[],parameters,...
                trend,LeaderChoices,FollowerChoices);

    for i = 1:length(t)
        revenue(i) = revenueLeader(t(i),ratk, parameters,trend,...
                        LeaderChoices,FollowerChoices, monopolytspan);
    end

    netrevenue = trapz(revenue)*(t(2)-t(1));

    retVal = netrevenue;
end
```

D.11 revenueLeader.m

```
function retVal = revenueLeader(t,ratk,parameters,trend,firm1strategy, ...
                    firm2strategy, monopolytspan)

    % t refers to time

    state = deval(ratk,t);

    N = parameters(1);
    mu = parameters(2);
    r = parameters(3);
    beta = parameters(4);

    c1 = parameters(6);
    c2 = parameters(7);

    a1 = firm1strategy.Quality;
    p1 = firm1strategy.Price;

    a2 = firm2strategy.Quality;
    p2 = firm2strategy.Price;

    q1 = state(1)+state(3); q2 = state(2)+state(3);

    if trend.type == 1 % linear trend
        effectOfTrendProduct1 = trend.d*a1*q1/N;
        effectOfTrendProduct2 = trend.d*a2*q2/N;
    elseif trend.type == 2 % parabeloid trend
        A = -trend.d/(trend.c^2);
```

```
        B = 2*trend.d/trend.c;

        effectOfTrendProduct1 = a1*(A*q1^2 + B*q1);
        effectOfTrendProduct2 = a2*(A*q2^2 + B*q2);
    end

    % denomI1, PSI2, PI1I12 commented as unnecessary for firm 1 revenue

    denomS = 1 + exp((1/mu)*(a1-p1+effectOfTrendProduct1))+ ...
                exp((1/mu)*(a2-p2+effectOfTrendProduct2));
    %denomI1 = 1 + exp((1/mu)*(max(a2-a1,0)-p2+effectOfTrendProduct2));
    denomI2 = 1 + exp((1/mu)*(max(a1-a2,0)-p1+effectOfTrendProduct1));

    PSI1 = exp((1/mu)*(a1-p1+effectOfTrendProduct1))/denomS;
    %PSI2 = exp((1/mu)*(a2-p2+effectOfTrendProduct2))/denomS;
    %PI1I12 = exp((1/mu)*(max(a2-a1,0)-p2+effectOfTrendProduct2))/denomI1;
    PI2I12 = exp((1/mu)*(max(a1-a2,0)-p1+effectOfTrendProduct1))/denomI2;

    salesSI1 = beta*PSI1*(N-state(1)-state(2)-state(3));
    salesI2I12 = beta*PI2I12*state(2);

    % The leader discounts to an earlier point of time than the follower
    % t -> t + monopolytspan(2)

    retVal = exp(-r.*(t+monopolytspan(2)))*(p1-c1.*(a1.^2+c2))*...
                (salesSI1+salesI2I12);
end
```

Appendix E

MATLAB scripts for the Simultaneous-choice duopoly model

The script requires four different structures in order to solve the model. These are the vectors *tspan* and *parameters* and the structure arrays *trend* and *latticeInfo*.

The vector *tspan* is a row vector with two elements. The elements describe the starting and the ending points of the time interval corresponding to the planning horizons of both firms. In this thesis the vector *tspan* then is either $(0, 10)$ corresponding to the time interval $[0, 10]$ or $(0, 100)$ corresponding to the time interval $[0, 100]$

The vector *parameters* is a row vector with seven elements. These elements correspond to the parameter choices described in chapter 3. Table E.1 describes the composition of vector *parameters*.

Element in *parameters*	Parameter name
$parameters_1$	N
$parameters_2$	μ
$parameters_3$	r
$parameters_4$	β
$parameters_5$	λ
$parameters_6$	c_1
$parameters_7$	c_2

Table E.1: Composition of vector *parameters*

The structure array *trend* describes the trend parameters. The double *trend.d* is the strength of the trend, i.e. the parameter d. The double *trend.type* determines the type of the trend, with *trend.type* $= 1$ standing for a linear trend and *trend.type* $= 2$ standing for a parabel trend. Finally, the double *trend.c* is the saturation point of the parabel trend, i.e. the parameter m.

The stucture array *latticeInfo* describes the choice sets A, Q and P (the sets (3.8)). The script generates each set from two values, the count and the interval values. As there are three different sets in total, there are six different values in total in *latticeInfo*: the values *latticeInfo.QualityInterval* and *latticeInfo.QualityCount* describing the quality set A, the values *latticeInfo.SamplesInterval* and *latticeInfo.SamplesCount* describing the free samples set Q, and finally the values *latticeInfo.PriceInterval* and *latticeInfo.PriceCount* describing the price set P. The script assumes that the 0 will always be a member of each choice set and will generate each sets starting from it. Differently from the previous monopoly and Stackelberg duopoly scripts, the count value gives the number of non-zero elements in each set, i.e. the number of elements in each set minus one. The interval value gives the interval between the sequential members of the set.

For example, let us consider how to generate quality set $\{0, 4, 8, 12, 16, 20\}$. First, we note that the interval value for the set is $4 = 4 - 0 = 8 - 4 = \ldots = 20 - 16$, and therefore we set *latticeInfo.QualityInterval* $= 4$. Next, we note that there are five non-zero members in the set we wish to generate, and therefore we set the count value as *latticeInfo.QualityCount* $= 5$. Generating the free samples and price sets follows the same logic.

E.1 Example runfile for the Simultaneous-choice duopoly model

The following runfile uses the scripts to solve the Simultaneous-choice duopoly model with parameter choices described in chapter 3 and choices $r = 0$ and $T = 10$.

```
% all of the script files should be in the same directory

% defining the path for the script files

functionPath = cd();
addpath(functionPath);

% defining how many cores to use for parallel calculation

matlabpool open 4

latticeInfo.PriceInterval = 3;
latticeInfo.PriceCount = 3;

latticeInfo.SamplesInterval = 200000;
latticeInfo.SamplesCount = 3;

latticeInfo.QualityInterval = 1;
latticeInfo.QualityCount = 3;

tspan = [0 10];

parameters = [1000000 0.5 0.00 1 0.5 0.5 1];

trend.type = 1;
trend.d = 0;

% unnecessary for the no trend case, add for sake of defining all
% variables
trend.c = 1*parameters(1)/3;

FirstStageNEQualities(latticeInfo, tspan, parameters,trend)
```

This script generates new folders that contain the profits and the subgame-perfect Nash equilibrium strategies for both firms.

E.2 FirstStageNEQualities.m

```
%%%%%%%%%%%%%%%%%%%%%% ubnofqczsqgjpejlcqsidv %%%%%%%%%%%%%%%%%%%%%%%%%%%
% If the reader actually wishes to play around with the scripts, please %
% send a message to ville.ka.makinen@gmail.com for the script files     %
%%%%%%%%%%%%%%%%%%%%% ijkwtsczyicrtemmgntuzzpfkspkt  %%%%%%%%%%%%%%%%%%%%%%

function retVal = FirstStageNEQualities(latticeInfo, tspan, parameters, ...
                    trend)

    iterationDirectoryName = './FirstStage/';

    mkdir(iterationDirectoryName)
    cd(iterationDirectoryName)

    matDim = latticeInfo.QualityCount+1;

    ProfitMatrixFirm1 = zeros(matDim,matDim);
    ProfitMatrixFirm2 = zeros(matDim,matDim);

    qualityInterval = latticeInfo.QualityInterval;

    for i = 1:matDim % rows
        for j = 1:matDim % columns

            firm1strategy.Quality = qualityInterval*(i-1);
            firm2strategy.Quality = qualityInterval*(j-1);

            SampleGameNEs = SecondStageNESamples(latticeInfo, ...
                    firm1strategy, firm2strategy, tspan, parameters,trend);

            % It is possible that there are multiple equilibria in the
            % third stage price choosing game -> necessary to choose
            % between those

            % I arbitrarily choose the first NE that the algorithm finds


            ProfitMatrixFirm1(i,j) = SampleGameNEs(1).rowPlayerUtility;
            ProfitMatrixFirm2(i,j) = SampleGameNEs(1).columnPlayerUtility;
```

```
        end
    end

    dlmwrite('RawProfitMatrix1', ProfitMatrixFirm1);
    dlmwrite('RawProfitMatrix2', ProfitMatrixFirm2);

    QualityGameNEs = NashEquilibriaBySupportEnumeration ...
                                (ProfitMatrixFirm1, ProfitMatrixFirm2);

    % profit matrices are edited for saving

    Firm2qualitystrats = 0:qualityInterval:latticeInfo.QualityCount*...
                            qualityInterval;

    ProfitMatrixFirm1 = vertcat(Firm2qualitystrats,ProfitMatrixFirm1);
    ProfitMatrixFirm2 = vertcat(Firm2qualitystrats,ProfitMatrixFirm2);

    Firm1qualitystrats = transpose(horzcat(666, Firm2qualitystrats));

    ProfitMatrixFirm1 = horzcat(Firm1qualitystrats,ProfitMatrixFirm1);
    ProfitMatrixFirm2 = horzcat(Firm1qualitystrats,ProfitMatrixFirm2);

    dlmwrite('ProfitMatrix1', ProfitMatrixFirm1,'delimiter','\t',...
        'precision',6);
    dlmwrite('ProfitMatrix2', ProfitMatrixFirm2,'delimiter','\t',...
        'precision',6);

    countFID = fopen('FirstStageCountofNEs','w');
    fwrite(countFID,sprintf('# of NEs: %g',length(QualityGameNEs)),'uchar');
    fclose(countFID);

    save('FirstStageNEdump','QualityGameNEs');

    dlmwrite('FirstStageNEStrategiesForRowplayer',...
            [QualityGameNEs(:).rowPlayerStrategyProfile]);
    dlmwrite('FirstStageNEStrategiesForColumnPlayer',...
            [QualityGameNEs(:).columnPlayerStrategyProfile]);

    dlmwrite('FirstStageFirstFoundNE',...
        [QualityGameNEs(1).rowPlayerStrategyProfile ...
        QualityGameNEs(1).columnPlayerStrategyProfile]);

    cd('..')

    retVal = QualityGameNEs;
end
```

E.3 NashEquilibriaBySupportEnumeration.m

```
% code based on Algorithm 1  in
% "Enumeration of Nash equilibria for two-player games",
% David Avis, Gabriel D. Rosenberg, Rahul Savani, Bernhard von Stengel,
% 2009

% assumptions:
%   nxn-payoff matrices
%   the game is non-degenerate

function NashEquilibria = NashEquilibriaBySupportEnumeration(A,B)

NashEquilibria = [];

n = length(A);

% generate all possible subsets of supports for solving the Nash
% equilibria

setOfSupports = 1:n;

generatedSupports(1).support = 0;
generatedSupports(2).support = setOfSupports(1);


for i = 2:length(setOfSupports)
    countOfGeneratedSupports = length(generatedSupports);
    for j = 1:countOfGeneratedSupports
            if generatedSupports(j).support == 0
                generatedSupports(j+countOfGeneratedSupports).support ...
                                                    = setOfSupports(i);
            else
                generatedSupports(j+countOfGeneratedSupports).support ...
                    = cat(1,generatedSupports(j).support,...
                        setOfSupports(i));
            end
    end
end

for i = 1:length(generatedSupports)
    for j = 1:length(generatedSupports)
        if (generatedSupports(i).support(1) ~= 0) && ...
                (generatedSupports(j).support(1) ~= 0) && ...
                length(generatedSupports(i).support) == ...
                length(generatedSupports(j).support)
```

```
% matrix is constructed to solve the linear equations

constructA = A;
constructB = B;

CountOfDeletedRows = 0;
CountOfDeletedColumns = 0;

for l = 1:n % checking and deleting rows and columns

    % row to be deleted
    if ismember(l,generatedSupports(i).support) == 0
        constructA(l-CountOfDeletedRows,:) = [];
        constructB(l-CountOfDeletedRows,:) = [];

        CountOfDeletedRows = CountOfDeletedRows +1;

    end

    % column to be deleted
    if ismember(l, generatedSupports(j).support) == 0
        constructA(:,l-CountOfDeletedColumns) = [];
        constructB(:,l-CountOfDeletedColumns) = [];

        CountOfDeletedColumns = CountOfDeletedColumns +1;
    end

end

% transposing B
constructB = transpose(constructB);

% adding padding to A and B

k = length(constructA);

constructA(:,k+1) = -1;
constructA(k+1,:) = 1;
constructA(k+1,k+1) = 0;

k = length(constructB);

constructB(:,k+1) = -1;
constructB(k+1,:) = 1;
constructB(k+1,k+1) = 0;

% finalizing the matrix used to solve the linear equations
```

```
k = length(constructB);

constructB = [constructB zeros(k,k)];

k = length(constructA);

constructA = [zeros(k,k) constructA];

finalMatrix = [constructB; constructA];

righthandSide = zeros(length(finalMatrix),1);

righthandSide(k,1) = 1;
righthandSide(2*k,1) = 1;

%possibleSolution = inv(finalMatrix)*righthandSide;
possibleSolution = finalMatrix\righthandSide;

% saving the possible mixed strategy weights for further use

possibleX = zeros(length(A),1);
possibleY = zeros(length(B),1);

for l = 1:length(generatedSupports(i).support)
    possibleX(generatedSupports(i).support(l),1) = ...
        possibleSolution(l,1);
end

for l = 1:length(generatedSupports(j).support)
    possibleY(generatedSupports(j).support(l),1) = ...
        possibleSolution(l+k,1);
end

% saving the utilities required for testing

v = possibleSolution(k,1);
u = possibleSolution(2*k,1);

% checking if the required conditions hold

isNEFlag = 1;

% x >= 0

if min(possibleX) < 0
    isNEFlag = 0;
```

```
end

% y >= 0

if min(possibleY) < 0
    isNEFlag = 0;
end

% condition (2)

MatrixProduct = A*possibleY;
maxMatrixProduct = max(MatrixProduct);

% this defines the required accuracy for comparisons -
% better users of MATLAB would probably do this
% differently and better/more elegantly
requiredAccuracy = 10^-6;

for l = 1:length(possibleX)
    if possibleX(l,1) > 0 && ...
            (abs(MatrixProduct(l,1) - u) > requiredAccuracy...
            || abs(MatrixProduct(l,1) - maxMatrixProduct) ...
            > requiredAccuracy || ...
            abs(u - maxMatrixProduct) > requiredAccuracy)
        isNEFlag = 0;
    end
end

 % condition (3)

 MatrixProduct = transpose(B)*possibleX;
 maxMatrixProduct = max(MatrixProduct);

 for l = 1:length(possibleY)
    if possibleY(l,1) > 0 && ...
            (abs(MatrixProduct(l,1) - v) > requiredAccuracy...
            || abs(MatrixProduct(l,1) - maxMatrixProduct) ...
            > requiredAccuracy || ...
            abs(v - maxMatrixProduct) > requiredAccuracy)
        isNEFlag = 0;
    end
 end

 % if all conditions hold we save the found NE
 % (last two conditions disregard possible solutions with
 % singular finalMatrix)
```

```
            if isNEFlag == 1 && max(isnan(possibleX)) == 0 && ...
                    max(isnan(possibleY)) == 0

               currentCountOfNEs = length(NashEquilibria);

               NashEquilibria(currentCountOfNEs+1).rowPlayerStrategyProfile...
                   = possibleX;
               NashEquilibria(currentCountOfNEs+1).columnPlayerStrategyProfile...
                   = possibleY;

               NashEquilibria(currentCountOfNEs+1).rowPlayerUtility = u;
               NashEquilibria(currentCountOfNEs+1).columnPlayerUtility = v;

            end

        end
    end
end



end
```

E.4 SecondStageNESamples.m

```
function retVal = SecondStageNESamples(latticeInfo, firm1strategy, ...
                                firm2strategy, tspan, parameters,trend)

    iterationDirectoryName = sprintf('./SecondStage.firm1q %g. firm2q %g/',...
        firm1strategy.Quality, firm2strategy.Quality);

    mkdir(iterationDirectoryName)
    cd(iterationDirectoryName)

    matDim = latticeInfo.SamplesCount+1;

    ProfitMatrixFirm1 = zeros(matDim,matDim);
    ProfitMatrixFirm2 = zeros(matDim,matDim);

    % parfor not implemented here since the code already takes advantage
    % of parallel computing in ParallelLatticeFindThidStageNEPrices-
    % function which is called in the loop

    samplesInterval = latticeInfo.SamplesInterval;

    for i = 1:matDim % rows
```

```
    for j = 1:matDim % columns

        firm1strategy.Samples = (i-1)*samplesInterval;
        firm2strategy.Samples = (j-1)*samplesInterval;

        PriceGameNEs = ThirdStageNEPricesParallel(latticeInfo, ...
                firm1strategy, firm2strategy, tspan, parameters,trend);

        % It is possible that there are multiple equilibria in the
        % third stage price choosing game -> necessary to choose
        % between those

        % I arbitrarily choose the first NE that the algorithm finds


        ProfitMatrixFirm1(i,j) = PriceGameNEs(1).rowPlayerUtility;
        ProfitMatrixFirm2(i,j) = PriceGameNEs(1).columnPlayerUtility;
    end
end

dlmwrite('RawProfitMatrix1', ProfitMatrixFirm1);
dlmwrite('RawProfitMatrix2', ProfitMatrixFirm2);

dlmwrite('NEPriceStrategyFirm1', ...
            PriceGameNEs(1).rowPlayerStrategyProfile);
dlmwrite('NEPriceStrategyFirm2', ...
            PriceGameNEs(1).columnPlayerStrategyProfile);

SampleGameNEs = NashEquilibriaBySupportEnumeration ...
                            (ProfitMatrixFirm1, ProfitMatrixFirm2);

% profit matrices are edited for saving

Firm2samplesstrats = 0:samplesInterval:latticeInfo.SamplesCount*...
                        samplesInterval;

ProfitMatrixFirm1 = vertcat(Firm2samplesstrats,ProfitMatrixFirm1);
ProfitMatrixFirm2 = vertcat(Firm2samplesstrats,ProfitMatrixFirm2);

Firm1samplesstrats = transpose(horzcat(666, Firm2samplesstrats));

ProfitMatrixFirm1 = horzcat(Firm1samplesstrats,ProfitMatrixFirm1);
ProfitMatrixFirm2 = horzcat(Firm1samplesstrats,ProfitMatrixFirm2);

dlmwrite('ProfitMatrix1', ProfitMatrixFirm1,'delimiter','\t',...
    'precision',6);
dlmwrite('ProfitMatrix2', ProfitMatrixFirm2,'delimiter','\t',...
```

```
        'precision',6);

    countFID = fopen('SecondStageCountofNEs','w');
    fwrite(countFID,sprintf('# of NEs: %g',length(SampleGameNEs)),'uchar');
    fclose(countFID);

    save('SecondStageNEdump','SampleGameNEs');

    dlmwrite('SecondStageNEStrategiesForRowplayer', ...
        [SampleGameNEs(:).rowPlayerStrategyProfile]);
    dlmwrite('SecondStageNEStrategiesForColumnPlayer', ...
        [SampleGameNEs(:).columnPlayerStrategyProfile]);

    dlmwrite('SecondStageFirstFoundNE',...
        [SampleGameNEs(1).rowPlayerStrategyProfile ...
        SampleGameNEs(1).columnPlayerStrategyProfile]);

    cd('..')

    retVal = SampleGameNEs;
end
```

E.5 ThirdStageNEPricesParallel.m

```
function retVal = ThirdStageNEPricesParallel(latticeInfo, firm1strategy, ...
                    firm2strategy, tspan, parameters,trend)
    iterationDirectoryName = ...
        sprintf('./ThirdStage.firm1q %g, S %g. firm2q %g, S %g/', ...
        firm1strategy.Quality, firm1strategy.Samples, ...
        firm2strategy.Quality, firm2strategy.Samples);

    mkdir(iterationDirectoryName)
    cd(iterationDirectoryName)

    matDim = latticeInfo.PriceCount+1;

    ProfitMatrixFirm1 = zeros(matDim,matDim);
    ProfitMatrixFirm2 = zeros(matDim,matDim);

    parfor i = 1:matDim  % row values
        TempProfitsFirm1 = zeros(1,matDim);
        TempProfitsFirm2 = zeros(1,matDim);

        for j = 1:matDim % column values
            TempProfitsFirm1(j) = parallelWrapperProfitsFirm1(i, j, ...
                latticeInfo, tspan, parameters, trend, ...
```

```
            firm1strategy, firm2strategy);
        TempProfitsFirm2(j) = parallelWrapperProfitsFirm2(i, j, ...
            latticeInfo, tspan, parameters, trend, ...
            firm1strategy, firm2strategy);
    end

    ProfitMatrixFirm1(i,:) = TempProfitsFirm1;
    ProfitMatrixFirm2(i,:) = TempProfitsFirm2;

end

%save the resulting profit matrices

PriceGameNEs = NashEquilibriaBySupportEnumeration ...
                        (ProfitMatrixFirm1,ProfitMatrixFirm2);

dlmwrite('RawProfitMatrix1', ProfitMatrixFirm1);
dlmwrite('RawProfitMatrix2', ProfitMatrixFirm2);

% profit matrices are edited for saving
% possibly should be done with a new function

Firm2pricestrats = 0:latticeInfo.PriceInterval:latticeInfo.PriceCount*...
                    latticeInfo.PriceInterval;

ProfitMatrixFirm1 = vertcat(Firm2pricestrats,ProfitMatrixFirm1);
ProfitMatrixFirm2 = vertcat(Firm2pricestrats,ProfitMatrixFirm2);

Firm1pricestrats = transpose(horzcat(666, Firm2pricestrats));

ProfitMatrixFirm1 = horzcat(Firm1pricestrats,ProfitMatrixFirm1);
ProfitMatrixFirm2 = horzcat(Firm1pricestrats,ProfitMatrixFirm2);

dlmwrite('ProfitMatrix1', ProfitMatrixFirm1,'delimiter','\t',...
    'precision',6);
dlmwrite('ProfitMatrix2', ProfitMatrixFirm2,'delimiter','\t',...
    'precision',6);

% save the found NE

countFID = fopen('ThirdStageCountofNEs','w');
fwrite(countFID,sprintf('# of NEs: %g',length(PriceGameNEs)),'uchar');
fclose(countFID);

save('ThirdStageNEdump','PriceGameNEs');

dlmwrite('ThirdStageNEStrategiesForRowplayer', ...
```

```
        [PriceGameNEs(:).rowPlayerStrategyProfile]);
    dlmwrite('ThirdStageNEStrategiesForColumnPlayer', ...
        [PriceGameNEs(:).columnPlayerStrategyProfile]);

    dlmwrite( 'ThirdStageFirstFoundNE', ...
        [PriceGameNEs(1).rowPlayerStrategyProfile ...
        PriceGameNEs(1).columnPlayerStrategyProfile]);

    cd('..')

retVal = PriceGameNEs;
end
```

E.6 parallelWrapperProfitsFirm1.m

```
function retVal = parallelWrapperProfitsFirm1(i,j, latticeInfo, tspan, ...
                        parameters, trend, firm1strategy, firm2strategy)

    firm1strategy.Price = (i-1)*latticeInfo.PriceInterval;
    firm2strategy.Price = (j-1)*latticeInfo.PriceInterval;

    retVal = profitsFirm1(tspan, parameters, trend, firm1strategy, ...
                                                            firm2strategy);
end
```

E.7 profitsFirm1.m

```
function retVal = profitsFirm1(tspan, parameters, trend, firm1strategy,...
                                                            firm2strategy)
    t = linspace(tspan(1),tspan(2),tspan(2)*100);
    revenue = zeros(1,length(t));

    A1 = firm1strategy.Samples;
    A2 = firm2strategy.Samples;
    N  = parameters(1);

    initialState = [0,0,0];

    % each consumer has the same probability to get a free sample, and the
    % probabilities are independent between firms

    initialState(1) = N.*(A1./N).*(1-A2./N);
    initialState(2) = N.*(1-A1./N).*(A2./N);
    initialState(3) = N.*(A1./N).*(A2./N);
```

```
    ratk = ode45(@duopolyDES,tspan,initialState,[],parameters,trend,...
                                        firm1strategy,firm2strategy);

    for i = 1:length(t)
        revenue(i) = revenueFirm1(t(i),ratk, parameters,trend,...
                                        firm1strategy,firm2strategy);
    end

    netrevenue = trapz(revenue)*(t(2)-t(1));

    retVal = netrevenue - A1.*parameters(6).*...
                    (firm1strategy.Quality.^2+parameters(7));

end
```

E.8 duopolyDES.m

```
function retval = duopolyDES(t,x, parameters,trend, firm1strategy,...
                                                    firm2strategy)

    N = parameters(1);
    mu = parameters(2);
    beta = parameters(4);
    lambda = parameters(5);

    a1 = firm1strategy.Quality;
    p1 = firm1strategy.Price;

    a2 = firm2strategy.Quality;
    p2 = firm2strategy.Price;

    q1 = x(1)+x(3); q2 = x(2)+x(3);

    if trend.type == 1 % linear trend
        effectOfTrendProduct1 = trend.d*a1*q1/N;
        effectOfTrendProduct2 = trend.d*a2*q2/N;
    elseif trend.type == 2 % parabeloid trend
        A = -trend.d/(trend.c^2);
        B = 2*trend.d/trend.c;

        effectOfTrendProduct1 = a1*(A*q1^2 + B*q1);
        effectOfTrendProduct2 = a2*(A*q2^2 + B*q2);
    end

    denomS = 1 + exp((1/mu)*(a1-p1+effectOfTrendProduct1))+...
                            exp((1/mu)*(a2-p2+effectOfTrendProduct2));
```

```
    denomI1 = 1 + exp((1/mu)*(max(a2-a1,0)-p2+effectOfTrendProduct2));
    denomI2 = 1 + exp((1/mu)*(max(a1-a2,0)-p1+effectOfTrendProduct1));

    PSI1 = exp((1/mu)*(a1-p1+effectOfTrendProduct1))/denomS;
    PSI2 = exp((1/mu)*(a2-p2+effectOfTrendProduct2))/denomS;
    PI1I12 = exp((1/mu)*(max(a2-a1,0)-p2+effectOfTrendProduct2))/denomI1;
    PI2I12 = exp((1/mu)*(max(a1-a2,0)-p1+effectOfTrendProduct1))/denomI2;

    I1prime = beta*PSI1*(N-x(1)-x(2)-x(3))-beta*PI1I12*x(1)+...
                                            lambda*x(3)-lambda*x(1);
    I2prime = beta*PSI2*(N-x(1)-x(2)-x(3))-beta*PI2I12*x(2)+...
                                            lambda*x(3)-lambda*x(2);

    I12prime = beta*PI1I12*x(1)+beta*PI2I12*x(2)-2*lambda*x(3);

    retval = [ I1prime; I2prime; I12prime];
end
```

E.9 revenueFirm1.m

```
function retVal = revenueFirm1(t,ratk,parameters,trend,firm1strategy,...
                                    firm2strategy)

    % t refers to time

    state = deval(ratk,t);

    N = parameters(1);
    mu = parameters(2);
    r = parameters(3);
    beta = parameters(4);

    c1 = parameters(6);
    c2 = parameters(7);

    a1 = firm1strategy.Quality;
    p1 = firm1strategy.Price;

    a2 = firm2strategy.Quality;
    p2 = firm2strategy.Price;

    q1 = state(1)+state(3); q2 = state(2)+state(3);

    if trend.type == 1 % linear trend
        effectOfTrendProduct1 = trend.d*a1*q1/N;
        effectOfTrendProduct2 = trend.d*a2*q2/N;
```

```
    elseif trend.type == 2 % parabeloid trend
        A = -trend.d/(trend.c^2);
        B = 2*trend.d/trend.c;

        effectOfTrendProduct1 = a1*(A*q1^2 + B*q1);
        effectOfTrendProduct2 = a2*(A*q2^2 + B*q2);
    end

    % denomI1, PSI2, PI1I12 commented as unnecessary for firm 1 revenue

    denomS = 1 + exp((1/mu)*(a1-p1+effectOfTrendProduct1))+ ...
                            exp((1/mu)*(a2-p2+effectOfTrendProduct2));
    %denomI1 = 1 + exp((1/mu)*(max(a2-a1,0)-p2+effectOfTrendProduct2));
    denomI2 = 1 + exp((1/mu)*(max(a1-a2,0)-p1+effectOfTrendProduct1));

    PSI1 = exp((1/mu)*(a1-p1+effectOfTrendProduct1))/denomS;
    %PSI2 = exp((1/mu)*(a2-p2+effectOfTrendProduct2))/denomS;
    %PI1I12 = exp((1/mu)*(max(a2-a1,0)-p2+effectOfTrendProduct2))/denomI1;
    PI2I12 = exp((1/mu)*(max(a1-a2,0)-p1+effectOfTrendProduct1))/denomI2;

    salesSI1 = beta*PSI1*(N-state(1)-state(2)-state(3));
    salesI2I12 = beta*PI2I12*state(2);

    retVal = exp(-r.*t)*(p1-c1.*(a1.^2+c2))*(salesSI1+salesI2I12);

end
```

E.10 parallelWrapperProfitsFirm2.m

```
function retVal = parallelWrapperProfitsFirm2(i,j, latticeInfo, tspan, ...
                        parameters, trend, firm1strategy, firm2strategy)

    firm1strategy.Price = (i-1)*latticeInfo.PriceInterval;
    firm2strategy.Price = (j-1)*latticeInfo.PriceInterval;

    retVal = profitsFirm2(tspan, parameters, trend, firm1strategy, ...
                                                        firm2strategy);
end
```

E.11 profitsFirm2.m

```
function retVal = profitsFirm2(tspan, parameters, trend, firm1strategy,...
                                                            firm2strategy)

    t = linspace(tspan(1),tspan(2),tspan(2)*100);
```

```
    revenue = zeros(1,length(t));

    A1 = firm1strategy.Samples;
    A2 = firm2strategy.Samples;
    N  = parameters(1);

    initialState = [0,0,0];

    % each consumer has the same probability to get a free sample, and the
    % probabilities are independent between firms

    initialState(1) = N.*(A1./N).*(1-A2./N);
    initialState(2) = N.*(1-A1./N).*(A2./N);
    initialState(3) = N.*(A1./N).*(A2./N);

    ratk = ode45(@duopolyDES,tspan,initialState,[],parameters,trend,...
                                        firm1strategy,firm2strategy);

    for i = 1:length(t)
        revenue(i) = revenueFirm2(t(i),ratk, parameters,trend, ...
                                        firm1strategy, firm2strategy);
    end

    netrevenue = trapz(revenue)*(t(2)-t(1));

    retVal = netrevenue - A2.*parameters(6).*...
                         (firm2strategy.Quality.^2+parameters(7));

end
```

E.12 revenueFirm2.m

```
function retVal = revenueFirm2(t,ratk,parameters,trend,firm1strategy, ...
                                                          firm2strategy)

    % t refers to time

    state = deval(ratk,t);

    N = parameters(1);
    mu = parameters(2);
    r = parameters(3);
    beta = parameters(4);

    c1 = parameters(6);
    c2 = parameters(7);
```

```
    a1 = firm1strategy.Quality;
    p1 = firm1strategy.Price;

    a2 = firm2strategy.Quality;
    p2 = firm2strategy.Price;

    q1 = state(1)+state(3); q2 = state(2)+state(3);

    if trend.type == 1 % linear trend
        effectOfTrendProduct1 = trend.d*a1*q1/N;
        effectOfTrendProduct2 = trend.d*a2*q2/N;
    elseif trend.type == 2 % parabeloid trend
        A = -trend.d/(trend.c^2);
        B = 2*trend.d/trend.c;

        effectOfTrendProduct1 = a1*(A*q1^2 + B*q1);
        effectOfTrendProduct2 = a2*(A*q2^2 + B*q2);
    end

    % denomI2, PSI1, PI2I12 commented as unnecessary for firm 1 revenue

    denomS = 1 + exp((1/mu)*(a1-p1+effectOfTrendProduct1))+...
                        exp((1/mu)*(a2-p2+effectOfTrendProduct2));
    denomI1 = 1 + exp((1/mu)*(max(a2-a1,0)-p2+effectOfTrendProduct2));
    %denomI2 = 1 + exp((1/mu)*(max(a1-a2,0)-p1+effectOfTrendProduct1));

    %PSI1 = exp((1/mu)*(a1-p1+effectOfTrendProduct1))/denomS;
    PSI2 = exp((1/mu)*(a2-p2+effectOfTrendProduct2))/denomS;
    PI1I12 = exp((1/mu)*(max(a2-a1,0)-p2+effectOfTrendProduct2))/denomI1;
    %PI2I12 = exp((1/mu)*(max(a1-a2,0)-p1+effectOfTrendProduct1))/denomI2;

    salesSI2 = beta*PSI2*(N-state(1)-state(2)-state(3));
    salesI1I12 = beta*PI1I12*state(1);

    retVal = exp(-r.*t)*(p2-c1.*(a2.^2+c2))*(salesSI2+salesI1I12);

end
```