Learning of Long Term Optimal Capacity: The Case of a Monopolist Facing Uncertainty.

by

Linda Margrethe Lien Knarvik

Master's thesis

This masters thesis is submitted for completion of the following degree

Master of Economics

University of Bergen, Department of Economics

December 2013

UNIVERSITETET I BERGEN

Acknowledgements

This thesis marks the end of several years of studies at the University of Bergen. Although it at times has been both frustrating and difficult, it has overall been an exciting process. A number of people deserve thanks for their support and help over the years.

First, I want to thank my supervisor Professor Sjur Didrik Flåm for being a good supporter and guide trough the process of writing this thesis, and for usefull discussions and quick replays. - It took some time, now it's complete. Thank you for your patience!

Many great fellow students in the Department of Mathematics and the Department of Economics have contributed to a social and enjoyable time over the years. A special thanks goes to Charlotte, Monica and Torje for curricular cooperation, not to mention countless hours of more or less interesting discussions over lunch. Charlotte deserves further thanks for help during the editing phase - It was proven essential ☺

A thanks also goes to Assistant Professor Amir M. Hashemi at Bergen University College. He gave me a recommendation many years ago which provided me with the opportunity to study at the university. I will forever be grateful for his trust in my abilities.

My gratitude also goes to the helping hands in the final editing phase, John Brottemsmo and Claus Jacobsen. They are both inspiring and generous people I am glad to know.

Last, but far from least, I would like to express my huge gratitude towards the most important people in my life. To my husband, to my two daughters and to my son:

"You raise me up . . . to more than I can be"

The strong support from my family have been essential, without it I would never have been able to get this far ♡

Any errors or omissions in this thesis are the author's sole responsibility.

Bergen - November 29, 2013

Abstract

Learning of Long Term Optimal Capacity:
The Case of a Monopolist Facing Uncertainty.

by

Linda Margrethe Lien Knarvik, Master in Economics

University of Bergen, December 2013

Supervisor: Professor Sjur Didrik Flåm

The key question is: *How can the optimal long term level of capacity be determined, if only short term conditions are known, and the agent's skills to uncover the long term characteristics are limited?*

This thesis develops a simple and unified solution algorithm as an alternative approach to long term decisions under uncertainty. It is based on stochastic approximation and the gradient method, and is designed such that an agent acquires knowledge of the best long term capacity via adaptive learning from discrete observations in a stochastic market.

By non-linear programming, the choice of long term capacity made by a profit maximizing monopolist under uncertainty is simulated numerically. The effect of various types of uncertainty on the agent's long term decisions are examined. Both one (capacity) and multiple learning objects (capacity price & slope) are implemented in the experiments.

Based on the numerical experiments it is shown that the algorithms produce results in line with economic intuition. The numerical results are also compliant to others findings for a wide range of initial guesses, and for all types of uncertainty applied in this thesis. Even when demand is unknown, and a proxy demand curve is applied, the algorithms provides sufficient information so that the agent becomes able to make an effecient long term capacity choice under uncertainty.

The computer program AMPL and the solver CPLEX 11.2 have been used to conduct the numerical experiments.

Key words: Adaptive Learning - Gradient Method - Proxy Demand - Dual Value - Convergence - Long Term Choice - Non-linear Programming.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Capacity matters! In terms of money spent or money earned, all businesses have, at some point, limits which restrain their possible output. It may be a physical limit, limited "know how", little, or no control over the availability of an input or even unknown demand. Some of these limitations can easily be overcome by employing people with the right knowledge, while others are more difficult to overcome. One such challenging constraint may be the availability of a random input over which the producer has no control, possibly with an unknown nature or unfamiliar distribution.

As an undergraduate in economics you encounter the method of maximization, usually solved with a deterministic knowledge of parameters and constraints, or at least a known distribution of those. A question then comesto mind:

> *How does one solve maximization problems, if the properties of random values are unknown, or one lacks sufficient skills to asses their characteristics?*

To solve maximization problems under uncertainty requires a relatively high level of mathematics and statistics. Even with these kinds of skills, the problems may be hard to find explicit solutions for and a simpler way to reach the target is in most cases preferable if available. This thesis suggests one such simpler way to reach the target.

Section 1.1 presents the motivation for why capacity decisions are interesting and provide the reader with the motivation for the experiments conducted in this thesis. This Section will also present the main objective of the experiments. Section 1.2 discusses some of the most significant limitations inherent in the model, and hence in the results derived from it. The structure of the remainder of the thesis is presented in Section 1.3.

1.1 Motivation and Objective

Limited physical transfer capacity through the electricity grid connecting domestic areas in Norway to other countries can cause a sharp increase in prices. Subsequently this has resulted in an uncomfortably high electricity bill for energy users. So lack of capacity does matter and is potentially costly in money terms. A problem naturally then arise:

> *How can optimal capacity be determined, or learned, in a simple way based on information available through realized values?*

Optimization problems under uncertainty are usually solved by using some sort of simulations, stochastic approximation or even more advanced statistical methods. Common for these methods is a requirement of a relatively high level of mathematical and/or statistical skills to both develop and analyze the results. Also a number of assumptions are required, assumptions which may be more or less adequate for the real world problem in question. More often than not, a huge amount of historical data must be examined, both to implement the solution, and to control the accuracy of the findings.

This thesis will suggest an alternative approach to solving these kinds of problems, an approach based on the ability to learn from discrete observations of current market conditions. The purposed solution method can therefore be seen as a simpler way to find an optimal solution in a world of uncertainty, as the assumptions and skills needed to be able to implement the method are limited.

The main objective in this thesis is therefore to develop a unified and straightforward solution algorithm which enables an average agent to learn the best choice of long term capacity, when faced with varying market conditions. To be deemed a unified solution, the method must be applicable for both deterministic and stochastic cases without significant modifications.

Using a method similar to that presented by Flåm and Sandmark (2000), the agent will adopt the needed knowledge trough discrete observations of current market conditions. The agent will determine the best level of capacity when faced with both random input availability and random demand, without any prior knowledge of the market conditions, using a solution algorithm based on the gradient method and by using adaptive learning.

A non-deterministic economic reality with partially available information is assumed. Realized random values can be observed before the short term decision is determined, but only after the current capacity decision has been made. By assumption the capacity can be changed by any incremental size at the end of each period, and will be available for production in the forthcoming period.

1.2 Limitations

This thesis is founded on a purely synthetic model, and is depolyed to display some standard properties seen in the real world. The results, i.e. numerical values, have no real interpretation for any real world problem and hence, can not be seen as a stand-alone empirical result. Real valued data could have been applied in the solution algorithm, but this would reduce the possibility to control the compliance between the results obtained by the method and the expected theoretical results. In this thesis compliance control is selected over application of data from a real economic setting.

The basic assumption of supply and demand is that the higher the price, the less quantity sold, for normal goods (Jones, 2004, p. 87). This is applied in this thesis. Note that the magnitudes of the effects are merely a result of choices made in the setup of the model, and do not have any significant interest beside the direction of the effect.

Immediately effective incremental changes in capacity at no extra cost is a simplification compared to a real world problem. The aim of this thesis is to show that a learning heuristic[1] can produce results in line with economic intuition and compliant with others findings. The aim is not to fully characterize optimal capacity under uncertainty. To account for this flaw, a flexibility cost is included. Either way, a time-lag will give a prolonged learning curve, but otherwise does not affect the results obtained.

The uncertainty is based on computer generated random values. In the computer program AMPL (A Modeling Language for Mathematical Programming) applied to conduct the experiments, the command `param R default Uniform(a,b);` produces a number `R` within the range specified by `a` and `b` based on a seed provided by the system[2]. The resulting values are therefore pseudo-random values only, but for the current project this is deemed sufficient (Fourer, Gay, and Kernighan, 2003, p. 122 & p. 209).

This thesis is written in the document preparation system LaTeX (2009) and used the bibliography management system BibTeX (2009), following the LaTeX-distribution, trough the program TeXnicCenter (2013). Being relatively new to this system, some words have split at the wrong places, and some figures have been placed differently than intended, and the reader are kindly asked to disregard this small errors.[3]

[1] Heuristic: Strategy for learning or problem solving, often based on experience.
[2] The seed can be predefined to assure reproducible results. Section 4.4 describe how this is done.
[3] The layout is based on a template provided by Thorsten Karrer, RWTH Aachen University (Karrer, 2012), modified for the Norwegian letters æ, ø and å in the Bibliography.

1.3 Structure

This chapter has introduced the motivation for the experiments conducted in this thesis, and asked the leading question on how to find a simple and unified solution method applicable to solve difficult maximization problems if the agent has limited abilities. The main objective was given and some of the limitations to the chosen set-up was discussed.

Chapter 2 presents some of the relevant background literature to which this thesis relates, and give a brief summary of some selected papers on both production planning under uncertainty and adaptive learning. This chapter also introduces a simple adaptive learning model applied in the developed algorithms to unveil the true value of a learning subject.

Chapter 3 introduces the model for the agent's business. Both the demand and cost structure are described, and uncertainty as it is modelled in this thesis is presented. The difference between ex-ante[4] and ex-post[5] decisions are discussed and formalized. The chapter is concluded with a summary of the agent's deterministic and stochastic objectives.

Chapter 4 considers the theoretical framework needed to develop a simple and unified solution procedure for uncertainty problems. Stochastic approximation, the online gradient method and the Kuhn-Tucker conditions are presented as they form the foundation on which the algorithms rely. This chapter is concluded by a presentation of the algorithms in pseudo code, and briefly discuss required syntax for programming in AMPL.

In Chapter 5, the results obtained from implementing the algorithms for various stochastic cases are presented. These are then compared to the results obtained for the deterministic case. This chapter also summarizes the numerical values assigned to the parameters in the model, and presents the different initial values applied in the experiments.

In Chapter 6 the compliance between the obtained results and theoretical postulates is discussed. An analysis of the reasoning behind the differing results between the algorithms is included. The applied learning process and the assumptions is also discussed in this chapter. By way of conclusion a critique of the main weaknesses of the applied method is included.

Chapter 7 concludes the thesis and presents a short summary of contributions, points to some avenues for further research and offers a conclusion.

[4] Ex-ante decision: A decision made **before** uncertainty is realized.
[5] Ex-post decision: A decision made **after** uncertainty is realized.

Chapter 2

Background

Consider a monopolist facing uncertain market conditions of some unspecified type. In the search for answers on how the monopolist reacts to uncertainty, computer simulation tools and numerical methods are applied. The experiment is founded on a rich literature concerning the impact of uncertain demand, stochastic modeling, adaptive learning and approximation theory. In order to frame the experiments, only a handful of related papers are selected as the area spans more than half a century of research.

This thesis relates to existing literature along two main lines of research, namely the effect of uncertainty in production and planning and research on adaptive learning, also called Bayesian updating. In Section 2.1 the main findings from some of the literature concerning uncertainty and production planning are presented, and provides an indication of what to expect in terms of results from the conducted experiments. In Section 2.2 papers considering the adaptive learning model are discussed, and will serve as an introduction to the main driver of the applied method, namely the updating of the agent's knowledge between periods. The adaptive expectation hypothesis is defined in Section 2.2, and a simple blueprint for an adaptive learning model is given.

2.1 Uncertainty in Production and Planning

Early research on the subject of *uncertain demand* conducted by Leland (1972) was published in the article "Theory of the firm facing uncertain demand". It introduced the *"principle of increasing uncertainty - PIU"* (p. 279). The PIU, as Leland defined it, implied an increased variation in total profit for increasing levels of profit, subject to changes in either price or quantity.

Leland (1972) used a framework of a utility maximizing agent under four different *"behavioral modes"* separated by which ex-post control the agent has to counter the effect of the risk by, after the realization of uncertainty (p. 279). He showed that uncertainty will affect the agent, and that the impact depends on the agent's mode and risk attitude. Unambiguous results derived by Leland showed that a monopolist would not be indifferent between the choice of quantity or price as ex-post control when faced with a risk. This is because the effect of the ex-post controls not only depends on the risk attitude, but also on the nature of the risk and the demand and cost curves of the company (p. 286). It was later shown by Korkie (1975) that Leland's "PIU" principle was easily violated for certain types of uncertain demand dependent of behavioral mode. This issue is also discussed by Hau (2004), among others.

Empirical findings to support, or counter, risk theory are limited in terms of actual agents answering questions on his or here risk attitude. However, Aiginger (1985) conducted a research specifically aimed at confronting existing risk theory with empirics by collecting insight trough questioners. He aimed to show *"sufficient conditions under which an unambiguous answer"* can be given about the effect of facing a risk on production levels (p. 52). His main empirical findings point towards a tendency of reduced output, if faced with uncertainty (p. 67).

Like Leland (1972), Aiginger (1985) also found unambiguous answers which pointed to *"risk attitude"*, *"technological concavity"* and *"ex-post-flexibility"* as factors to consider when addressing the impact of risk on production quantity choices. Aiginger (1987) discusses the influence of uncertainty on optimal decisions and concludes that *"production will most probably be lower under uncertainty than under certainty"* (p. 178).

The effect of a risk on the agent's long term decision regarding production capacity was considered neither by Leland (1972) nor Aiginger (1985) so their main findings are not directly applicable in this context. However, one might expect a risk averse agent (i.e. having linear utility and concave technology) to reduce output when faced with uncertainty based on the propositions given by Leland and Aiginger.

Driver, Abubacker, and Argiris (1996) model a risk-neutral monopolist faced with multiplicative demand uncertainty, and determine using numerical methods that the *"optimal uncertainty capacity is often less than the certainty level"* (p. 532). When deriving the results, they use a price elasticity approach under a Leontief technology to show that the optimal level of capacity under uncertainty depends on price elasticity and capital intensity. They also conclude that the relation between fixed and variable cost will affect the *"downward bias"* of the uncertain demand (p. 532).

In contrast to the current model in this thesis, where perfectly reversible capacity decisions are assumed, Driver, Abubacker, and Argiris (1996) assume capital costs to be irreversible once the decision has been made. They also assume flexible prices and point out that *"price acts as an ex-post control to counter the risk of inflexible excess capacity"* (p. 529)

These differences may affect the results obtained, and hence their findings are not directly comparable to the ones derived in this thesis. However, the findings of Driver, Abubacker, and Argiris (1996) might lead to the presumption that higher fixed costs compared to variable costs will lead to a even lower long term capacity choice if faced with the same uncertainty, despite a flexible capital assumption [1].

Summing up the above, the authors point to the following results:

- Uncertainty in demand will most likely result in a lower choice of production, and hence one might expect a lower best choice of long term capacity.
- Higher fixed costs compared to variable costs might imply an even lower long term best choice of capacity, when faced with uncertainty in demand (given that all other parameters are the same).

2.2 Adaptive Learning

The adaptive expectations hypothesis is a key factor in many economic models and has been the subject for a lot of research and has resulted in an extensive literature. As early as in the 1930's John Maynard Keynes highlighted the influence of *long term expectations* as opposed to short term expectations (Keynes, 1936).

The adaptive expectations hypothesis is an axiom on how economic agents form their expectations about some future value of a factor by adjusting past observed values for the same factor (Pallister and Law, 2006, p. 14). It can be formulated as a model of learning, for example to learn a correct price. A simple construction of adaptive learning of price can be given by:

$$p_{t+1} \leftarrow (1-\delta_t)p_t + \delta_t \hat{p}_t \tag{2.1}$$

where p_{t+1} denotes expected price in the next period, p_t is initial expected price and $\hat{p}_t$ is actual observed price, all within the current period, where δ_t is a weight parameter assumed to be between 0 and 1. The above states that the future expected price will be a projection of current best guess and an error-adjustment accounting for the deviation between actual observed price and current best guess.

[1] A literature review of models in production planning under uncertainty can be found in Mula, Poler, García-Sabater, and Lario (2006).

An early paper on learning in economics by Cross (1973) presents a scenario tree model, where the agent chooses among alternative actions, each with a defined probability of occurring. For the monopoly case, Cross shows that if an agent is faced with multiple states of the world, *"the learning process will lead to choices tending to maximize expected pay-offs"* (p. 254).

Research based on adaptive learning, or other learning heuristics has increased significantly over the years following the paper of Cross (1973). Today adaptive learning is used within a wide range of areas, spanning from physics and signal inference, to psychology and behavioral studies using various heuristic models. In economics these methods, combined with stochastic approximation are used in subjects such as decision making under uncertainty, the theory of rationality and Bayesian updating.

The adaptive learning scheme implemented in this thesis is inspired by Flåm and Sandmark (2000). Flåm and Sandmark consider a market of producers who are incapable of assessing their demand functions, but highly capable of knowing the marginal costs. They suggest an *"adaptive learning scheme"* so the producers eventually learn the true market conditions surrounding them. They show that repeated updating of beliefs as a moving average, similar to eq. (2.1) provide convergence in expectations towards market clearing prices and a individual optimal quantity. (p. 7)[2]

Behind the seemingly simple model suggested by Flåm and Sandmark (2000) are some rather complex stochastic approximations. Additionally advanced mathematical are used to show the equilibrium properties for the solutions found. They show that a market participant does not need any special skills to eventually learn the level of optimal output and the correct price at which the product may be sold. Some of the same heuristics will be applied in the proposed algorithm to show that an iterative process based on learning can provide a good best choice of long term capacity when faced with uncertainty.

[2] Eq. (8) given in Flåm and Sandmark (2000, p. 4) is $p^{t+1} \leftarrow (1-\delta_t)p^t + \delta_t \hat{p}^t$ which equals eq. (2.1) for a slightly different notation.

Chapter 3

Model

To understand how the hypothetical model is set up one may interpret it in a setting of a hydrological production facility. The size of the reservoir is the capacity, the product is electricity, and rainfall is the inflow. Under the random demand cases, production is constrained by the capacity only and the price is the unknown component. In the random inflow case, capacity is constrained by the size of the reservoir or the amount of rain.

- The *short term problem* is to choose the level of production (q) for current period, limited by the capacity of the reservoir: If rainfall, and hence inflow to the reservoir, is very small the effective production will be constrained despite a large reservoir.
- The *long term problem* is to determine the optimal size of the reservoir (K) when faced with uncertainty regarding future inflows or future prices.[1]

Assume a profit maximizing monopolist, the agent in this model, with a simple model for the business process. He applies one input to produce one output (q). All output units are sold to the market at a price (P). The producer faces a production cost per unit (C), and there are costs related to the currently installed capacity (Ψ). The agent's main objective is to determine the best choice of long term capacity to install in his simple business.

This chapter introduces the model and presents the agent's objectives ex-ante and ex-post. In Section 3.1 uncertainty as it is modeled in this thesis is presented. The capacity in the agent's business is described in Section 3.2, and will define how the capacity constrains the agent's production under uncertainty. Section 3.3 deals with the income side of the agent's business, and discusses how the demand function is modeled. The different types of

[1] The determination of the size of a water reservoir will, in the real world, be subject to analysis of typical pattern of inflow at the location, an analysis conducted based on using historical data for catchment areas published by Norwegian Water Resources and Energy Directorate (NVE).

random demand are also presented. The cost structure of the agent's business is described in Section 3.4, and the difference between running costs associated wth production and costs related to maintaining installed capacity are discussed. In Section 3.5 the formal definition of the agent's objectives both ex-post and ex-ante are given, and includes a short discussion of the difference between the deterministic and stochastic cases.

Notational remarks:
The hat notation will from now on be used to denote realized values assumed to be exogenously known at the start of period t. By this notation $\hat{K}$ is the currently installed capacity and $\hat{\theta}$ is the realized value of uncertainty. In the following, time indexing is suppressed unless it is needed, in order to ease readability.

3.1 Uncertainty

Uncertainty is introduced through a stochastic factor θ, applied differently in the different cases. A uniform random variable is deemed sufficient to provide randomness in the experiments conducted, thus the random factor is a real valued continuous uniformly distributed random variable on the support $[\overline{\theta}, \underline{\theta}]$.

$$\theta_t \sim \mathcal{U}\left(\underline{\theta}, \overline{\theta}\right) \qquad (3.1)$$
$$\text{where} : 0 \leqslant \underline{\theta} < \overline{\theta} < \infty$$

The parameters $\underline{\theta}$ and $\overline{\theta}$, and the significance of the distance between them, will be discussed when results are presented. The uniform probability density function (pdf) of a uniform random variable is given by (Hogg and Tanis, 2005, p. 147):

$$f(\theta) = \begin{cases} \dfrac{1}{\overline{\theta} - \underline{\theta}} & \text{for } \theta \in [\underline{\theta}, \overline{\theta}] \\ 0 & \text{otherwise} \end{cases} \qquad (3.2)$$

Expectation of a uniform random variable is defined as (Hogg and Tanis, 2005, p. 154):

$$E(\theta) = \int_{\underline{\theta}}^{\overline{\theta}} \theta f(\theta) \, \mathrm{d}\theta = \frac{\overline{\theta} + \underline{\theta}}{2} \qquad (3.3)$$

As the random factor is implemented differently in different cases, a shift in notation is used to clarify which case is considered. The notation is given in eq. (3.4) for reference. The general notation θ is used if the equation is valid for all uncertainty cases.

$$\begin{aligned} \theta &:= R && \text{Random Inflow} \\ \theta &:= s && \text{Random Demand} \end{aligned} \qquad (3.4)$$

3.2 Capacity

At the start of each new period the installed capacity, $\hat{K}$, is exogenously given as it results from either an initial guess or an ex-ante decision made in previous period. Initially, the agent chooses capacity more or less at random as the market conditions are unknown to him. The capacity is however assumed bounded:

$$\underline{K} \leqslant \hat{K} \leqslant \overline{K} \tag{3.5}$$

where $\underline{K}$ is the lower bound and $\overline{K}$ is the upper bound of current capacity. Within the hydrology analogy this assumption is partly justified by available historical data for catchment areas published by the Norwegian Water Resources and Energy Directorate (NVE), and partly by tractability in the modeling phase.[2]

The short term choice variable is the production q, naturally assumed to be non-negative. In the deterministic case production is constrained by currently installed capacity $\hat{K}$, and hence the constraint in the deterministic case in each period is:

$$0 \leqslant q \leqslant \hat{K} \tag{3.6}$$

The above constraint applies in the random demand case as the uncertainty enters the model trough the demand function. If the uncertainty is introduced as random inflow, the situation will be different as the production will be constrained by the smallest of the realized random inflow ($\theta := \hat{R}$) and current available capacity ($\hat{K}$). This gives the following effective capacity constraint in each period for the random inflow case:

$$0 \leqslant q \leqslant \min\left(\hat{K}, \hat{R}\right) \tag{3.7}$$

An inequality constraint of the type defined by eq. (3.6) for $\hat{K}, \hat{R} > 0$ may or may not impose a limit on the production quantity compared to the current capacity, $\hat{K}$. The constraint can thus be seen as a chance constraint, and can be implemented in a mathematical program by defining two separate constraints, one for each of the two components.

[2] Based on the amount of available historical data for catchment areas published by NVE for Norway, we may assume the upper limit of the inflow to be known, at least historically. (NVE)

3.3 Demand

The demand during one period is given by a price curve, assumed to be continuous and differentiable. As described in Section 1.2, empirical studies suggest that demand is decreasing in quantity, thus the demand in each period is defined by eq. (3.8) for $a, b > 0$ which may be known or unknown by the agent. Marginal revenue is defined by eq. (3.9).

$$P(q) = a - bq \tag{3.8}$$

$$MR(q) = P(q) + \frac{\partial P(q)}{\partial q} q \tag{3.9}$$

In the deterministic and random inflow cases the price depends only on the choice of production level q. In the random demand cases the price also depends on the outcome of the uncertainty parameter ($\theta := s$), modeled as described in Section 3.1. Following Leland (1972) and Hau (2004), both additive and various types of multiplicative uncertainty will be examined.

Additive uncertainty, labeled "Type 1", describes a situation where the uncertainty term is *added* to the demand function, shown in Figure 3.1 for $s \sim \mathcal{U}(-0.5, 0.5)$. Similarly, multiplicative uncertainty describes a situation for which the uncertainty term is *multiplied* by the demand function in some way. Two types of multiplicative uncertainty will be examined, "Type 2" defined by eq. (3.11) and "Type 3" defined by eq. (3.12).

$$P(q, s) = a - bq + s \qquad \text{Type 1} \tag{3.10}$$

$$P(q, s) = a - (b + s)q \qquad \text{Type 2} \tag{3.11}$$

$$P(q, s) = (a - bq)s \qquad \text{Type 3} \tag{3.12}$$

As pointed out above, and as Figure 3.1 (Type 1) and Figure 3.2 (Type 3) show, the resulting price depends on *how* the uncertainty is introduced in the demand function. Note that the bounds of the uncertainty parameter vary across types, and so will the expected value. To assure a non-negative price and a negative marginal price in quantity in all cases, assume $a > b + s \; \forall \; s$.

From Figure 3.1 we see that the outcome space[3] under the impact of Type 1 uncertainty is constant and hence independent of quantity produced. For Type 3 uncertainty shown in Figure 3.2 the outcome space is decreasing for increasing quantity, implying a reduced volatility in prices, and thus reduced uncertainty for higher production quantities. For Type 2 uncertainty, the situation will be the opposite compared to Type 3 uncertainty.

[3] Outcome space: Distance between upper and lower bound of the resulting prices under uncertainty, indicated by the arrows in Figure 3.1 and Figure 3.2

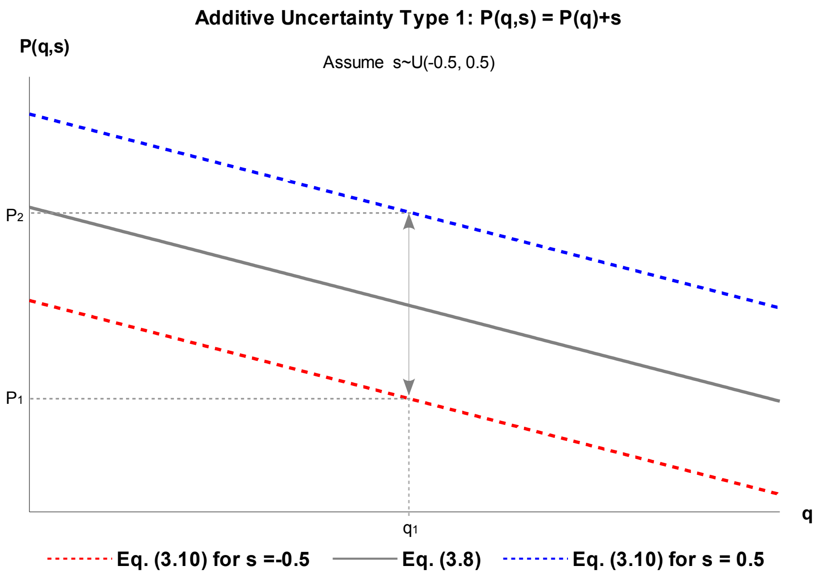


Figure 3.1: Additive Demand Type 1, eq. (3.10): The upper dotted line show $s = 0.5$ and lower dotted line show $s = -0.5$, compared to deterministic demand (middle line). $\updownarrow$ indicates the possible variation in prices $(P_1 \leftrightarrow P_2)$ for the quantity q_1, assuming $s \sim \mathcal{U}(-0.5, 0.5)$

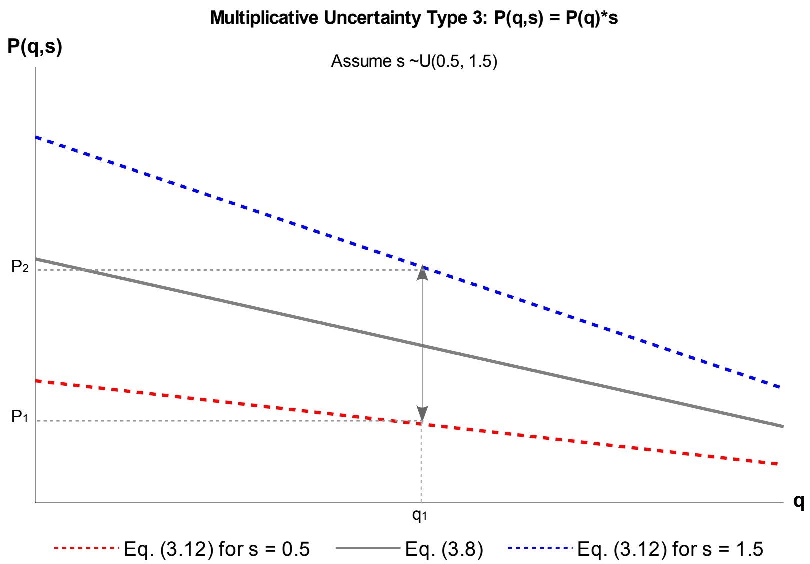


Figure 3.2: Multiplicative Demand Type 3: eq. (3.12) The upper dotted line show $s = 1.5$ and lower dotted line show $s = 0.5$ compared to deterministic demand (middle line). $\updownarrow$ indicates the possible variation in prices $(P_1 \leftrightarrow P_2)$ for the quantity q_1, assuming $s \sim \mathcal{U}(0.5, 1.5)$

3.4 Cost

As the agent owns the business, the cost structure of the business is by assumption known to him. The total costs of the firm is a sum of production costs per unit and costs related to currently installed capacity. The cost of producing one unit is:

$$C(q) = Aq + Bq^2 \tag{3.13}$$

where the two parameters A and B is positive and presumably known to the agent. The parameter A account for the major part of the production costs, while parameter B affects the costs more for higher production. Marginal production cost is denoted $MC(q)$:

$$MC(q) = A + 2Bq > 0 \tag{3.14}$$

The costs of current capacity consists of the depreciation rate, the interest rate paid for the current capacity stock and a flexibility cost. The possibility to install or demolish any incremental units of capacity at any time comes with a cost, called flexibility cost. This can be seen as paying an additional running cost of capacity in order to have the possibility to alter the capital stock as quickly as wanted. The cost of maintaining current capacity $\hat{K}$ is

$$\Psi\left(\hat{K}\right) = (r + d + f)\,\hat{K} \tag{3.15}$$

where r is the interest rate, d is the depreciation rate and f is the flexibility cost, assumed known to the agent. The marginal cost of capacity, denoted $MC(\hat{K})$ given by eq. (3.16), will be positive and constant as long as current capacity is positive.

$$MC(\hat{K}) = \frac{\partial\Psi\left(\hat{K}\right)}{\partial\hat{K}} = (r + d + f) \geqslant 0 \tag{3.16}$$

3.5 The Agent's Objective

The main long term goal for the agent is to determine the long term level of capacity which maximizes profit, given the outcome a random factor, when production is the agent's short term choice. Since the agent has a short term and a long term objective, a distinction between decisions made *before* and *after* realization of the uncertainty is necessary. Assume uncertainty is realized and thus known at the start of period t, while next period's uncertainty is still unknown at time t. The following definitions will be used:

Definition 1 *The agent's **objective ex-post** is to maximize profits by choosing the optimal level of production **after** uncertainty is realized at time t, subject to current capacity $\hat{K}$ (θ_t known).*

Definition 2 *The agent's **objective ex-ante** is to select a level of capacity **before** next period's uncertainty will be known at time $t+1$ in order to maximize long term profits (θ_{t+1} unknown).*

All model components needed are described in the Sections 3.2, 3.3 and 3.4, so without any further discussion **the deterministic objective** for the agent in period t is given by:

$$\max_K \pi(K) = \max_K \left\{ \max_q \pi(q, \hat{K}) \right\} \tag{3.17}$$
$$\text{Where } \pi(q, \hat{K}) = P(q)q - C(q) - \Psi(\hat{K})$$
$$\text{Such that : } 0 \leqslant q \leqslant \hat{K}$$

where $\pi(q, \hat{K})$ is profits, $P(q)$ is unit price obtained in the market as given by eq. (3.8) $C(q)$ is unit cost of production as given by eq. (3.13) and $\Psi(\hat{K})$ is the cost of maintaining the currently installed capacity $\hat{K}$ as given in eq. (3.15).

For a given level of capacity determined in the previous period, the agent chooses q in order to solve $\max_q \pi(q, \hat{K})$ within the current period to maximize current profit. The deterministic objective in eq. (3.17) can be solved by analytical expressions if such are available as there will be no difference between ex-ante and ex-post decisions due to no uncertainty. If, however, the analytical expressions are unavailable, numerical methods such as the gradient method can be applied. This method will be presented in Section 4.1.

As there will be differences between ex-ante and ex-post objectives in the uncertainty cases, the agent's ex-ante objective in the stochastic cases is formulated as an expected value, dependent on the outcome $\hat{\theta}$. The distribution of θ may or may not be known to the agent, and this will make the calculation of the stochastic ex-ante objective a non-trivial job, as discussed in 1.1. **The stochastic objective ex-ante** is thus formally defined by:

$$\max_K E_\theta \left\{ \max_q \Pi(q, \hat{K}, \hat{\theta}) \right\} \tag{3.18}$$

where Π is the profit dependent on current outcome of uncertainty $\hat{\theta}$ and current level of capacity $\hat{K}$. The production q is the control variable in each period and E is the expectation operator as defined in Sydsæter, Strøm, and Berck (2002, Def. 31.10, p. 180).

Solving eq. (3.18) directly may require skills the agent does not have, or involve an evaluation of the expectation of a random factor with unknown distribution. Thus in order to proceed in this non-trivial task, define $\pi(\hat{K}, \hat{\theta})$ as the profit which results from solving the objective ex-post. That is: Solve $\max_q \Pi(\hat{K}, \hat{\theta}, q)$ and find optimal quantity production subject to current (known) values of random variable $(\hat{\theta})$ and capacity $(\hat{K})$. The **ex-post objective** is then defined by:

$$\pi(\hat{K},\hat{\theta}) := \max_q \Pi(q,\hat{K},\hat{\theta}) \tag{3.19}$$

Note that the right hand side of eq. (3.19) are the inner problem of the expectation in the ex-ante objective for the stochastic case given in eq. (3.18). To progress further one utilizes that for a specific value of the random value $\hat{\theta}$ one may define the **ex-post approximation** of expected profit, conditional on $\hat{\theta}$ by:

$$\pi(K) := E_{\hat{\theta}}\left\{\pi(K,\hat{\theta})\right\} \tag{3.20}$$

Why define an ex-post objective? $\pi(\hat{K},\hat{\theta})$ is the result when short term choice of optimal production is determined by $\max_q \Pi(q,\hat{K},\hat{\theta})$ subject to current capacity and realized random factor. The agent decides the short term choice q subject to $\hat{K}$ and $\hat{\theta}$ ex-ante. Ex-post the approximation of expected profit found from the ex-ante choice can be then utilized by the agent to approximate the real long term target, which is to determine the best overall choice of capacity.

In principle, by using the definitions in eq. (3.19) and (3.20), the agent will in every period set $\frac{\partial\pi}{\partial K} = 0$. In practice a change in capacity will be approximated ex-post by using the value of $\frac{\partial\pi}{\partial K}$ found from solving $\max_q \Pi(q,\hat{K},\hat{\theta})$ ex-ante. By applying the Kuhn-Tucker first order necessary conditions to eq. (3.20) an *ex-post estimate* of expected change in profit for a change in capacity, dependent on current market conditions can be estimated for each period. The stochastic objective can be summarizes as follows:

The stochastic Objective:

$$\begin{aligned}
\max_K \Pi(K) &= \max_K E_\theta\left\{\max_q \Pi\left(q,\hat{K},\hat{\theta}\right)\right\} \\
\text{where } \Pi\left(q,\hat{K},\hat{\theta}\right) &= P\left(q,\hat{\theta}\right)q - C\left(q,\hat{\theta}\right) - \Psi\left(\hat{K}\right) \\
\text{Such that either (1)} &: 0 \leqslant q \leqslant \hat{K} \\
\text{or (2)} &: 0 \leqslant q \leqslant min(\hat{K},\hat{\theta})
\end{aligned} \tag{3.21}$$

where constraint (1) in eq. (3.21) is defined by eq. (3.6) and is valid for the random demand cases. Constraint (2) are given in eq. (3.7) and valid for the random inflow cases. As discussed in Section 3.2, this is due to the fact that the agent faces different types of uncertainty in the stochastic cases, and dependent on the type of uncertainty, the constraint for the production therefore varies. Since the production is influenced by uncertainty, the costs also vary dependent on the outcome of θ, hence the notation $C(q,\hat{\theta})$ in eq. (3.21).

Chapter 4

Method

In order to perform an experiment by appropriate tools, the theoretical framework need to be considered. The objective given in Section 1.1 is twofold:

1. Develop a simple and unified solution procedure for complex uncertainty problems.
2. Experiment to see how uncertainty influence a long term capacity decision.

In this chapter an adaptive learning scheme is developed, which creates a solution procedure sufficient for the above requirements. Based on the properties of the components in the method, the results will be in line with stochastic theory.

In Section 4.1 the main ideas of stochastic approximation are introduced. The online gradient method applied as optimization is presented, along with an alternative implementation to accommodate multiple learning subjects for the agent. The weight shifting parameter called step-size is introduce, along with its required properties.

In Section 4.2 the Kuhn-Tucker Conditions (KKT) for both deterministic and stochastic (random demand and random input) cases are derived. It is shown that the derived expression can be interpreted as an ex-post approximation for the derivative, and hence can replace the general gradient in the online gradient method when the numerical experiment is conducted.

In Section 4.3 the applied algorithms as pseudo code is given. The algorithms are implemented by non-linear programming in AMPL. Section 4.4 covers some basic AMPL syntax, and discusses some of the most significant commands. For a detailed presentation of the programing, please see Appendix A. The programs developed to conduct the experiments are given in Appendix B and C for the single and multiple learning objects respectively.

4.1 Stochastic Approximation

A stochastic approximation algorithm typically solves a problem of finding a root of an unknown function based on noisy measurements of the function. Robbins and Monro (1951) is a classic reference for recursive stochastic approximation, often called the RM-algorithm. Robbins and Monro showed that the method implied convergence in probability towards a consistent[1] estimator for a constant root of an unknown function (Lemma 1, p. 403).

Kiefer and Wolfowitz (1952) applied the RM-algorithm to estimate a stationary point for an unknown function, often called the KW-algorithm. To estimate a stationary point is to approximate the value of a variable for which the derivative of the unknown function is zero. The KW-algorithm was originally developed for a finite difference replacing the assumed unobservable derivative for the function of interest.

Following the RM- and KM-algorithm, many iterative methods have been developed. One such method is the gradient method. The gradient method can be seen as both an iterative optimization method and as a learning mechanism applied to uncover the true value of a learning subject by gradually adapting information from observations in the real world.

4.1.1 Online Gradient Method

The slope of a line describes the steepness, or rate of change, for that line. In the n-variable case, a gradient describes the steepness and direction of the slope. This vector function has given the name to the gradient method. The term "online" refers to how the estimate of the learning subject is approximated from observed values. The online gradient method approximates the value of the learning subject from values observed in the real world (Bottou, 2004, p. 155). In this thesis the online gradient method is used [2].

The online gradient method is an iterative first order method which starts of by an arbitrary initial guess for the target variable. Then the target variable is gradually increased (if maximization) or decreased (if minimization) by a fraction of the current observed gradient of the objective. The online gradient method is dependent on the nature of the gradient, applicable for both deterministic and stochastic problems.

Let $\{\delta_t\}$ be a sequence of positive fixed constants, called step-size, and let $\frac{\Delta \Pi_t}{\Delta K_t}$ be a gradient

[1] Consistent: The mean squared difference between actual and estimated root tend to zero.

[2] The literature discuss the difference between optimizing an empirical value versus an expected value. Vapnik (1999) derive and discuss asymptotic results. The online gradient method provide on average an approximation of the learning subject which converge towards the expected learning subject.

for the current objective. Also, let $\{K_t\}$ be a non-stationary Markov Chain for an arbitrary initial value K_0 and define the following relation:

$$K_{t+1} = K_t + \delta_t \frac{\Delta \Pi_t}{\Delta K_t} \tag{4.1}$$

The relation given in eq. (4.1) is the iterative step in the gradient method, and dependent on which case is considered, the gradient is either a deterministic or a stochastic value.

However, the main goal of the agent is to determine a level of capacity at which the profit is maximized, in other words he will determine a (unique) root for the profit function. For an arbitrary chosen value of initial capacity (K_0) the corresponding profit is observed. By performing the iteration in eq. (4.1) an approximation of the new level of capacity K_1 is determined, and the process is repeated.

The sequence of positive fixed constants (δ_t) must be defined in such a way that some standard conditions are satisfied in order for the Markov Chain defined by eq. (4.1) to converge. Before these properties are addressed in Section 4.1.3 it is convenient to set up the general online gradient method in the case of a maximization problem:

Online Gradient Method:

1. Initial guess: $K_t = K_0$
2. Repeat $K_{t+1} \leftarrow K_t + \delta_t \frac{\Delta \Pi_t}{\Delta K_t}$
3. Until $K_{t+1} - K_t$ is sufficiently small.
4. Claim $K_{t+1} \to K^*$ as t increases.

The gradient $\frac{\Delta \pi}{\Delta K}$ in step 2 can be an ordinary derivative $(\frac{\partial \pi}{\partial K})$ if available, or a proxy such as a point estimated difference. Either way, if

$$\lim_{t \to \infty} (K_{t+1} - K_t) \to 0$$

results from repeated iterations of eq. (4.1) in point 2 of the online gradient method, the method converge towards a long term capacity equal to a root of the profit function K^*, that is $\lim_{t\to\infty} K_{t+1} \to K^*$. A unified solution procedure for uncertainty problems based on the online gradient method requires an expression for the gradient valid for all types of uncertainty. As will be seen in Section 4.2 such a unified criterion can be derived based on the Kuhn-Tucker conditions. Before the unified criterion is developed, the inclusion of other learning objects for the agent will be described in Section 4.1.2.

4.1.2 Gradient Method & Learning

The online gradient method presented in Section 4.1.1 include only the capacity as a learning object for the agent, and hence must bee seen as an iterative optimization method. If, however, the agent holds an a-prior belief about price (p_0) and slope (dp_0) when choosing production, the solution algorithm can include simultaneous learning of the true price and slope, alongside the learning of best choice of capacity.

The incremental updating of capacity is done in the second step of the general online gradient method and the adaptive learning scheme as presented in Section 2.2 can be implemented at this step. To update the knowledge about the learning objects the adaptive learning model given in eq. (2.1) is included for each of the learning objects. The agent then updates his knowledge of both learning objects p_t and dp_t according to:

$$\begin{aligned} &\text{Learning Price} \quad p_{t+1} \leftarrow (1-\delta_t)p_t + \delta_t \hat{p}_t \\ &\text{Learning Slope} \quad dp_{t+1} \leftarrow (1-\delta_t)dp_t + \delta_t \hat{dp}_t \end{aligned} \tag{4.2}$$

where $\hat{p}_t$ is the realized price, p_t is the initial guess of price and p_{t+1} is the updated belief. Similarly, $\hat{dp}_t$ is the realized change in price, dp_t is the initial guess of slope, and dp_{t+1} is the updated belief the agent holds for next periods change in price. As before δ_t is a sequence of positive fixed constants, called step-size. Including a secondary learning object in the online gradient method will give the following iterative step:

Repeat:

- Eq. (4.1) for capacity: $K_{t+1} \leftarrow K_t + \delta_t \frac{\Delta \Pi_t}{\Delta K_t}$
- Eq. (4.2) for price & slope: $p_{t+1} \leftarrow (1-\delta_t)p_t + \delta_t \hat{p}_t$ and $dp_{t+1} \leftarrow (1-\delta_t)dp_t + \delta_t \hat{dp}_t$

Assume the initial guess for capacity $(K_t = K_0)$ and for the learning objects $(p_0 = p_0$ and $dp_t = dp_0)$ are given. The iterations are repeated until a sufficiently small difference between current and updated belief is reached. In Section 4.1.3 the standard properties of the step-size (δ_t) required to assure that learning may take place in every iteration are discussed.

4.1.3 Step-Size

The adaptive learning scheme defined as a moving average in eq. (4.2) should be designed such that new information has high weight in the start of the learning process, since the agent have limited information (and knowledge) at this point. As more information accumulates, and the agent's knowledge builds up, the last arriving information should have less weight.

The step-size is used in step 2 of the gradient method to iteratively develop the best choice of the learning objects by shifting weight between new and old information. In order to have this desired weight shifting effect, a step-size must be defined to meet certain properties. Standard assumptions for a step-size used in stochastic approximation is [3]:

$$\lim_{t\to\infty} \delta_t \to 0 \tag{4.3}$$

$$\sum_{t\to\infty} \delta_t = +\infty \tag{4.4}$$

$$\sum_{t\to\infty} \delta_t^2 < +\infty \tag{4.5}$$

The assumption in eq. (4.3) contributes to higher weight on already accumulated knowledge through diminishing weight on new information in eq. (4.1) and eq. (4.2). As more information on the correct capacity is compiled over the iterations, the assumption in eq. (4.3) contributes to a decreasing short term error in the prevailing long term capacity.

The second assumption, given in eq. (4.4), states that learning is unbounded over iterations. This implies a possibility to learn at every iterative step, since the step-size is always positive. The last property given in eq. (4.5) is a sufficient condition for convergence. Let the step-size be a dynamic parameter defined by:

$$\delta_{t+1} = \frac{1}{m + kt} \tag{4.6}$$

It can be shown that a step-size defined by eq. (4.6) will, for some positive constants m and k, have the desired properties and provide convergence between updated capacity and current best choice of capacity over time. A level which may, or may not, be equal to the free optimal capacity.

For constants $m = k = 1$ a step-size as defined by eq. (4.6) applied in the learning step as defined by eq. (4.2) will provide a long term learning object at a level equal to an empirical average, that is

$$p_{t+1}\frac{p_0+p_1+\ldots+p_t}{t+1} \quad \text{and} \quad dp_{t+1}\frac{dp_0+dp_1+\ldots+dp_t}{t+1}$$

after t iterations. From statistical theory the updated learning parameter will comply with Bayesian updating [4]. This contributes to the agent's subjective perception of best choice in such a way that all knowledge based on available observations is incorporated in to the choice made prior to the next period. Thus, the agent chooses a level of the learning such that his choice is in line with his perception.

[3] Robbins and Monro (1951, p. 401-403) develop the convergence properties needed for a positive sequence to converge. The same standard properties is applied in Flåm and Sandmark (2000, p. 5).

[4] Bayesian updating or learning refer to a subjective probability interpretation and essentially the application of Bayes's Theorem, discussed in Hogg and Tanis (2005, Ch. 7).

When the step-size, as defined by eq. (4.6), is applied in the gradient method one might show the convergence by taking the limit on both sides of the iterative step (point 2 of the gradient method). When doing so, the following results:

$$\lim_{t\to\infty} K_{t+1} \leftarrow \lim_{t\to\infty} \left[K_t + \delta_t \frac{\Delta\pi}{\Delta K}\right]$$
$$\lim_{t\to\infty} K_{t+1} \leftarrow \lim_{t\to\infty} K_t + \underbrace{\lim_{t\to\infty} \left[\delta_t \frac{\Delta\pi}{\Delta K}\right]}_{\delta_t \to 0}$$
$$\lim_{t\to\infty} K_{t+1} \leftarrow \lim_{t\to\infty} K_t$$

From the above it can be seen that there will be (at least) two reasons for the gradient method to converge. First the step-size parameter will approach zero as time increases, as is given by the first condition in eq. (4.3). Second, the gradient, denoted $\frac{\Delta\pi}{\Delta K}$ in the second step of the method may become zero, as it will if the first order condition is fulfilled for the current capacity level. For a sufficiently small change in capacity, i.e. if $\Delta K = K - K^* \to 0$ the following yields:[5]

$$\lim_{\Delta K\to 0} \frac{\Delta\pi}{\Delta K} \simeq \lim_{K\to K^*} \frac{\partial\pi}{\partial K} \to 0$$

It should be noted that convergence in the sense considered here, need not be convergence in terms of the most efficient, or best overall optimal sense (Robbins and Monro, 1951, p. 401). The term convergence in this case should be interpreted as finding a level of capacity where the change in capacity from one period to the next is sufficiently small, i.e. less than or equal to some predefined value.

Implemented in the gradient method, convergence is checked by evaluating the absolute difference between K_{t+1} and K^*, and is stopped if sufficiently small. Also note that the positive constants m and k in eq. (4.6) are chosen after a series of trials. This will be discussed when the results from the numerical experiments are presented in Chapter 5.

4.2 The Kuhn-Tucker Conditions

Linear programming is a solution method used to solve problems of optimizing a linear objective subject to linear constraints, i.e. both objective and constraints represents straight lines. If either of the objective or the constraints are of the non-linear type, the Kuhn-Tucker (KKT) conditions must be applied (Fourer et al., 2003, Ch. 18; Adams, 2003, p. 805-806).

[5] See Definition (5) given in Appendix D: $\frac{\partial\pi}{\partial K^*} = 0$ if K^* is a stationary point for $\pi(K)$.

From the objectives given by either eq. (3.17) or eq. (3.21), one observes that non-linearity will be the case, since the agent's objective is a continuous concave[6] (non-linear) profit function subject to linear inequality constraints, and hence non-linear methods must be applied.

In addition to Lagrange multipliers, called dual values, as in ordinary Lagrange method, the KKT conditions introduce a *complementary slackness* condition, and the *primal* and *dual* feasible condition. The presentation of the KKT conditions is limited in this thesis, for a thorough discussion one might consult Kuhn and Tucker (1951). The following definition for a dual value will be used:

Definition 3 *The multiplier λ_i, called dual value, denotes the price of a slack in constraint i, and gives the slope of the objective at current level of capacity: $\lambda_i = \frac{\partial \Pi(K)}{\partial K}$. (Fourer et al., 2003, p. 243)*

As the objectives are different in the different cases the KKT conditions must be calculated separately for each case. In Section 4.2.1 the KKT conditions for the deterministic case are presented, while in Section 4.2.2 the stochastic case is discussed. The goal is a unified solution procedure in any case, and based on Section 4.2.1 and 4.2.2 a unified long term break-even criterion can be determined, and it is presented in Section 4.2.3.

4.2.1 Deterministic KKT-Conditions

Consider the constrained deterministic objective given in eq. (3.17) for a given capacity $\hat{K}$. A solution to this problem is classified as optimal if the KKT-conditions, called the first order necessary conditions for non-linear programing are satisfied. To derive the first order necessary conditions for such a non-linear program start by defining the Lagrange function for the problem given in eq. (3.17) and introduce dual values λ_i for the constraint $0 \leqslant q \leqslant \hat{K}$:

$$\mathcal{L}\left(q, \hat{K}\right) = \pi(q, \hat{K}) + \lambda_1(\hat{K} - q) + \lambda_2 q \tag{4.7}$$

Maximize the Lagrange function given in eq. (4.7) by finding the first order condition (FOC) with respect to the control variable q:

$$\frac{\partial \mathcal{L}}{\partial q} = \frac{\partial \pi(q, \hat{K})}{\partial q} - \lambda_1 + \lambda_2 = 0$$

Introduce complementary slackness conditions (CSC) for the constraints:

$$\lambda_1(\hat{K} - q) = 0 \text{ and } \lambda_2 q = 0$$

[6] Appendix D give the definitions of continuous and concave by Definition 4 and 5, respectively.

Define the primal feasible condition (PFC) for the problem:

$$q \leqslant \hat{K} \text{ and } q \geqslant 0$$

Define the dual feasible conditions (DFC) for the constraints:

$$\lambda_1 \geqslant 0 \text{ and } \lambda_2 \geqslant 0$$

The first order necessary conditions given above define a feasible set $\Omega = \left\{q | q \leqslant \hat{K} \text{ and } q \geqslant 0\right\}$ in which an optimal solution to the problem in eq. (3.17) must be contained. For the above problem, let $\bar{q} = q(\hat{K})$ denote optimal quantity for a given capacity $\hat{K}$. Then, for appropriate dual values $\lambda_1, \lambda_2 \geqslant 0$, the KKT conditions for the Lagrangian in eq. (4.7) can be given more compactly by:

$$\frac{\partial \mathcal{L}(\bar{q}, \hat{K})}{\partial q} = \frac{\partial \pi(\bar{q}, \hat{K})}{\partial q} + \left\{ \begin{array}{rl} 0 & \text{if } 0 < \bar{q} < \hat{K} \\ -\lambda_1 & \text{if } 0 < \bar{q} = \hat{K} \\ \lambda_2 & \text{if } 0 = \bar{q} < \hat{K} \end{array} \right\} = 0 \tag{4.8}$$

where $\lambda_1(\hat{K} - \bar{q}) = 0$ and $\lambda_2 \bar{q} = 0$ are the complementary slackness conditions (CSC) which states that either a constraint is binding[7] or the corresponding dual value must be zero. Hence, if the capacity is binding then $\lambda_1 \geqslant 0$ for $\bar{q} = \hat{K} < K^*$, if K^* denotes the free optimal level of capacity. At optimum the constrained first order condition in eq. (4.8) is zero, and thus the marginal change in profit for a change in production is:

$$\frac{\partial \pi(\bar{q}, \hat{K})}{\partial q} = \left\{ \begin{array}{rl} 0 & \text{if } 0 < \bar{q} < \hat{K} \\ \lambda_1 & \text{if } 0 < \bar{q} = \hat{K} \\ -\lambda_2 & \text{if } 0 = \bar{q} < \hat{K} \end{array} \right.$$

To find the marginal change in profit for a change in capacity at current level of production one might use the envelope theorem. Let $\mathcal{L}^*(K)$ denote the maximal value of the Lagrangian when $\bar{q} = q(\hat{K})$ is the optimal production quantity subject to current capacity $\hat{K}$. Then the envelope theorem states that the rate of change in the maximal value of the Lagrangian subject to a change in capacity can be found by the partial derivative of the maximal value when holding production fixed at the maximal level $\bar{q}$. Hence, by the envelope theorem the marginal change in the Lagrangian for a change in capacity is:

$$\frac{\partial \mathcal{L}(\bar{q}, \hat{K})}{\partial \hat{K}} = \frac{\partial \pi(\bar{q}, \hat{K})}{\partial \hat{K}} + \lambda_1$$

[7] Definition of a binding constraint can be found in Sydsæter et al. (2002, p. 98)

and thus the marginal change in profit for a change in capacity is denoted by:

$$\frac{\partial \pi(\bar{q}, \hat{K})}{\partial K} = \lambda_1 - MC(\hat{K}) \tag{4.9}$$

Eq. (4.9) states that the optimal choice of capacity level is obtained when marginal cost of current capacity equals the associated dual value. Intuitively this is natural as it is the level at which the potential profit from an additional unit of production equates the cost of the last incremental unit of production capacity installed. It is also in line with fundamental economic theory. Thus, in the long term eq. (4.9) can be interpreted as a break-even criterion, or the target for the agent.

4.2.2 Stochastic KKT-Conditions

The KKT conditions for the stochastic case are divided into two separate cases as the uncertainty impose two different constraints on the objective according to eq. (3.21). As discussed in Section 3.5 a separation between objective ex-ante and ex-post is utilized in the short term decision. Under uncertainty both the marginal revenue and marginal cost of producing are expected values conditional regarding a specific outcome of the uncertainty.

First consider **the random demand case**. The objective given by eq. (3.19) is maximized ex-post subject to constraint (1) in eq. (3.21) valid for the random demand case: $0 \leqslant q \leqslant \hat{K}$. The first order necessary conditions for this non-linear problem are derived by defining the Lagrange function for the problem including dual values λ_i, one for each of the constraints:

$$\mathcal{L}\left(q, \hat{K}, \hat{s}; \lambda_i\right) = \Pi\left(q, \hat{K}, \hat{s}\right) + \lambda_1 \left(K - q\right) + \lambda_2 q \tag{4.10}$$

To maximize the Lagrangian in eq. (4.10) one follows the same approach as in the deterministic case by applying the KKT-conditions. For the ex-post objective given by eq. (3.19) subject to the constraint (1) in eq. (3.18), the following FOC, CSC, PFC and DFC conditions for the random demand case results:

$$\begin{aligned}
\text{FOC} &: \frac{\partial \mathcal{L}}{\partial q} = \frac{\partial \Pi\left(q, \hat{K}, \hat{s}\right)}{\partial q} - \lambda_1 + \lambda_2 = 0 \\
\text{CSC} &: \lambda_1 \left(\hat{K} - q\right) = 0 \text{ and } \lambda_2 q = 0 \\
\text{PFC} &: q \leqslant \hat{K} \text{ and } q \geqslant 0 \\
\text{DFC} &: \lambda_1 \geqslant 0 \text{ and } \lambda_2 \geqslant 0
\end{aligned}$$

Similar to the deterministic case the KKT conditions can be given compactly. Let $\hat{s}$ denote current realization of uncertainty under current capacity $\hat{K}$, and let $\bar{q} = q(\hat{K}, \hat{s})$ denote optimal quantity found when solving eq. (3.19) subject to $0 \leqslant q \leqslant \hat{K}$. Then, for a pair of suitable multipliers $\lambda_1, \lambda_2 \geqslant 0$, the Lagrangian will by the first order necessary condition be zero:

$$\frac{\partial \mathcal{L}(\bar{q}, K, s)}{\partial q} = \frac{\partial \Pi(\bar{q}, K, s)}{\partial q} + \begin{Bmatrix} 0 & \text{if } 0 < \bar{q} < \hat{K} \\ -\lambda_1 & \text{if } 0 < \bar{q} = \hat{K} \\ \lambda_1 & \text{if } 0 = \bar{q} < \hat{K} \end{Bmatrix} = 0 \tag{4.11}$$

At the optimum eq. (4.11) yields a marginal change in profit for a change in production under current outcome of uncertainty given by:

$$\frac{\partial \Pi(\bar{q}, \hat{K}, \hat{s})}{\partial q} = \begin{cases} 0 & \text{if } 0 < \bar{q} < \hat{K} \\ \lambda_1 & \text{if } 0 < \bar{q} = \hat{K} \\ -\lambda_2 & \text{if } 0 = \bar{q} < \hat{K} \end{cases} \tag{4.12}$$

where $\lambda_1(\hat{K} - \bar{q}) = 0$ and $\lambda_2 \bar{q} = 0$ are the CSC from the KKT conditions. The marginal change in the Lagrangian for a change in capacity, under current realization of random demand, is by the envelope theorem thus given by:

$$\frac{\partial \mathcal{L}(\bar{q}, \hat{K}, \hat{s})}{\partial K} = \frac{\partial \Pi(\bar{q}, \hat{K}, \hat{s})}{\partial K} + \lambda_1$$

And hence, the marginal change in profit for a change in capacity, under current realized random demand, is:

$$\frac{\partial \Pi(\bar{q}, \hat{K}, \hat{s})}{\partial K} = \lambda_1 - MC(\hat{K}) \tag{4.13}$$

As the agent maximizes short term production ex-post, i.e. when current random demand is known, the deterministic and random demand cases are essentially equal problems in the short term, hence the similarity in the break-even criteria's given by eq. (4.9) and eq. (4.13), despite the stochastic objective. This leads to the analog stochastic interpretation that the long term target is to install a level of capacity at which the *expected* marginal profit of production equals the cost of the last incremental unit of capacity installed.

Next consider **the random input case**. The objective in the random input case is, as in the random demand case, given by eq. (3.19), and maximized ex-post. Since the uncertainty now enters the problem trough the constraint, constraint (2) in eq. (3.18) are implemented. The Lagrangian for the random input case including dual values for the constraint $0 \leqslant q \leqslant min(\hat{K}, \hat{R})$ is thus:

$$\mathcal{L}(q; \lambda_i) = \Pi\left(q, \hat{K}, \hat{R}\right) + \lambda_1 \left(\hat{K} - q\right) + \lambda_2 \left(\hat{R} - q\right) + \lambda_3 q$$

By applying the KKT-conditions to the objective ex-post, the FOC, CSC, PFC and DFC conditions which results for the random input case is given by:

$$\begin{aligned}
\text{FOC} : \frac{\partial \mathcal{L}}{\partial q} &= \frac{\partial \Pi}{\partial q} - \lambda_1 - \lambda_2 + \lambda_3 = 0 \\
\text{CSC} : \lambda_1 \left(\hat{K} - q\right) &= 0 \text{ , } \lambda_2 \left(\hat{R} - q\right) = 0 \text{ and } \lambda_3 q = 0 \\
\text{PFC} : q &\leqslant \hat{K} \text{ , } q \leqslant \hat{R} \text{ and } q \geqslant 0 \\
\text{DFC} : \lambda_1 &\geqslant 0 \text{ , } \lambda_2 \geqslant 0 \text{ and } \lambda_3 \geqslant 0
\end{aligned}$$

The last two CSC's state that if current capacity is binding, i.e. if $q \leqslant \hat{K} < \hat{R}$ then the dual value of the capacity constraint is positive and the dual value of the random factor is zero, thus $\lambda_1 > 0$ and $\lambda_2 = 0$. If, however, the random input is the binding constraint, i.e. if $q \leqslant \hat{R} < \hat{K}$, then the opposite relation emerges, thus $\lambda_1 = 0$ and $\lambda_2 > 0$.

The KKT conditions can, as in the other cases, be given more compactly, thus let $\hat{R}$ denote the current realization of uncertainty under current capacity $\hat{K}$, let $\bar{q} = q(\hat{K}, \hat{R})$ denote the optimal quantity found when solving eq. (3.19) subject to the constraint $0 \leqslant q \leqslant min(\hat{K}, \hat{R})$. Then, following the KKT conditions for a triplet of suitable multipliers $\lambda_1, \lambda_2, \lambda_3 \geq 0$, the Lagrangian will by the first order necessary condition be equal to zero:

$$\frac{\partial \mathcal{L}(\tilde{q}, K, R)}{\partial q} = \frac{\partial \Pi(\tilde{q}, K, R)}{\partial q} + \left\{ \begin{array}{rl} 0 & \text{if } 0 < \bar{q} < \hat{K} < \hat{R} \\ -\lambda_1 & \text{if } 0 < \bar{q} = \hat{K} < \hat{R} \\ -\lambda_2 & \text{if } 0 < \bar{q} = \hat{R} < \hat{K} \\ \lambda_3 & \text{if } 0 = \bar{q} < \hat{K} < \hat{R} \end{array} \right\} = 0$$

Which at the optimum yields the following rate of change in profit for a change in production:

$$\frac{\partial \Pi(\bar{q}, \hat{K}, \hat{R})}{\partial q} = \left\{ \begin{array}{rl} 0 & \text{if } 0 < \bar{q} < \hat{K} < \hat{R} \\ \lambda_1 & \text{if } 0 < \bar{q} = \hat{K} < \hat{R} \\ \lambda_2 & \text{if } 0 < \bar{q} = \hat{R} < \hat{K} \\ -\lambda_3 & \text{if } 0 = \bar{q} < \hat{K} < \hat{R} \end{array} \right. \tag{4.14}$$

where $\lambda_1(\hat{K} - \bar{q}) = 0$, $\lambda_2(\hat{R} - \bar{q}) = 0$ and $\lambda_3 \bar{q} = 0$ are the familiar CSC's from the KKT conditions. The envelope theorem gives the marginal expected change in the Lagrangian under current realization of uncertainty for a change in capacity by:

$$\frac{\partial \mathcal{L}(\bar{q}, \hat{K}, \hat{R})}{\partial K} = \frac{\partial \Pi(\bar{q}, \hat{K}, \hat{R})}{\partial K} + \lambda_1$$

Hence, marginal changes in profit for a change in capacity under current inflow is:

$$\frac{\partial \Pi(\bar{q}, \hat{K}, \hat{R})}{\partial K} = \lambda_1 - MC(\hat{K}) \tag{4.15}$$

Again, the random inflow break-even criterion given by eq. (4.15) is similar to the deterministic criterion in eq. (4.9). Note, however, that the objectives naturally are different since the stochastic cases maximize a conditional objective ex-post (Π), while the deterministic case maximize the certain objective (π). Since the dual value λ_1 in both eq. (4.13) and eq. (4.15) is dependent on one specific outcome of a stochastic variable ($\hat{s}$ or $\hat{R}$), the dual value must be considered a stochastic variable.

Let $E_{\hat{\theta}}MR(q)$ denote expected marginal revenue and let $E_{\hat{\theta}}MC(q)$ denote expected marginal cost of production. Then, at optimum the change in marginal profit for a change in production given by eq. (4.12) or eq. (4.14) can be rewritten in terms of a sum of expected marginal revenue and costs. For the stochastic cases the marginal profit for a change in production then is generally denoted by:

$$\frac{\partial \Pi(\bar{q}, \hat{K}, \hat{\theta})}{\partial q} = E_{\hat{\theta}}MR(q) + E_{\hat{\theta}}MC(q)$$

From which it is clear that the properties of the random factor may influence the level of long term capacity choice in the stochastic cases, since one might have $MR(q) \neq E_{\hat{\theta}}MR(q)$ or $MC(q) \neq E_{\hat{\theta}}MC(q)$ or both. The magnitude and direction of the effect of uncertainty, compared to the deterministic optimal choice of capacity, will thus be determined by how the expected marginal revenue and expected marginal costs are influenced by the random factor.

4.2.3 The Long Term Criterion

Intuitively the agent should build up the capacity only if he expects an increase in future profit as a result of the increased capacity. Hence, when the ex-post dual value of capacity is higher than marginal cost of capacity, an increase in potential profit is expected, and the agent should install additional units of capacity. If the opposite situation occurs, the agent should demolish some of the installed capacity as the expected marginal change in profit for an additional unit of capacity then will be negative.

In the above sections, the KKT-conditions were presented for all cases considered in this thesis. By the envelope theorem the marginal change in profit for a change in capacity is given by eq. (4.9), eq. (4.13) and eq. (4.15) for each case respectively. Despite the differences in objectives and constraints, the equations forms a unified long term criterion for the capacity development process in all cases, and is summarized in eq. (4.16).

$$\frac{\Delta\pi}{\Delta K} := \lambda_1 - MC(\hat{K}) = \begin{cases} \frac{\partial\pi(\bar{q},\hat{K})}{\partial K} & \text{Deterministic} \\ \frac{\partial\Pi(\bar{q},\hat{K},\hat{s})}{\partial K} & \text{Random Demand} \\ \frac{\partial\Pi(\bar{q},\hat{K},\hat{R})}{\partial K} & \text{Random Input} \end{cases} \tag{4.16}$$

If the dual value of capacity is less than marginal cost of capacity, the long term criterion is negative, and otherwise positive. The long term criterion defined by eq. (4.16) thereby follows economic intuition in all cases. A unified solution procedure is thus achieved, and hence, eq. (4.16) replaces the gradient in the (deterministic or stochastic) online gradient method when the experiments are conducted.

4.3 Algorithms

As described in Section 4.1 the agent may have a single learning object (capacity) or multiple learning objects (capacity, price and slope). The number of learning objects does not alter the long term criterion given by eq. (4.16). However, different numbers of learning objects requires different iterative algorithms. Pseudo code for the single learning object (capacity) algorithm (*Algorithm 1*) is given below.

***Algorithm 1* - Algorithm for learning capacity:**

1. Initial guess: $K_t = K_0$
2. If stochastic: Draw $\theta_t := \hat{\theta}$ (If deterministic, go to Step 3).
3. Solve Objective:
   (a) Either solve the deterministic objective given by eq. (3.17).
   (b) Or solve the stochastic ex-ante objective given by eq. (3.19).
4. Repeat: $K_{t+1} := K_t + \delta_t \left[\lambda_K - MC(\hat{K})\right]$
5. Check if $K_{t+1} \in (\underline{K}, \overline{K})$:
   (a) If $K_{t+1} < \underline{K}$ then $K_{t+1} := \underline{K}$.
   (b) If $K_{t+1} > \overline{K}$ then $K_{t+1} := \overline{K}$.
6. Stop Criterion: (Assume $\epsilon :=$ numeric value.)
   (a) **Stop** if $|K_{t+1} - K_t| \leq \epsilon$.
   (b) Otherwise **Repeat** from Step 2 (Stochastic) or Step 3 (Deterministic)

For the agent, Step 1 in *Algorithm 1* is the only step where he more or less randomly chooses some initial capacity without any prior knowledge. In all subsequent periods, the available capacity is predetermined from previous iterations based on learning. In the deterministic case, all subsequent adjustments of capacity level is a consequence of the outcome in Step 4 or 5. For the stochastic cases, subsequent adjustments will also depend on the stochastic factor drawn at Step 3.

The algorithm for multiple learning objects includes some additional initial values, several iterative updates and one stop-if criterion for each of the learning objects. The complete algorithm for multiple learning objects (*Algorithm 2*) is given below.

***Algorithm* 2 - Algorithm for learning capacity, price and slope:**

1. Initial guess: $K_t = K_0$ (capacity), $p_t = p_0$ (price) and $dp_t = dp_0$ (slope)
2. If stochastic: Draw $\theta_t := \hat{\theta}$ (If deterministic, go to Step 3).
3. Solve Objective:
   (a) Either solve the deterministic objective given by eq. (3.17).
   (b) Or solve the stochastic ex-ante objective given by eq. (3.19).
4. Repeat:
   (a) Capacity: $K_{t+1} := K_t + \delta_t \left[\lambda_K - MC(\hat{K})\right]$
   (b) Price: $p_{t+1} \leftarrow (1-\delta_t)p_t + \delta_t \hat{p}_t$
   (c) Slope: $dp_{t+1} \leftarrow (1-\delta_t)dp_t + \delta_t \hat{dp}_t$
5. Check if $K_{t+1} \in (\underline{K}, \overline{K})$:
   (a) If $K_{t+1} < \underline{K}$ then $K_{t+1} := \underline{K}$.
   (b) If $K_{t+1} > \overline{K}$ then $K_{t+1} := \overline{K}$.
6. Stop Criteria: (Assume $\epsilon :=$ numeric value.)
   (a) **Stop** if: $|K_{t+1} - K_t| \leq \epsilon$ and $|p_{t+1} - p_t| \leq \epsilon$ and $|dp_{t+1} - dp_t| \leq \epsilon$.
   (b) Otherwise **Repeat** from Step 2 (Stochastic) or Step 3 (Deterministic)

Again, Step 1 in *Algorithm 2* is the only step where the agent makes any decisions more or less random, and without any prior knowledge. Naturally *Algorithm 2* is similar to *Algorithm 1*, except for the multiple updates and the multiple stop criteria in Step 4 and Step 6 respectively.

Step 5 in both *Algorithm 1* and *Algorithm 2* is essentially a projection of the new guess of capacity into a predefined allowed range. If the agent believes the bounds of the capacity is unbounded, Step 5 may be omitted. The overall results will not be affected by omitting this step. Either way, *Algorithm 1* is terminated if absolute change in capacity from one period to the next is smaller than some predefined threshold. Similarly, *Algorithm 2* is stopped if the absolute change for all learning objects is bounded by the predefined threshold.

The agent may have the needed skills to solve the problems analytically, or he may not. Either way, the proposed algorithms requires nothing more from the agent but to guess some initial values and choose the best possible production in each period, subject to current capacity, and keep on updating as described in either of the two algorithms.

4.4 AMPL

There are several computer programs designed to solve iterative problems. In this thesis AMPL (a Modeling Language for Mathematical Programming) is chosen due to an intuitive syntax and it's flexible data input. AMPL also facilitates for interactive dialog with predefined solvers to perform optimization of a non-linear objective subject to linear constraints (Fourer et al., 2003, Ch. 18). This section describes some basic AMPL syntax, and present the most significant commands applied in the AMPL programs.

An AMPL program consists of keywords with a predefined interpretation:

- `set T;` Defines a set of values named `T`.
- `param A;` Defines a (single valued) constant named `A`.
- `var q;` Defines a variable in the model, usually the control variable.
- `maximize profit;` Tell AMPL to maximize the objective named `profit`.
- `subject to Kconst;` Defines a constraint named `Kconst`

The model is solved iteratively, by conditional incremental adjustments of the capacity at the end of each period. To do so, the commands `for`, `break-continue` and `if-then-else` is implemented. For example is Step 6 of *Algorithm 2* programmed by:

```
for {t in T} {
   if n >= 2
    then if abs(K[n+1]-K[n]) <= stopIF
    and abs(phat[n+1]-phat[n]) <= stopIF
    and abs(dphat[n+1]-dphat[n]) <= stopIF
   then break;
   else continue }
```

The syntax `for {t in T} {...}` evaluates all statements contained in the body ({...}) for each member of the set `T`. To control the calculations and eventually terminate the for-loop, a logic condition in a `break-continue` command is used, together with an `if-then-else` syntax. The above code snippet will, for iteration 2 or higher, check the absolute difference between new and old belief for all the learning subjects. If it is small enough, i.e. smaller than the `stopIF` criterion, the `for`-loop is terminated by `break` or otherwise repeated.

As discussed in Section 1.2, uncertainty is represented by computer generated random values, usually based on a seed provided by the system. The seed can be predefined to assure reproducible results by including the following lines of code in the run-file[8]:

```
param l := 4;                  # Assign l a numeric value.
let X := l;                    # then assign the value to X.
let randseed[l] := X;          # and assign X to the random seed.
option randseed (X);           # Result is a predefined seed := 4.
```

The above code only sets a predefined seed, so in order to assure the pseudo-random numbers generated inside the for-loop to be different numbers in each iteration, the following lines is included.

```
reset data s_nr;               # Resets previous random number
let s[n] := s_nr;              # Assigns a new random number to s.
```

This section is concluded by a few words on the selection of CPLEX 11.2 as solver (Technologies, 2010). CPLEX handles both nonlinear and linear constraints, and it provides suffixes for analysis as a part of the solver code (Ch. 8). Also, CPLEX applies to continuous quadratic objective functions as assumed in this thesis (p. 30). By the line:

```
option solver cplex; option cplex_options 'sensitivity';
```

the CPLEX solver is selected, and both sensitivity and dual value suffixes is activated. Then dual values for a constraint can be accessed by adding the suffix `.dual` to the name of the constraint.

A more detailed presentation of the programing is given in Appendix A. The complete programs for the numerical experiments can be found in Appendix B and C for the single and multiple learning objects respectively.

[8] Code snippet is selected from Section B.2 of Appendix B. The applied seeds are listed in Table 5.1. The construction to set a seed is based on page 122 and 209 in the book by Fourer, Gay, and Kernighan (2003).

Chapter 5

Results

Based on the objectives given in Chapter 1 and the model presented in Chapter 3, a unified solution method for learning in both the deterministic and stochastic cases was developed in Chapter 4. Two algorithms were summarized in Section 4.3, one for the learning of optimal capacity (*Algorithm 1*) and one for multiple learning objects (*Algorithm 2*). The influence of uncertainty on the level of long term capacity is examined by applying numerical experiments, and both of the developed algorithms are tested. This chapter presents the results from these experiments.

To conduct the experiments, the parameters in the model have been assigned numerical values. These values are presented in Section 5.1. The different initial values for capacity, price, slope and the seed for the random number generator which have been implemented in the experiments is also presented in Section 5.1

In order to ease comparisons between the results for the stochastic cases and the unconstrained optimal solution, the analytical solution to the unconstrained deterministic problem is derived in Section 5.2. The results obtained by the single object learning algorithm (*Algorithm 1*), for both the deterministic and stochastic (random demand and random inflow) cases are presented in Section 5.3. In Section 5.4 the results derived using the multiple learning objects algorithm (*Algorithm 2*) are presented.

Tables of the numerical results (excerpts) obtained by applying *Algorithm 1* for the $Ex1$-combination are given in Appendix B. Similarly, Appendix C includes excerpts of the numerical $Ex1$-results obtained by applying *Algorithm 2* for each case. This chapter is concluded by Table 5.3 which summarizes the results from all the experiments performed by implementing both algorithms for all cases considered in this thesis.

5.1 Applied Initial Values and Data

As described in the introduction to this chapter, several combinations of initial values for price, slope and capacity have been used in the experiments. The random seeds also varies across the experiments when stochastic values are included. The different combinations of random seed and initial values are listed in Table 5.1.

	Matrix of initial value-combinations.			
Label:	$\hat{K}_0$	p_0	dp_0	seed
$Ex1$	0	50	1	1
$Ex2$	10	100	5	2
$Ex3$	20	150	12	3
$Ex4$	30	300	15	4
AMPL	`K[0]`	`phat[0]`	`dphat[0]`	`l`

Table 5.1: Initial values and seeds set for the random values applied in the experiments. The labels are used in the graphs, and indicates the combination of initial values applied to produce a particular result.

Each type of the stochastic cases have been simulated once for each of the four different sets of initial values in Table 5.1. The labels $Ex1$-$Ex4$ in Table 5.1 will also serve as a reference for the labels applied in the graphs (and tables) which is presented in this thesis, such that one particular label indicates the specific set of initial values which have been applied to obtain a particular result.

The functions in the model presented in Chapter 3 have been assigned numerical parameter values in order to conduct the numerical experiments. The numerical values assigned to the parameters are summarized in Table 5.2.

	Parameter values for the functions.						
parameter:	a	b	A	B	r	d	f
value:	150	2	6	0.5	0.15	0.10	3.75
function:	$P(q)$	$P(q)$	$C(q)$	$C(q)$	$\Psi(K)$	$\Psi(K)$	$\Psi(K)$

Table 5.2: The table summarize the numerical values assigned to the parameters in the model. These values are applied in all of the experiments reported in Chapter 5, and will not vary across the cases.

5.2 Benchmark Results

The unconstrained deterministic problem is somewhat trivial as long as the analytical expressions are known. Even so, to facilitate easy comparison to the outcome of the different numerical experiments for the various stochastic cases, the analytical result for the unconstrained certainty problem is summarized below.

To maximize the unconstrained deterministic problem, let demand be given by eq. (3.8) let unit production costs be defined by eq. (3.13) and let the cost of capacity be defined by eq. (3.15). Then the objective is given by a profit function such that the problem is:

$$\max_{K,q} \pi(q) = \max_{K,q} \Big\{ (a - bq)q - (Aq + Bq^2) - (r + d + f)K \Big\}$$

The problem is maximized with respect to both production q and capacity K, and the first order condition[1] for the profit is given by the total derivative set to zero.

$$0 = (a - 2bq) - (A + 2Bq) - (r + d + f) \tag{5.1}$$

Since the problem is unconstrained, let $q := K$ and solve eq. (5.1) for K to find the unconstrained optimal capacity (K^*). For the parameters values in Table 5.2 K^* is:

$$K^* = q^* = \frac{a - A - (r + d + f)}{2(b + B)} = \frac{150 - 6 - (0.15 + 0.10 + 3.75)}{2(2 + 0.5)} = 28 \tag{5.2}$$

Thus, a unconstrained deterministic optimal capacity and production equal to 28 gives a maximized profit of 1960. The numeric values have not any real interpretation or specific measurement of any kind. They merely serve as a benchmark to facilitate comparison, when assessing how uncertainty affects the agent's decision about long term capacity when he is faced with uncertainty of various types.

5.3 Results when Learning Capacity - *Algorithm 1*

The results obtained by *Algorithm 1* are presented in separate sections according to which case is considered. In Section 5.3.1 the results when the agent implements *Algorithm 1* for the deterministic case are presented. The results for both multiplicative and additive stochastic demand are presented in Section 5.3.2, while Section 5.3.3 reports the results for the random inflow case. The AMPL-programs used in these experiments are given as codes in Appendix B, seperately for each of the cases considered in this thesis.

[1] Definition (5) given in Appendix D: $\frac{\partial \pi}{\partial K^*} = 0$ if K^* is a stationary point for $\pi(K)$.

The results are mainly presented as graphs which shows the development in capacity during the learning process. Though, excerpts of the tables of the numerical results are given for the $Ex1$-experiments (Section B.4 of Appendix B). Table 5.3 summarize all the results for all experiments conducted in this thesis.

Recall that the agent updates his knowledge about the best choice of capacity according to Step 4 of *Algorithm 1* given in Section 4.3 when the general gradient is replaced by the unified long term criterion developed in Section 4.2, defined by eq. (4.16) for all cases.

4. Repeat: $K_{t+1} := K_t + \delta_t \left[\lambda_K - MC(\hat{K})\right]$

5.3.1 Learning Capacity under Certainty

The results from applying *Algorithm 1* in the deterministic case are presented in Figure 5.1 for different initial values of capacity, and are compared to the unconstrained optimal level of capacity found analytically by eq. (5.2). Figure 5.1 shows that the agent, by applying the developed method, learns a best choice of capacity equal to the unconstrained optimal level fairly quickly. The change in capacity is less than 5% already at iteration 12. The average long term capacity found by applying *Algorithm 1* is 28, with 100% capacity utilization. The step-size regime used in the trials shown in Figure 5.1 was $\delta_t = \frac{1}{10+t}$. This step-size gave rise to a 10% change in capacity from the first period to the next.

From Table 5.3 one finds that the number of iterations needed to obtain convergence is in the range of 29 to 40, dependent on the initial choice of capacity. From Figure 5.1 one observes that a small gap between the initial guess and the target requires less learning time before the target is reached, then is the case for a larger gap. Intuitively this is natural since the agent needs to learn less for a small discrepancy than if the discrepancy is high, and hence needs less time to acquire the needed knowledge. Due to this dependency, the speed of convergence can be increased (decreased) by a decrease (increase) in the length of the step-size. Tests show that a step-size of $\delta_t = \frac{1}{1+t}$ would require only 11 iterations to obtain convergence in this experiment.

A reduction in number of iterations required to learn the best long term capacity choice can potentially increase total profit in a business, if the full lifetime of a project is considered. As will be seen in the stochastic cases, this speedy learning rate is somewhat unrealistic in face of uncertainty. Even in the deterministic case for multiple learning objects, a significant increase in the learning rate is found, as will be seen in Section 5.4.1.

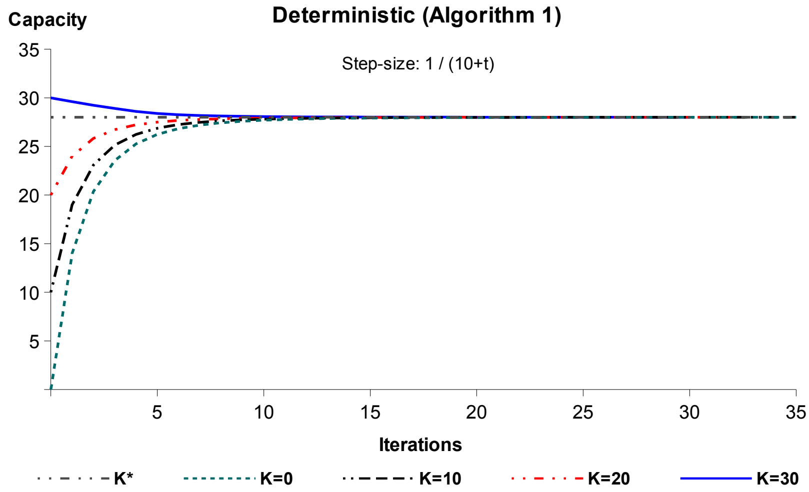


Figure 5.1: The graph shows the deterministic results obtained by *Algorithm 1* for the initial values of capacity in Table 5.1 versus K^*. Average long term capacity is 28, on average obtained at $t = 34$. Numeric results for $K_0 = 0$ is given in Table B.1. Table 5.3 summarizes the trials (DET *Alg.1*).

5.3.2 Learning Capacity under Random Demand

The random demand cases mimics a situation where the agent faces stochastic demand conditions, simulated by a random factor included in the demand function. As described in Section 3.3 both multiplicative and additive uncertainty is examined. The results are derived by running the appropriate AMPL-program in Appendix B.2 and will be summarized separately for each type.

When stochastic demand enters the problem, the agent maximizes the stochastic objective summarized in eq. (3.21) subject to the constraint given by eq. (3.6). For the additive stochastic demand case, called Type 1, the uncertainty is a zero mean random factor, that is $E(s) = 0$. For the multiplicative stochastic demand of Type 2, the random factors have a positive expectation defined by $E(s) = 1.5$. For the multiplicative stochastic demand of Type 3, the random factor has an expectation given by $E(s) = 1$.

Random Demand - Type 1

A zero mean random factor is assumed when the experiments for the additive random demand of Type 1, defined by eq. (3.10) is performed for all initial values in Table 5.1. The random factor is defined by eq. (3.1) for $E(s) = 0$. For a zero mean uncertainty, the outcome space is constant and symmetric, as shown in Figure 3.1 and intuitively one might expect convergence towards a long term capacity close to the free optimal capacity.

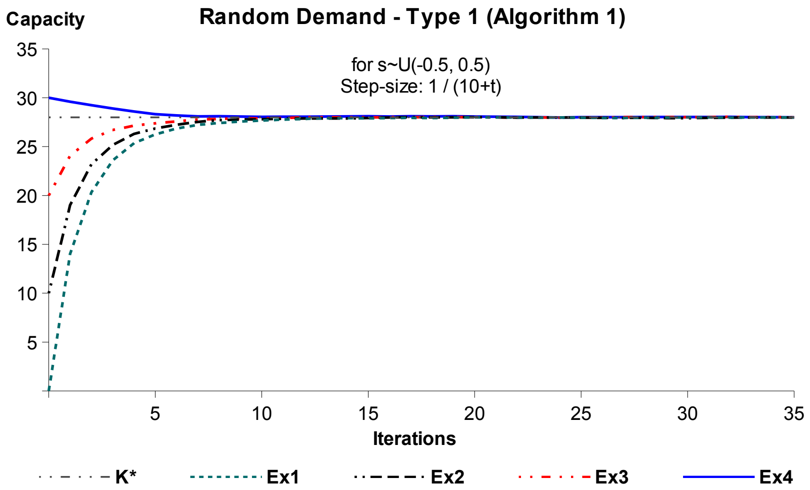


Figure 5.2: The graph shows the results for a Type 1 random demand case obtained by *Algorithm 1* for the initial value as given by Table 5.1 compared to K^*. Average long term capacity is 28, on average obtained at $t = 82$. Capacity utilization is close to 100%. Numeric results for $Ex1$ is given in Table B.2 and all trials are summarized in Table 5.3 (Rows: T1 *Alg.1*).

Figure 5.2 shows the results derived for the Type 1 experiments for a step-size set to $\frac{1}{10+t}$. It is clear from Figure 5.2 that the agent finds a long term level of capacity very close to the benchmark level. *Algorithm 1* requires on average 82 iterations for the agent to acquire this knowledge. The required number of iterations is thus higher than in the deterministic case, but the capacity development in Figure 5.2 has a close resemblance to the development in the deterministic case shown in Figure 5.1.

A zero mean noise has an impact on the price which is independent of selected production level. The empiric expectation found for the realized random values in this case is $E(s) = -0.0059$, and the results obtained by applying *Algorithm 1* for the Type 1 demand are therefore directly comparable to the findings in the certainty case since the empirical expectation is (close to) zero.

Random Demand - Type 2

For a multiplicative random demand defined by eq. (3.11) the random value is added to the slope in the demand function according to $P(q, s) = a - (b + s)q$. For a zero mean random factor, the Type 2 uncertainty creates a symmetric outcome space around the deterministic demand curve. However, the variation in price is proportional to the production level such that an increase in uncertainty for higher production results.

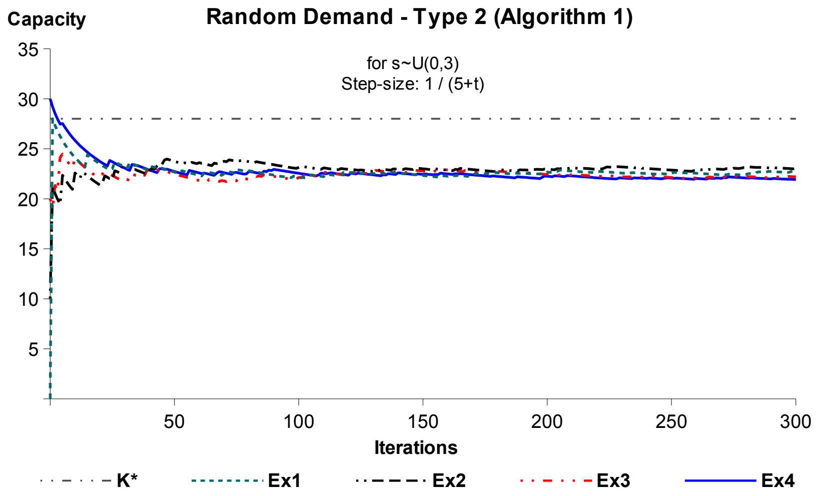


Figure 5.3: The graph shows the results for a Type 2 random demand case obtained by *Algorithm 1* for the initial values in Table 5.1 compared to K^* and $E(q|s)$. Average long term capacity is 22.3, on average obtained at $t = 468$. Capacity utilization is 82%. Numeric results for $Ex1$ is given in Table B.3 and all trials are summarized in Table 5.3 (Rows: T2 *Alg.1*).

If the expectation for the random factor is positive, the expected price is lower than the deterministic price since uncertainty enters the demand as a negative factor[2]. The long term capacity choice acquired by the method should reflect this downward bias. For a step-size given by $\frac{1}{5+t}$ and a random factor with $E(s) = 1.5$ it is from Figure 5.3 obvious that the convergence level is lower than K^* in all trials. *Algorithm 1* requires 468 iterations to reach a capacity level of 22.3 on average.

During the process, the capacity utilization is close to 82% with an average production of only 18.2 units. Based on this, one may conclude that the agent acquires the needed knowledge by applying the method to make a capacity choice which reflects the increased uncertainty for higher levels of production, and which is compliant with basic economic intuition without any prior knowledge about the market.

Random Demand - Type 3

A random demand of Type 3 is of a multiplicative type for which the uncertainty term is multiplied by the demand function according to eq. (3.12): $P(q, s) = (a - bq)s$. As in the previous case, the effect of uncertainty on demand depends on both expectation of the random factor and the chosen level of production. For a random factor as defined by eq.

[2] By eq. (3.1) for $\underline{s} := 0$ & $\overline{s} := 3$ then $E(s) = 1.5$ by eq. (3.3) and hence $E(P) < P$ (See Appendix D).

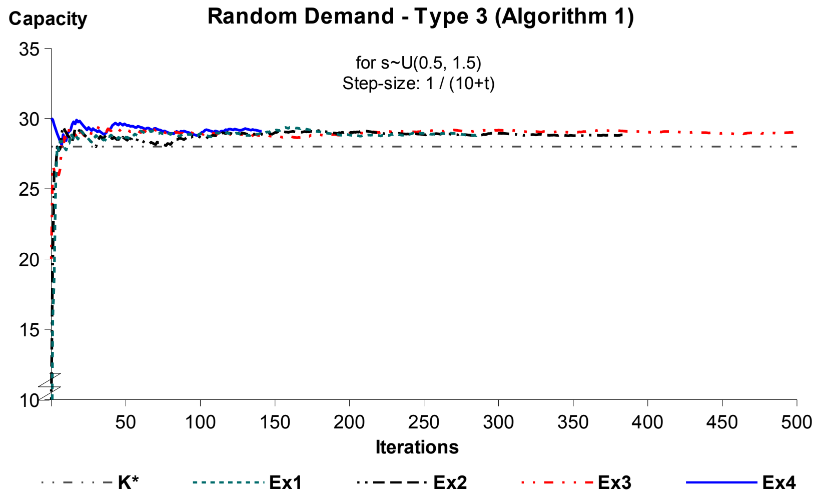


Figure 5.4: The graph shows the results for a Type 3 random demand case obtained by *Algorithm 1* for the initial values in Table 5.1 compared to K^*. Average long term capacity is 28.94 on average obtained at $t = 351$. Capacity utilization is close to 95.5%. Numeric results for $Ex1$ are given in Table B.4 and all trials are summarized in Table 5.3 (Rows: T3 *Alg.1*).

(3.1) with $E(s) = 1$ the expected price is equal to the sure price[3]. However, as discussed in Section 3.3 and as shown in Figure 3.2, a decreasing volatility in prices for an increasing production quantity results. Thus, a reduced variation in price for higher quantities might lead the agent to choose a production, and hence capacity, higher than previously found.

Figure 5.4 shows the results obtained by applying *Algorithm 1* when $E(s) = 1$ for the combination of initial values in Table 5.1, using a step-size set to $\frac{1}{10+t}$. From Figure 5.4 it can be seen that the agent consequently selects a higher level of long term capacity than in any of the previous experiments. The average long term capacity is 28.94, on average found at iteration 351. During the learning process, the average production is close to 27.7, such that the capacity utilization is slightly above 95% on average.

5.3.3 Learning Capacity under Random Inflow

In the random inflow case, the agent faces uncertain inflow of raw materials, such that the random factor may, or may not inflict a binding constraint on the short term choice of production. The implemented stochastic objective is summarized in eq. (3.21) in which the applied constraint, defined by eq. (3.7) is labeled as constraint (2). This section presents the results obtained by running the AMPL-program in Appendix B.3 for each of the initial value combinations given in Table 5.1.

[3] By eq. (3.1) for $\underline{s} := 0.5$ and $\overline{s} := 1.5$ then $E(s) = 1$ by eq. (3.3) and hence $E(P) = P$ (See Appendix D).

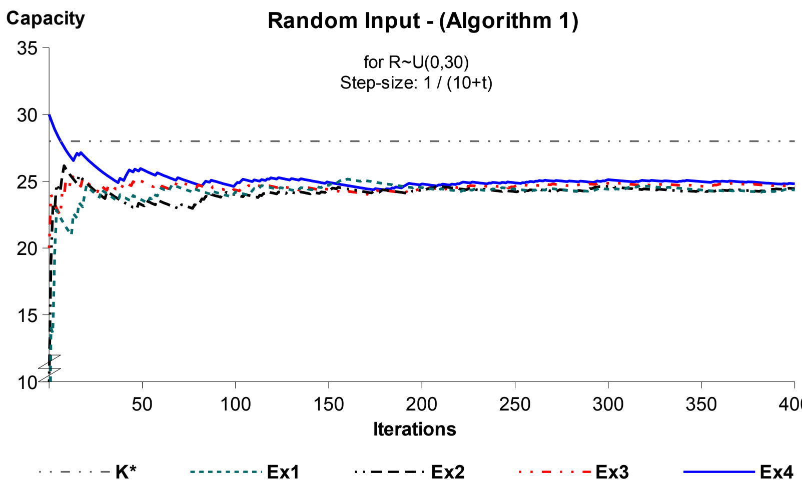


Figure 5.5: The graph shows the results for the random inflow case obtained by *Algorithm 1* for the initial values in Table 5.1 compared to K^*. Average long term capacity is 24.5 on average obtained at $t = 1991$. Capacity utilization is 59.5% and the input utilization is 96.5%. Numeric result for $Ex1$ is given in Table B.5 and summarized in Table 5.3 (Rows: RDI *Alg.1*).

The random factors follows a uniform distribution, defined by eq. (3.1) such that the expected outcome is $E(R) = 15$. Figure 5.5 displays the results obtained for the random inflow experiments. Average long term capacity is 24.5, thus approximately 87.5% of the free optimal level of capacity. The average production during the development process shown in Figure 5.4 is 14.6, which gives a capacity utilization as low as 60% for the four trials.

From Table 5.3 we find that the empirical expectation is $E(\hat{R}) = 15.14$ over the trials displayed in Figure 5.5. The input utilization is close to 97%, and hence only 3% of the inflows goes to waste. If the long term capacity found by the algorithm is close to the free optimal level, the long term criterion should be close to zero. In this case the gradient is -4 at the convergence step in all of the trials. At iteration 1991 the absolute value of the new information[4] is less than the defined `stopIF` criteria (set to 0.002) and the iterations is terminated. Convergence is thus obtained, mostly due to the definition of the step-size.

A long term capacity of 9.5 units above the expected level of inflow enables the agent to utilize approximately 63% of the inflows at levels above expected inflow. However, the probability of overcapacity is as high as 0.82 at this level[5]. In the experiments we find that overcapacity occurs in 81% of the iterations, consistent with the calculated probability.

[4] The absolute value of the new information is given by $|\delta_{1991}[\lambda_K - MC(\hat{K})]|$.

[5] $P(0 < R < 24.5) = \frac{24.5}{30} = 0.82$

The cost of maintaining a capacity of 24.5 units is 98, independent of actual inflows in each period. By inserting $E(R)$ for $\hat{K}$ in eq. (3.15) the cost of maintaining a capacity level equal to the expected inflow is 60, and the cost of overcapacity is thus 38. The probability of observing inflows between expected level and the selected long term level of capacity is 0.32 and expected additional profit is 124 for inflows in this range[6]. Hence, the expected additional profit more than outweighs the additional cost. For the range between the long term choice and free optimal level of capacity, expected additional profit is 3.57. Since the capacity cost increase by 14 over this interval, these units has a net negative contribution to the agent's profit[7].

5.4 Results when Learning Capacity, Price & Slope - *Algorithm 2*

When the agent applies *Algorithm 1* he presumably knows the demand function, and hence knows the price he can get for the produced quantity. In *Algorithm 2* the agent is by assumption unfamiliar with the actual demand function, and instead he holds a firm initial belief about the nature of it. Hence, unfamiliar with the actual demand in eq. (3.8) the agent optimizes his short term profit by solving a maximization problem for the actual demand replaced by a proxy demand curve.

$$\widetilde{P}(q) = p_t - \frac{dp_t}{2} q_t \tag{5.3}$$

For a proxy given by eq. (5.3) the agent *believe* his ex-post stochastic objective is

$$\max_q \Pi(q) = \max_q \left\{ \widetilde{P}(q)q - C(q) - \Psi(\hat{K}) \right\} \tag{5.4}$$

for an initial guess of capacity given by $\hat{K}$. The agent maximizes the *believed* profit function in eq. (5.4) subject to a capacity constraint given by either eq. (3.6) (deterministic and random demand cases) or eq. (3.7) (random inflow case). When multiple learning objects are implemented in the experiments, the perceived problem in eq. (5.4) thereby replaces the agent's original problem given in either eq. (3.17) (deterministic case) or eq. (3.21) (stochastic cases). By following the same approach as described in Section 4.2 the marginal change in the agent's *perceived* profit for a change in capacity can be found by the envelope theorem. The perceived marginal change in profit for a change in capacity is thus

$$\frac{\partial \Pi(\bar{q}, \hat{K})}{\partial K} = \lambda_K - MC(\hat{K}) \tag{5.5}$$

[6] $E[\Delta\Pi(15 < R < 24.5)] = P(15 < R < 24.5)[\Pi(24.5) - \Pi(15)] = \frac{24.5-15}{30}[391.88] = 124.29$

[7] $E[\Delta\Pi(24.5 < R < 28)] = P(24.5 < R < 28)[\Pi(28) - \Pi(24.5)] = \frac{28-24.5}{30}[30.63] = 3.57$

when holding production fixed at the maximal level $\bar{q}$. The proxy first order criteria for the agent's perceived problem in eq. (5.5) equals the long term break-even criteria in eq. (4.16) and hence, the developed algorithms in Chapter 4 applies without modifications. Recall that the agent updates his beliefs according to Step 4 in *Algorithm 2*.

4. Repeat:
   (a) Capacity: $K_{t+1} := K_t + \delta_t \left[\lambda_K - MC(\hat{K})\right]$
   (b) Price: $p_{t+1} := (1-\delta_t)p_t + \delta_t \hat{p}_t$
   (c) Slope: $dp_{t+1} := (1-\delta_t)dp_t + \delta_t \hat{dp}_t$

Step 4.a and 4.b follows the adaptive learning scheme as described in Section 2.2. By rewriting the average function constructed for learning in eq. (2.1) to $p_{t+1} \leftarrow p_t + \delta_t(\hat{p}_t - p_t)$ it is evident that future expected price is a projection of current best guess (p_t) and an error-adjustment term accounting for deviation between belief and observation $(\hat{p}_t - p_t)$. This learning heuristic also applies to learning of correct slope.

The observed price is given by the demand curve valid in each case. In the deterministic and random inflow cases eq. (3.8) applies, such that $\hat{p}_t = a - bq_t$. For the random demand cases the observed prices results from applying eq. (3.10), eq. (3.10) or eq. (3.10) respectively. In the random inflow case and in the deterministic case, the observed slope is $\hat{dp}_t = b$. Similarly, the observed slope is case dependant in the random demand cases. In Section 5.4.1 results for the deterministic case are presented, while the random demand results are presented in Section 5.4.2. In Section 5.4.3 the results from implementing *Algorithm 2* in the random inflow case are discussed.

5.4.1 Learning Capacity, Price and Slope under Certainty

The results acquired by applying *Algorithm 2* in the deterministic case are presented in Figure 5.6, compared to the free optimal level of capacity (K^*) given by eq. (5.2). These results are obtained by running the run-file in Section B.1 modified with the changes described in Section C.3 for the initial values of capacity, price and slope as given in Table 5.1 using data from Table 5.2.

From Figure 5.6 it is clear that the agent obtains knowledge of a long term capacity very close to the benchmark level by applying *Algorithm 2*, however the number of iterations increases significantly compared to *Algorithm 1*. In the $Ex1$-experiment, the agent adapts his knowledge over 759 iterations. From an initial perceived price of 50, the agent learns the price to be 93.85, close to the actual price (94) for capacity and production equal to 28.

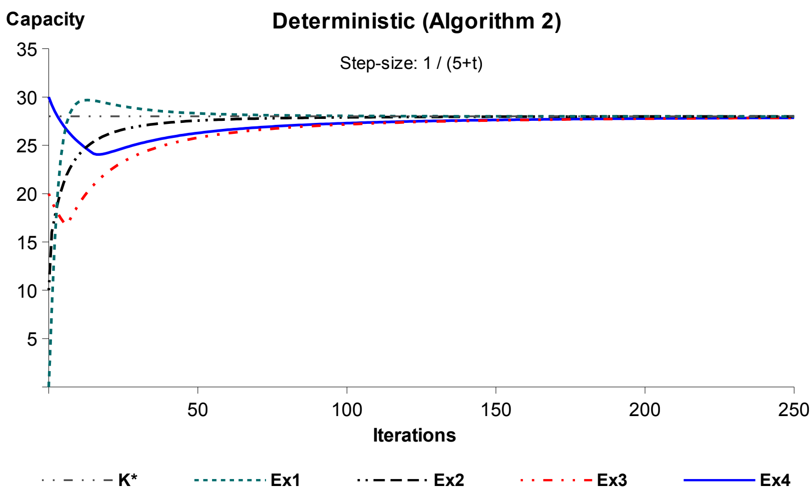


Figure 5.6: The graph shows the results for the deterministic case obtained by *Algorithm 2* for the initial values in Table 5.1 compared to K^*. Average long term capacity is 28 on average obtained at $t = 1773$. Capacity utilization is $\approx 100\%$. Numeric result for $Ex1$ is given in Table C.1 and summarized in Table 5.3 (Rows: DET *Alg.2*).

Notice that the development pattern of the capacity during the learning process is different in Figure 5.6 than what was seen in the experiments reported in Figure 5.1. Since the perceived nature of demand differs from the true nature of it, the information obtained by the agent is slightly different than in the previous experiments. When the difference between believed and observed values decrease over the learning process, the building pattern becomes more similar to the pattern found in Figure 5.1. This issue is discussed in Section 6.2 since the difference in perceived and observed marginal revenue affects the long term adaption, dependant on the type of uncertainty which is considered.

In this experiment, the marginal change in profit for a change in capacity is found to be 0.003, and thus the long term criteria is close to zero, and one may conclude that the agent choose a long term capacity very close to the efficient level. From Table C.1 and the summary in Table 5.3 one also see that the agent learns both price and slope with high accuracy compared to the deterministic values. How will the agent adjust his capacity when *Algorithm 2* is applied in the random demand cases?

5.4.2 Learning Capacity, Price and Slope under Random Demand

This section presents the results obtained by implementing *Algorithm 2* for the stochastic demand cases, when the agent is presumably unfamiliar with the original demand curve, as discussed in Section 5.4.1. Instead the agent maximizes profit for a proxy demand curve given by eq. (5.3) $\left(\widetilde{P}(q) = p_t + \frac{dp_t}{2} q_t\right)$. Learning occurs according to Step 4 in Algorithm 2, and for the random demand cases the observed price $\hat{p}_t$ and slope $\hat{dp}_t$ are set according to the specific functions applied in the previous random demand experiments. That is:

- Type 1: Let $\hat{p}_t := a - bq_t + s_t$ and let $\hat{dp}_t := b$, for $s_t \sim \mathcal{U}(-1, 1)$.
- Type 2: Let $\hat{p}_t := a - (b + s_t)q_t$ and let $\hat{dp}_t := b + s_t$, for $s_t \sim \mathcal{U}(0, 3)$.
- Type 3: Let $\hat{p}_t := (a - bq_t)s_t$ and let $\hat{dp}_t := b \cdot s_t$, for $s_t \sim \mathcal{U}(0.5, 1.5)$.

The results are derived by running the type-specific AMPL program in Appendix C for the data in Table 5.2 and for the initial values in Table 5.1. The agent learns price, slope and capacity when using *Algorithm 1*, although the long term capacity in some cases differs from the results obtained by applying *Algorithm 1*.

Random Demand - Type 1

As discussed in the introduction to this section, the observed demand is given by eq. (3.10) in this case assuming a zero mean random factor with a symmetric outcome space as shown in Figure 3.1 and the risk is from the agent's perspective perceived as constant over the entire range of possible production choices.

Figure 5.7 displays the capacity development in the random demand Type 1 case. Again one notice the similarity to the deterministic analogue experiment in Figure 5.6. No matter what initial price, slope or capacity the agent starts off with, convergence is on average obtained at the 1116^{th} iteration for a long term capacity equal to 28. The most significant difference compared to applying *Algorithm 1* is the increased number of iterations required.

The agent's initial beliefs about price spans from 50 to 300 over the four experiments. When convergence is obtained, the agent's belief has adjusted such that the price is perceived to be in the range from 93.34 ($Ex1$) to 95.23 ($Ex4$), close to the actual price of 94 obtained for a production of 28 units. Similarly the agent has a prior belief about slope varying over the range from 1 to 15. By the end of the learning process, the perceived slope is in the range from 1.97 ($Ex1$) to 2.04 ($Ex4$). The actual slope is $b = 2$, and again a good approximation to the actual market conditions surrounding the agent results from the learning process.

Based on Table 5.3 the empirical average of the random factors is 0.0029, thus the random values reflects a zero mean noise. On average the long term criteria in eq. (4.16) is close to zero $\left(\frac{\Delta\Pi}{\Delta K^*} = 0.0992\right)$ for the four trial, which confirms that the long term capacity is close to the efficient level. Thus, despite the wide range of initial beliefs, the agent acquires the needed knowledge by applying *Algorithm 2* to be optimally adjusted in terms of price, slope and capacity when the learning process is complete.

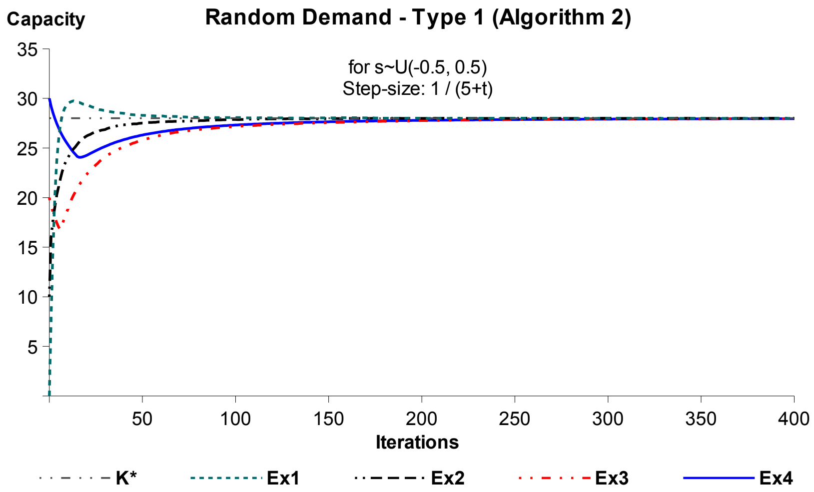


Figure 5.7: The graph shows the results for the random demand Type 1 case obtained by *Algorithm 2* for the initial values in Table 5.1 compared to K^*. Average long term capacity is 28 on average obtained at $t = 669$. Capacity utilization is 99%. Numeric result for $Ex1$ is given in Table C.2 and summarized in Table 5.3 (Rows: T1 *Alg.2*).

Random Demand - Type 2

Observed demand is, for stochastic demand of Type 2, given by eq. (3.11) for an uncertainty factor with positive expectation, thus $E(s) = 1.5$. As discussed in Section 5.3.2 this result in a downward bias on the production since the risk perceived by the agent is increasing for an increasing production, and thereby may result in a reduced long term capacity.

Figure 5.8 displays the results for *Algorithm 2* for a random demand Type 2 uncertainty, from which it is obvious that the long term capacity is below the benchmark level. This is the same as found by *Algorithm 1*, although the level of the long term capacity differs slightly between the algorithms. The average long term capacity over the four trials is 17.5 versus 22.3 found by *Algorithm 1*. On average 1041 iterative steps are required to obtain convergence, more than twice as many iterations are applied to obtain convergence when learning capacity, price and slope simultaneously, as compared to *Algorithm 1*.

When the learning process ends, the perceived price is in the range $88.56 - 89.59$ for a slope in the range $3.48 - 3.57$ over the four trials. Based on Table 5.3 the average random factor is 1.51. Expected price given by eq. (3.11) is 88.78 for a production of 17.46 units when $E(\hat{s}) = 1.51$, and thus the agent adjusts his knowledge into a narrow, and quite accurate estimate for both price and slope. However, note that the capacity utilization increases from 82% in *Algorithm 1* to 99.8% in *Algorithm 2*.

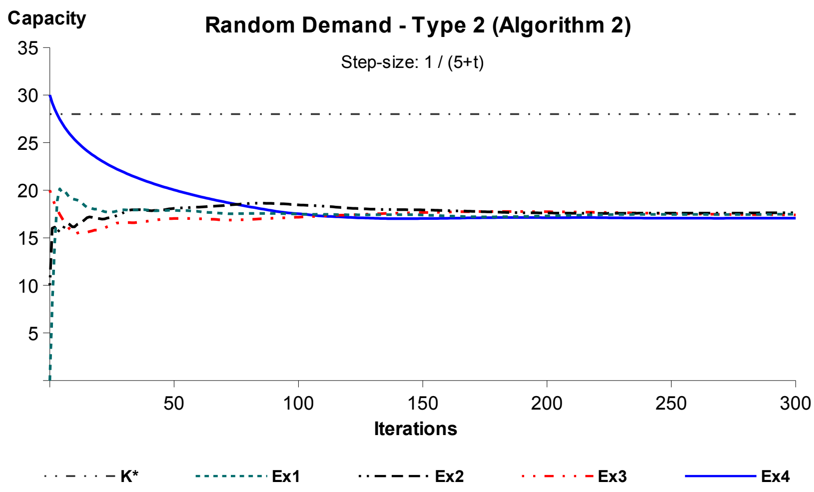


Figure 5.8: The graph shows the results for the random demand Type 2 case obtained by *Algorithm 2* for the initial values in Table 5.1 compared to K^*. Average long term capacity is 17.46 on averagee obtained at $t = 1041$ with an average capacity utilization close to 99.8%. Numeric result for $Ex1$ is given in Table C.3 and summarized in Table 5.3 (Rows: T2 *Alg.2*).

Since the random demand Type 2 uncertainty is such that the slope of the actual demand is affected by the uncertainty, the use of a proxy affects the long term choice made by the agent. The agent does not face the uncertainty directly through the price when applying a proxy, and hence adapts differently in the long run. These characteristics are discussed in Section 6.2.

Random Demand - Type 3

This section presents the agent's choice of long term capacity when the observed price is of the random demand Type 3, defined by eq. (3.12). Assume $E(s) = 1$, such that the agent faces a decreasing volatility in prices for an increase in production, as shown in Figure 3.2. According to the results in Section 5.3.2 the agent installed a long term level of capacity higher than the unconstrained optimal level when *Algorithm 1* was applied.

Figure 5.9 shows how the the agent's perception about price and slope affects the expected positive bias for higher production levels. The average long term capacity is 28.03 for a Type 3 uncertainty when using *Algorithm 2*. This long term choice differs only slightly from the optimal level found analytically by eq. (5.2) and it is 0.9 units less than previously found. At the end of the learning process the agent has obtained a capacity utilization of 99.6% on average, approximately 4.2% higher than in the previous Type 3 experiment. On average convergence is obtained at iteration number 1091.

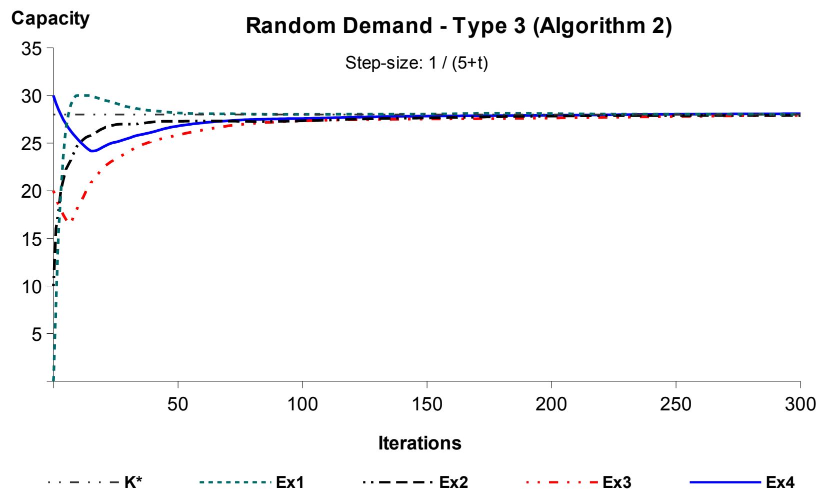


Figure 5.9: The graph shows the results for the random demand Type 3 case obtained by *Algorithm 2* for the initial values in Table 5.1 compared to K^*. Average long term capacity is 28.03 on average obtained at $t = 1091$ with an average capacity utilization close to 99.6%. Numeric result for $Ex1$ is given in Table C.4 and summarized in Table 5.3 (Rows: T3 *Alg.2*).

The agent's knowledge about price has been narrowed from the wide range of initial values in Table 5.2 into a perceived range between 93.33 and 95.68. Again we see that the agent learns a price which is close to the actual price for an efficient choice of capacity and production. The agent learns the slope to be in the range $1.99 - 2.08$ over the four trials. The average gradient is close to zero (0.337), and hence the agent has adjusted his knowledge by applying *Algorithm 2* such that an efficient long term capacity is selected, and a quite accurate estimate for both price and slope is obtained.

5.4.3 Learning Capacity, Price and Slope under Random Inflow

In the random inflow case the agent maximizes the *perceived* objective in eq. (5.4) subject to the chance constraint in eq. (3.7). This section presents the results obtained from applying *Algorithm 2* in the random input case, derived by running the appropriate AMPL-program in Appendix C for the data in Table 5.2 and the initial values in Table 5.1. Similar to *Algorithm 1*, the random inflow is a uniform random variable defined according to eq. (3.1) on the support $[0, 30]$, thus the expected inflow is $E(R) = 15$ by eq. (3.3).

Figure 5.10 displays the random inflow results when using *Algorithm 2*. Average long term capacity is 26.5, close to 95% of the free optimal level, an increase of 7% from the previous experiment. An increase in capacity results in a decrease in capacity utilization since the inflows are almost equal. Thus, the capacity utilization is only 56% versus 60% previously,

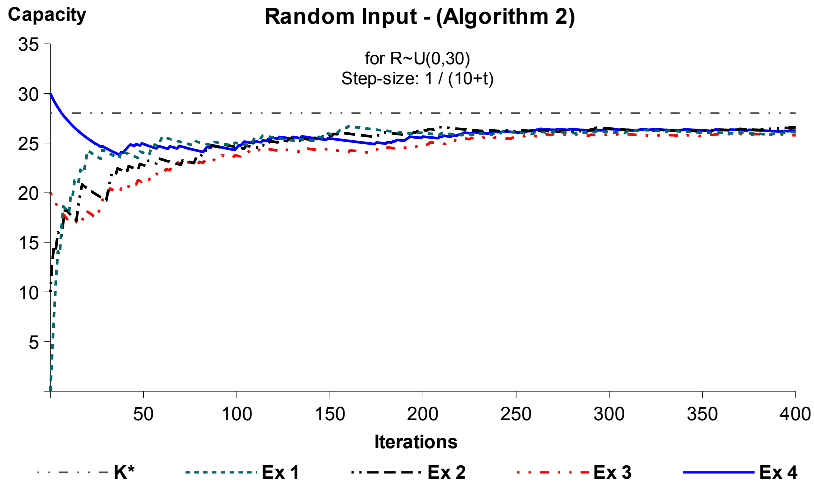


Figure 5.10: The graph shows the results for the random inflow case obtained by *Algorithm 2* for initial values in Table 5.1. Average long term capacity is 26.56 on average obtained at $t = 1993$. Average capacity utilization is 56% with 98% input utilization. Numeric result for $Ex1$ is given in Table C.4 and summarized in Table 5.3 (Rows: RDI *Alg.2*).

the lowest utilization in all the cases. However, a capacity higher than expected inflow enables the agent to utilize more of the inflows, and a capacity of 26.5 enables a utilization of 77% of the inflows above the expected level.

The initial values applied in *Algorithm 2* (Table 5.1) spans a wide range. During the learning process is the agent's perception about price and slope compressed such that the perceived price is in the range from 119.9 to 120.8, and slope is on average $dp = 2.03$. Both are close to what would be obtained by deterministic demand in eq. (3.8) for the same production. Again a good approximation of the market conditions results from the learning process.

When using a proxy demand function, the perceived profit is given by eq. (5.4) as opposed to the actual profit defined by eq. (3.21). By eq. (5.4) the expected additional profit for inflows in the range from 15 to 26.5 is found to be close to 211, thus higher than expected overcapacity costs which is 46. Thus, the installed long term capacity is perceived to contribute positively in expectation[8]. For capacities in the range from $K = 26.5$ to $K = 28$ the expected additional profit is only 2 versus additional expected capacity cost close to 5.8, and thus perceived to have a negative contribution to expected profit.

[8] $E[\Delta\pi(15 < R < 26.54)] = P(15 < R < 26.54)[\pi(26.54) - \pi(15)] \approx 211$

	Summary of Results Obtained by *Algorithm 1* and *Algorithm 2*.								
TYPE	seed	K_0	t	K_t	$\frac{1}{t}\sum_{i=1}^{t} g_i$	$\frac{1}{t}\sum_{i=1}^{t} \theta_i$	$\frac{\Delta\Pi_t}{\Delta K_t}$	p_t	dp_t
Benchmark				28.00	28.00		0.0000	94.00	2
DET *Alg.1*		0	40	28.00	27.13		0.0093	95.75	
Deterministic		10	37	28.00	27.39		0.0083	95.22	
Algorithm 1		20	31	28.00	27.68		0.0077	94.64	
		30	29	28.00	28.17		-0.0076	93.66	
DET *Alg.2*		0	758	28.00	27.96		0.0033	93.85	1.994
Deterministic		10	1297	28.00	27.87		-0.0010	94.26	2.009
Algorithm 2		20	2372	28.00	27.82		-0.0006	94.47	2.017
		30	2701	28.00	27.88		-0.0001	94.54	2.019
T1 *Alg.1*	1	0	57	28.01	26.91	-0.007	0.0130	96.17	
RD Type 1	2	10	60	27.96	27.32	-0.089	0.0021	95.26	
Algorithm 1	3	20	55	28.02	27.68	0.029	-0.0034	94.67	
	4	30	155	28.00	28.04	0.044	0.0264	93.95	
T1 *Alg.2*	1	0	164	28.01	27.95	-0.007	-0.0022	93.34	1.976
RD Type 1	2	10	327	27.99	27.84	-0.001	0.0149	95.02	2.036
Algorithm 1	3	20	995	28.00	27.75	-0.001	0.0196	95.14	2.041
	4	30	1191	28.00	27.75	0.020	0.0078	95.23	2.044
T2 *Alg.1*	1	0	678	22.21	18.26	1.492	0.0890	86.24	3.492
RD Type 2	2	10	115	22.87	18.22	1.504	-0.0231	86.15	3.504
Algorithm 1	3	20	698	22.47	18.17	1.507	0.0145	86.27	3.507
	4	30	379	21.96	18.09	1.521	0.0537	86.29	3.521
T2 *Alg.2*	1	0	841	17.55	17.52	1.490	-0.0676	88.56	3.480
RD Type 2	2	10	1233	17.39	17.51	1.512	-0.0157	88.51	3.517
Algorithm 2	3	20	915	17.48	17.32	1.509	-0.1569	89.36	3.548
	4	30	1176	17.41	17.41	1.529	0.0316	89.59	3.569
T3 *Alg.1*	1	0	286	28.81	27.64	0.993	-0.0447	94.06	
RD Type 3	2	10	382	28.83	27.40	0.981	-0.0354	93.36	
Algorithm 1	3	20	595	29.03	27.66	1.005	-0.1099	95.16	
	4	30	140	29.10	27.84	1.025	0.0221	96.70	
T3 *Alg.2*	1	0	529	27.94	27.95	0.993	-0.0026	93.33	1.983
RD Type 3	2	10	1233	28.07	27.84	1.004	0.0144	94.72	2.017
Algorithm 2	3	20	614	28.05	27.24	1.004	0.0497	96.33	2.076
	4	30	787	28.05	27.73	1.008	-0.1217	96.40	2.084
RDI *Alg.1*	1	0	1990	24.49	14.40	14.916	-4.0000	121.21	
Random Inflow	2	10	1990	24.66	14.76	15.300	-4.0000	120.49	
Algorithm 1	3	20	1990	24.42	14.53	15.039	-4.0000	120.94	
	4	30	1993	24.60	14.77	15.272	-4.0000	120.47	
RDI *Alg.2*	1	0	1991	26.57	14.65	14.916	-4.0000	120.39	1.996
Random Inflow	2	10	1992	26.68	15.02	15.296	-4.0000	119.87	2.014
Algorithm 2	3	20	1993	26.31	14.69	15.045	-4.0000	120.74	2.045
	4	30	1994	26.60	15.00	15.271	-4.0000	120.82	2.058

Table 5.3: Summary of the results obtained by the algorithms in each case/type, for all the initial values in Table 5.1.

Chapter 6

Discussion

By the results reported in Chapter 5 it was shown that the method developed in Chapter 4 provides convergence towards a long term level of capacity for the varying stochastic market situations which is considered in this thesis. To evaluate the performance of the method, this chapter discusses the compliance between the results obtained by the algorithms versus both theoretical expectations and the research presented in Chapter 2.

Since the model is known, expectations and marginal values for the various stochastic cases can be derived. Section 6.1 derives the most significant of these, and discuss the compliance between the numerical findings and theoretical expectations. When the proxy demand function is applied, a divergence occurs between the long term decisions in some of the cases, as seen in Chapter 5. A discussion on the reason for the differing results which is caused by the application of a proxy demand curve is given in Section 6.2. By way of conclusion the main features of the method is discussed in Section 6.3.

Note: To ease readability, the notation $f_x = \frac{\partial f(x)}{\partial x}$ is applied. Appendix D refer the calculations (intermediate or complete) used to derive the results and marginal values which is applied in this chapter.

6.1 Discussion of Compliance to Theory

Recall from Chapter 2 that uncertainty might lead to a ”tendency of reduced output” (Leland, 1972, p. 67). Risk attitude is also considered important when this issue is addressed. No explicit assumptions have been made about the agent's risk attitude, however, a linear utility ($\mathcal{U}(\pi, q) = \alpha\pi + \beta,$ for $\alpha, \beta > 0$) would imply that the agent is risk averse since the technology is concave (i.e. due to a concave profit function).

To check compliance to research, proposition 2 in Aiginger (1985, p. 55) will be used when applicable. The proposition states that *"a linear utility and technological concavity, neutrality, convexity* $(\Pi_{qss} < 0, \Pi_{qss} = 0, \Pi_{qss} > 0)$ *yield the following sufficient condition"*:

$$\Pi_{qss} \lesseqqgtr 0 \ \rightarrow \hat{q} \lesseqqgtr q^* \tag{6.1}$$

The proof of eq. (6.1) is obtained by applying Jensen's inequality $E[f(x)] \leqslant f(E[x])$ for a concave function (Sydsæter et al., 2002, p.181, def. 31.19)[1]. Thereby one may utilize that the effect of uncertainty on expected marginal revenue and marginal cost will determine the agent's adoption both in the short and long term, since profit is concave.

If the rate of change in marginal revenue is increasing in s $(MR_{ss} > 0)$, then expected marginal revenue is decreasing for a change in uncertainty. In order to keep expected marginal revenue in line with marginal costs in this situation, the agent can reduce production or increase price compared to benchmark level. Similarly, if the rate of change in marginal cost is decreasing in s $(MC_{ss} > 0)$, thus the expected marginal cost is increasing in uncertainty. To keep it in line with marginal revenue, production may increase or price may decrease compared to benchmark level. A combination of changes in both price and production may also occur as a response to uncertainty. If expected marginal revenue is unaffected by uncertainty $(MR_{ss} = 0)$ then production will be equal to the benchmark decision. Similar reasoning is given by Aiginger (1987, p. 52-53) and Leland (1972, p. 282).

Based on the definition of uncertainty which is examined in the experiments, the expectations can be derived. For an additive random demand defined by eq. (3.10), a zero-mean assumption and an additive nature will give an expected price equal to the certain price.

$$E[P(q,s)] = \int_{\underline{s}}^{\overline{s}} [a - bq + s]\, f(s)\, \mathrm{d}s = P(q) \text{ if } E(s) = 0$$

As discussed in Section 5.3.2 this is intuitive since a zero-mean random variable produces a symmetric outcome space which is equally positive and negative around the deterministic demand curve, as seen in Figure 3.1. Since marginal cost is linear, and since the effect on marginal cost and revenue from a change in uncertainty is oppositely dependant on the sign of s, $\left(MC_s \gtreqqless 0 \text{ if } s \lesseqqgtr 0 \text{ and } MR_s \lesseqqgtr 0 \text{ if } \ s \gtreqqless 0\right)$ no distortion caused by uncertainty occurs *if* the rate of change in marginal cost from uncertainty is zero. For the Type 1 uncertainty $MC_{ss} = 0$, and no distortion from benchmark level is detected in either Figure 5.2 nor Figure 5.7. From Table 5.3 we find that the price is only 0.5 above the benchmark price, for an average of 0.5 units lower production. Thus, one may conclude that the algorithms give results in line with Aiginger's proposition quoted in eq. (6.1) and in line with expectations for the Type 1 uncertainty case since the difference is negligible.

[1] For a convex function, the inequality is reversed. Aiginger (1985, note 5, p. 73) gives a proof for eq. (6.1).

For a multiplicative demand defined by eq. (3.11) with a positive expectation, the situation will according to Leland (1972) and Aiginger (1987) be different. Since the uncertainty is added to the slope, the expected price is decreasing for all outcomes of uncertainty[2].

$$E[P(q,s)] = \int_{\underline{s}}^{\overline{s}} [a - (b+s)\,q]\, f(s)\, \mathrm{d}s < P(q) \text{ if } E(s) > 0$$

A lower price affects the production costs since they are dependant on the selected level of production. For any given level of production, quantity can be given as a function of price since the functions are invertible. Let $\tilde{P}$ denote the price corresponding to quantity $\tilde{q}$, and let s denote uncertainty. Then the following apply:

$$\tilde{P} = a - (b+s)\tilde{q} \;\Leftrightarrow\; \tilde{q} = \frac{a - \tilde{P}}{b+s}$$

To find marginal change in quantity subject to a change in uncertainty $(\tilde{q}_s)$, differentiate the inverse expression for quantity with respect to uncertainty (s):

$$\tilde{q}_s = \frac{\partial \tilde{q}}{\partial s} = \frac{\partial}{\partial s}\left[\frac{a - \tilde{P}}{s+b}\right] = -\left[\frac{a - \tilde{P}}{(s+b)^2}\right] < 0 \;\; \forall s \geqslant 0$$

From the above expression it is clear that an increase in s will reduce the production *if* price remains unchanged. Since $q_s < 0$ and hence $MC_s < 0$ for all $s \geqslant 0$, such that $E(s) > 0$ implies a negative relation between uncertainty and costs. By inserting $E(s) = 1.5$ for s in the first order criteria in eq. (5.1) the optimal production is found to be 17.5, which is lower than the long term capacity derived by applying *Algorithm 1*. This divergence is caused by two simultaneous shifts with opposite effect, namely $MC_s < 0$ and $MR_s > 0$, for Type 2 uncertainty when *Algorithm 1* is applied.

As discussed above, a negative rate of change in marginal revenue $(MR_{ss} < 0)$ implies that the agent will reduce his production compared to benchmark in order to keep expected marginal revenue in line with marginal costs. However, since $MR_s > 0$ and $MC_s < 0$, the production will occur at a level below the certain production, but above the level found by eq. (5.1) for $s = 1.5$, since some of the downward bias is countered by a positive, but diminishing rate of change in profit for a change in uncertainty $(\Pi_{qss} \geqslant 0 \,\forall\, s)$ when *Algorithm 1* is applied. According to Aiginger (1987, p. 55) a convex marginal cost in uncertainty will result in a reduced price compared to the deterministic price, dependant on the rate of change in marginal cost. In the Type 2 case $MC_{ss} > 0 \,\forall\, s$, such that a distortion from benchmark level is expected. If the algorithms are compliant with this, the results should reflect this downward bias compared to benchmark.

[2] A value of 0 can be the outcome, although the probability of this event is zero

In both Figure 5.3 and Figure 5.8 a negative distortion is evident since the long term capacity which is installed in the business is below the benchmark level. According to Table 5.3 the average production is 18.19, which yields an average price of 86.2. By which it is shown that *Algorithm 1* produce results compliant to Aiginger's proposition.

Since the proxy demand curve does not include any form of uncertainty directly, the rate of change in both marginal costs and marginal revenue is by definition unaffected by uncertainty when *Algorithm 2* is applied. Both short term production and long term level of capacity found by the algorithm should therefore reflect this. Eventually the adaptive learning of price and slope will uncover the surrounding market conditions, such that the long term capacity thus should converge towards a level closer to the expected level of production in this experiment. The application of *Algorithm 2* in the Type 2 case results in a long term level of capacity which is equal to the expected level of production for an average price of 89, and hence compliant with Aiginger's proposition.

Thus it is shown that both algorithms produce results which is in line with the earlier referred results since both production and price is reduced compared to benchmark. The lower price compensates the agent for the uncertainty in the higher regions of the demand curve and counteracts some of the risk as perceived by the agent if demand turns out to become low. The conclusion is that the method provides the agent with the needed knowledge such that he finds the best choice of long term capacity and short term production, both at levels in line with expectations for the random demand Type 2 case.

Similar to the Type 1 uncertainty, Type 3 uncertainty also creates a symmetric outcome space around the deterministic demand curve, though with reduced volatility in prices for an increasing production, as shown in Figure 3.2. For random demand Type 3, defined by eq. (3.12) the expected price equals the deterministic price when $E(s) = 1$.

$$E[P(q,s)] = \int_{\underline{s}}^{\overline{s}} [a - bq]\, sqf(s)\, \mathrm{d}s = P(q) \text{ if } E(s) = 1$$

Since the rate of change in marginal revenue for a change in uncertainty for Type 3 case is zero, no difference from benchmark production is expected in order to keep expected marginal revenue in line with the marginal cost for this type of uncertainty. Inasmuch as the average production in Table 5.3 is equal to the benchmark level, we find that the algorithms produce results which harmonize well with theory and performs in line with expectation[3]. However, the long term capacity which is obtained by implementing *Algorithm 1* is (slightly) higher than benchmark capacity, while it is equal when *Algorithm 2* is applied.

[3] Compare $\frac{1}{t}\sum_{i=1}^{t^*} g_i$ in rows T3*Alg.1* and T3*Alg.2* to the rows DET*Alg.1* and DET*Alg.1*

By a similar approach as for the Type 2 case, the inverse relations can be applied to show why this positive difference results from using *Algorithm 1*. By doing so, one finds that since marginal cost is concave in uncertainty ($MC_{ss} < 0$) the expected marginal cost is decreasing in uncertainty such that the rate of change in profit for a change in uncertainty is positive ($\Pi_{qss} > 0$).

According to Aiginger's proposition quoted in eq. (6.1) the production under uncertainty will be higher than under certainty *if* the price remains unchanged. However, from Table 5.3 one finds a small increase in price compared to benchmark, for on average the same production level. The influence on the long term decision, is an increase in capacity since a positive expected rate of change in profit for a change in uncertainty implies an increased dual value for an increased uncertainty. In turn, this will increase the installed level of capacity via the iterative steps of the algorithm where the long term criteria in eq. (4.16) is implemented. This is natural since an increased long term capacity will not impose a higher risk to the agent in the Type 3 case. On the contrary, the lower the capacity, the higher is the observed volatility in prices, and the higher the risk as perceived by the agent. Thus, by increasing the long term capacity, the risk as perceived by the agent is reduced since there are no discrepancy between produced and sold quantity in these experiments.

When *Algorithm 2* is applied for the Type 3 demand case, a similar effect as was seen in the Type 2 case occurs, since the rate of change in both marginal revenue and marginal cost by definition is not influenced by uncertainty. In Figure 5.9 it is clear that there are no difference between the benchmark level and the long term level of capacity found by using *Algorithm 2*. This shows that both the developed algorithms contribute to a learning process such that the agent acquires sufficient knowledge in order to choose both production and capacity in line theory also for the random demand Type 3 case.

For the **random inflow availability case**, the expectation is not as readily available, however note that expected input is $E(R) = 15$ according to the data applied in the experiments. In the long run, the expected constraint on the production is given by

$$q \leqslant E[\min(K, R)] = \min(K, 15)$$

such that the production most probably is close to $E(R) = 15$ on average. This is confirmed by Table 5.3 since the average production is 14.61 and 14.84 for *Algorithm 1* and *2* respectively. Still, the main interest is whether the agent learns the best choice of long term capacity by applying the algorithms, when faced with random inflow.

The agent chooses to install a long term capacity of 24.54 or 26.54 units on average when he uses *Algorithm 1* or *Algorithm 2* respectively. Despite the fact that the agent has high capacity costs at these levels, the potential additional profit is higher than the expected over-capacity cost, as shown in Section 5.3.3 and Section 5.4.3. These long term levels of capacity enables the agent to utilize approximately 82% or 88% of any inflow, and since potential income is high if inflow is above the expected level, the costs incurred are accepted. However, the price also increases significantly compared to benchmark level in the random inflow case. Since the random inflow is a uniform random value, the probability of observing all levels of inflow is equal, thus approximately 50% of the inflow is below expected level. The high prices should therefore be interpreted as an insurance to reduce the risk of low profit when the random inflow turns out to be below expectations.

By utilizing the definition of the long term criteria and the numeric dual values returned from the experiments, we can show that these levels of capacity are (approximately) efficient, by showing convergence in probability. Recall from Chapter 4 that the dual values can be regarded as stochastic variables in the stochastic cases, because they are dependant on a specific outcome of stochastic inflow. By the law of large numbers, a sequence of random values λ_i will, under suitable conditions, converge towards the expected value of the random value, $E(\lambda)$. For a large enough sample this implies that the average of the observed dual values in the random inflow case should converge towards the expected marginal costs of capacity in probability (Hogg and Tanis, 2005, p. 258).

$$\frac{1}{n}\sum_{t=1}^{n}\lambda_i = \bar{\lambda} \to E(\lambda) \Rightarrow \lim_{n\to\infty} P\left(|\bar{\lambda} - E(\lambda)| < \epsilon\right) = 1 \tag{6.2}$$

when ϵ is an arbitrary small value. It is assumed by the long term criteria in eq. (4.16) that the marginal change in profit for a change in capacity is equal to the difference between the dual value and the marginal cost of capacity, that is $\frac{\Delta\pi}{\Delta K} = \lambda_K - MC(\hat{K})$. Hence, if the average dual value of capacity from the experiments are close to constant marginal cost of capacity, i.e. if $\bar{\lambda} \to MC(\hat{K})$ then the expected long term criteria also converges towards zero in probability. That is:

$$\lim_{n\to\infty} P\left(\left|\bar{\lambda} - E(\lambda)\right| < \epsilon\right) = 1 \;\Rightarrow\; \lim_{n\to\infty} P\left(\left|E\left(\frac{\Delta\pi}{\Delta K}\right)\right| < \epsilon\right) = 1$$

Table B.5 and Table C.5 give only an excerpt of the full numerical results, however by using the complete data we find the average dual values to be $\bar{\lambda} = 4.06$ using *Algorithm 1*, and $\bar{\lambda} = 4.52$ when applying *Algorithm 2*. Both of these values are close to actual marginal cost of capacity equal to $MC(K) = 4$ by the data in Table 5.2. This implies that convergence in probability is obtained by the method, at least for the data used in these experiments.

Tests show that a change in step-size or in the stop-if criteria does not alter the long term capacity significantly. By repeating the $Ex1$-experiment for a step-size of $\delta_t = \frac{1}{1+t}$

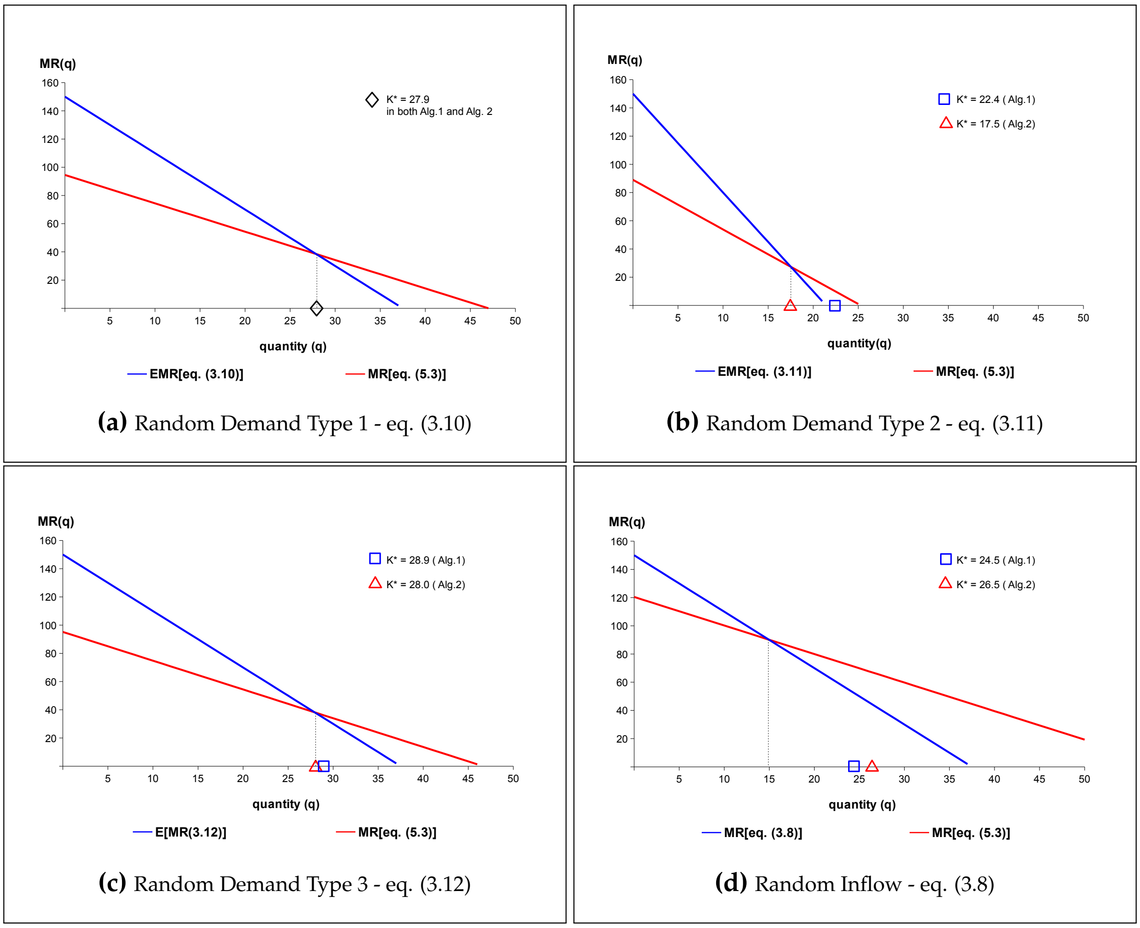
and with a Stop-if-criteria set to 0.0002, the long term capacity acquired by *Algorithm 1* is 24.55 and by *Algorithm 2* is 26.51. The change in parameters increase the number of iterations required to obtain convergence more than ten times for a change of 0.3% and -0.12% compared to the previous $Ex1$-experiments shown in Figure 5.5 and Figure 5.10. Repeated tests using different values for step-size and stop-if criteria display the same consistent results, such that it is fair to conclude that the algorithms provides the agent the knowledge to choose an approximately efficient long term level of capacity.

Summing up the above discussion shows, that by applying functional analysis we find that both of the developed algorithms provide results which harmonize well with theory, represented by Aiginger's proposition in eq. (6.1). Thus, both algorithms performs in line with expectation in all of the random demand cases. The reduced price in the Type 2 case compensates for the uncertainty in the higher regions of demand, and counteracts some of the risk as perceived by the agent, if demand turns out to be low. It was also shown by using probability approximation that the resulting choice of long term capacity in the random inflow case is at a level which implies convergence in probability towards the expected level. The price level in the random inflow case is also in line with Aiginger's proposition, since price is increased significantly by both algorithms. It serves as an insurance against the high risk of low profit if inflow turns out below expectation. Note that ex-post flexibility implies a possibility of divergence between demand and production, while this thesis has implicitly assumed equilibrium between demand and production. Hence, the effect of ex-post flexibility has not been addressed in this thesis.

By way of conclusion it is thereby fair to claim that the algorithms provide the agent with the necessary knowledge about the various stochastic market conditions such that he becomes able to make an efficient choice of long term capacity by learning. The algorithms thereby meets the learning requirement as requested by the main objective in Chapter 1.

6.2 Proxy Demand and Differing Results

In some of the stochastic cases, a divergence occurs between the long term decisions when the agent implements *Algorithm 2* compared to *Algorithm 1*, as seen in Chapter 5. This is caused by the application of the proxy demand in eq. (5.3) since the perceived nature of demand differs from the true nature of it. A proxy demand leads to a choice of production, and thus price, which is different from what would be the choices if the true demand was applied, since the costs are the same in both algorithms. As a consequence the dual values of capacity differs between the algorithms, since they reflect marginal profit under current market conditions. Thus, when the proxy demand is used, the effect of uncertainty only comes as a secondary effect of learning, and not as a direct consequence of market interactions. Consequently, the rate of change in perceived marginal revenue is unaffected by uncertainty since demand, as perceived by the agent, is independent of uncertainty. The anticipated rate of change in marginal costs are accordingly also independent of uncertainty. For that reason, dependant on type of uncertainty, the choice of long term capacity can be influenced by the differences between the algorithms.



(a) Random Demand Type 1 - eq. (3.10)

(b) Random Demand Type 2 - eq. (3.11)

(c) Random Demand Type 3 - eq. (3.12)

(d) Random Inflow - eq. (3.8)

Figure 6.1: The figures show expected marginal revenue for actual demand versus marginal revenue for proxy demand in eq. (5.3) for price (p) and slope (dp) derived by *Algorithm 2*, see Table 5.3. Average long term capacity indicated by □ for *Algorithm 1*, △ for *Algorithm 2*, or by ◇ if long term capacity is equal in the two algorithms.

Figure 6.1 compare the marginal revenue curves for actual and proxy demand in the stochastic cases. Marginal revenue of the proxy in eq. (5.3) is graphed for perceived price and slope after convergence is obtained in each case. Expected actual marginal revenue is graphed for a random factor equal to it's expected value. Average long term capacity found by each of the algorithms are indicated by markers on the quantity axis. By examining Figure 6.1 we see that average production when *Algorithm 2* is applied (indicated by the dotted vertical line), is consistent with the point where the expected marginal revenue intersects with the perceived marginal revenue. This is intuitively natural since the agent uses the same short term maximizing strategy, and has the same cost structure as when *Algorithm 1* is applied. Thus, the agent's efficient choice of production occurs at the level where the perceived marginal revenue equates the actual marginal costs.

The observed discrepancy in production between the algorithms is small. According to Table 5.3 it is less than $+0.3$ for all of the cases, except for the random demand Type 2 case where the difference is -0.74. As shown by the distance between the markers (△ - □) on the q-axis in the graphs shown in Figure 6.1, the difference in the long term capacity decisions between the algorithms are relatively small, except for the Type 2 case.

From Figure 6.1b we see that the Type 2 uncertainty causes a significant divergence in the long term capacity decisions between the two algorithms, from 22.4 by *Algorithm 1* to 17.5 by *Algorithm 2*. As discussed in Section 6.1, part of the downward bias caused by a positive expectation of random demand, is countered by a positive, but diminishing rate of change in profit for a change in uncertainty when random demand defined by eq. (3.11) is applied in *Algorithm 1*. In *Algorithm 2* the price and slope which is applied by the agent, is a result of adaptive learning from observed values. Thus, uncertainty is only a secondary effect from the learning scheme, such that perceived rate of change in profit for a change in uncertainty in this situation is zero. Hence, no of the downward bias will be countered when *Algorithm 2* is applied, such that both production and long term capacity ends up at a level consistent with expected price when $s = 1.5$. A decrease in both capacity and production compared to benchmark levels results, and despite a small increase in unit prices, the average profit decreases significantly.

In the Type 3 case, the reason for the difference is naturally similar to the reason in the Type 2 case, since the rate of change in profit for a change in uncertainty is zero also for this case when *Algorithm 2* is used. Hence, both production and long term capacity results at a level consistent with the solution of the first order criteria when $s = 1$. Thus, the long term level of capacity and average production is equal to the benchmark levels in the Type 3 case when *Algorithm 2* is implemented.

6.3 Simplicity and Assumptions

The main goal was to develop a *simple* method which enabled learning from real world observations, and doing so by using a limited number of assumptions. From the previous sections, with all the mathematics and statistics which have been applied, the impression is perhaps that specific skills are required in order to obtain efficient results by this method, skills which an average agent might not have. Section 6.1 showed that the learning requirement was achieved, thus, one question remains to be addressed:

*Is the developed method a **simple** way of learning stochastic market conditions?*

Figure 6.2 show a schematic set-up of the learning process applied in the developed method. Mathematics and statistics aside, all which is required of the agent in order to apply the method is (1) to choose some arbitrary values for the learning objects in order to initialize the process, and (2) observe the phenomenon of interest. Based on the observations, he (3) then chooses a level of production to maximize his profit, and (4) update his knowledge according to the new information. Then all he needs to do, is to repeat the process by using the new information to initialize the next sequence.

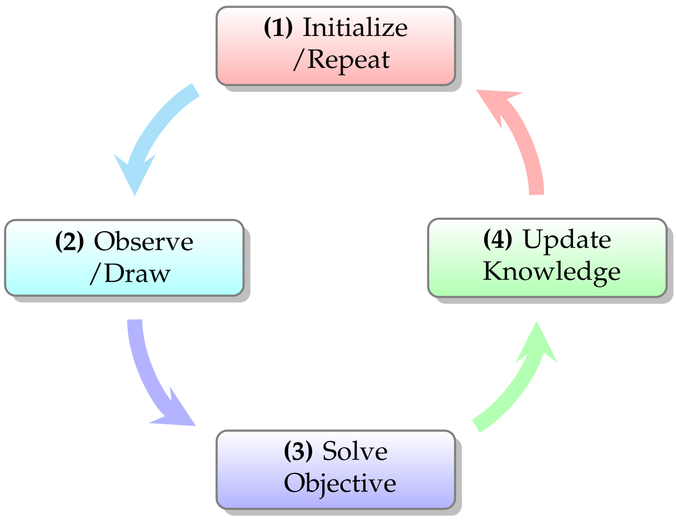


Figure 6.2: A schematic view of the learning process. Four steps are required: (1) initialize the algorithm, (2) observe phenomenon of interest, (3) select production by maximizing objective (4) update knowledge by using the dual value, and repeat.

The learning process presented in Figure 6.2 is pretty straight forward, and easy to implement in a real decision process. There are however two pivotal considerations which affects the simplicity, and possibly also the applicability of the method. The first pivotal point is the determination of initial values for the learning objects. Based on the results reported in Chapter 5 it is evident that the selected initial values does not have any significant impact at which level of long term capacity convergence is obtained. Thus, the choice of initial values may be random, without affecting the efficiency of the method.

The second pivotal consideration which mostly affects the applicability of the method, is whether or not the agent can evaluate, or determine a suitable approximation for, the dual value since this value in most cases is unavailable if the agent's skills is limited. However, from Section 4.1 we know that the developed algorithms are based on the gradient method. Common for iterative methods are that they are applicable for a finite difference instead of the derivative of the function of interest. Thus, if the dual values are unavailable, a suitable proxy for the gradient in eq. (4.16) is a finite difference as defined by eq. (6.3).

$$\frac{\Delta \Pi_t}{\Delta K_t} = \frac{\Pi(K_t + h) - \Pi(K_t)}{h} \tag{6.3}$$

For a sufficiently small change in capacity, i.e. for a sufficiently small value of h, the finite difference defined by eq. (6.3) essentially equals the definition of the derivative of a function (Sydsæter et al., 2002, Def. 3.11, p. 20). That is:

$$\frac{\partial \Pi(K_t)}{\partial K_t} = \lim_{h \to 0} \frac{\Pi(K_t + h) - \Pi(K_t)}{h}$$

Thus, when using eq. (6.3) as a proxy for the gradient, the discussion of compliance to theory is still valid, since the functional analysis which is used in Section 6.1 also is applicable in a discrete context. Consequently, the gradient can be replaced by eq. (6.3) and still provide efficient results which is compliant to economic intuition and others findings. Though, the accuracy might change somewhat since a finite difference necessarily is a rougher estimate than a shadow price calculated by linear programming.

Number of Necessary Assumptions

The number of necessary assumptions required for the method to obtain the results as discussed above is fairly limited and non-restrictive. The cost curve must be assumed to be a known continuous function, convex in own decision. Expected profit need to be assumed concave in own decision. The existence of some isolated maximal points at which the functions are defined is also a necessary assumption. This assumption is not as restrictive as it may seems, as it only requires the profit curve to exist in the point at which the maximal value is achieved . Thus, a discrete model may also be implemented in the method if the profit function is defined, and continuous, over a arbitrary small interval around the maximal point.

In order for the algorithms to produce results in line with theory, the short term decision made by the agent must be in line with the usual first order condition. This implies that the agent's choice of short term production is such that marginal costs equates expected marginal revenue. Last necessary assumption is that the observed random factors are independent identically distributed variables.

When the proxy demand curve defined in eq. (5.3) is used in a learning heuristic given by eq. (4.2), some additional assumptions are required to assure convergence of price and slope towards the expected values. The required properties and assumptions are discussed by Flåm and Sandmark (2000, p 5-7), and will thus not be repeated here. To show that the proxy demand function applied in this thesis have the required properties, we may apply Theorem 1 in Flåm and Sandmark (2000, p 5). It states that *"convergence in beliefs that are confirmed in the mean"* occurs *if* the asymptotic limit of the composite average function of price and slope, defined as $f(p, dp) = EF(P, \theta) := [EP(q, \theta), \frac{\partial}{\partial q} EP(q, \theta)]$ is bounded on it's domain. Without going in to all the details of the proof for asymptotic convergence, note only that the proof uses Bendixon's criteria. It states that a periodic solution is precluded if $div f(p, dp) \neq 0$ (Feckan, 2001, p. 3395) and thus the composite average function is then bounded. Since $div f(p, dp) < 0$, the sequence of p_t and dp_t defined by eq. (4.2) almost surely convergence to a fixed point in expectation, and thus the required limiting properties are established for our case[4].

6.4 Critique

Some of the most significant limitations to the model were discussed in Section 1.2, and will therefore not be repeated here. Still, the model and the experiments have some obvious weaknesses which requires a brief comment. We will here point to some of the most prominent properties of the model and parameters which potentially corrupts the possibility to draw general conclusions based on the experiments conducted in this thesis.

1. The model is simple, known and all of the components are known and well defined.
2. The width of the support in the random inflow case.
3. The definitions of the random demand cases.
4. The upper and lower boundary applied in the development of the capacity.

To start at the top of the list above, it is important to emphasise that the model which has been applied is simple in terms of the included functional relations. All of the curves are continuous with well known, well defined properties for all the components included. This is contrary to what is usual in reality, where at least parts of the model would be the result of estimations or approximations for most (if not all) of the parameters. However, the simplicity of the model provides the opportunity for compliance control, which was deemed important in order to document that the method produced results in line with expectations.

[4] See Appendix D for the derival of the divergence $div f(p, dp)$.

The second and third points on the above list are somewhat interconnected, since they both point to uncertainty has been modelled in this thesis. Uncertainty in real life business decisions has most often unknown, possibly asymmetric distributions, which in some (or most) cases also are dependent on unobservable parameters. This is contrary to the uncertainty which have been applied in this thesis. It has been modelled as a symmetric uncertainty, where all possible outcomes have the same probability of occurrence, a feature which might seem far from real life.

The width of the support for the random inflows is such that it affects the results derived, since it is defined such that the level of production at which the profit is maximized is included in the possible outcomes. This implies that the maximal level of production is achievable in our experiments, since $\underline{\theta} < K^* < \overline{\theta}$. If the support for the inflow was defined such that the maximal level of production was unachievable, the results obtained would be different from the results reported in Chapter 5 for this case. We have no reason to claim that it is more or less realistic to observe random values which includes or precludes inflows at levels which makes the maximal level of production achievable.

The random demand cases and specifically the way the uncertainty is introduced in those cases, is defined such that they mimic different random demand situations. There is no reason to prefer either one of the types of uncertain demand applied in this thesis, since neither one of them is more realistic than the others (Aiginger, 1985, p. 61). Uncertainty in demand is usually not a choice, but merely an outcome served to the agent by the market.

Both upper and lower boundaries were applied when developing capacity. It is somewhat unreasonable to assume that the maximal level of production will be achievable within some predefined boundaries, especially if the knowledge about either the market or the profit curve is limited. The application of boundaries for the capacity may however be justified by available data, as discussed in Section 3.2, or by budgets which restrains the amount of capital which is available to build the capacity.

Despite the flaws and shortcomings as discussed above, our claim is that the method provides the agent with enough knowledge such that he acquires the competence required to make an good long term decision. The results indicate that even if the parameters were to be replaced with estimated values, the long term capacity would reflect an approximately efficient choice, though the results would be influenced by a somewhat higher uncertainty.

Chapter 7

Summary and Future Work

The key question underlying the work of this thesis was: *How can the optimal long term level of capacity be determined, if only short term conditions are known, and the agent's skills to uncover the long term characteristics are limited?*

The primary aim was therefore to develop a simple and unified solution algorithm for complex stochastic long term decision problems. The algorithm should enable learning of best choice of long term capacity under uncertainty, based on discrete observations of some selected learning objects. This thesis secondary purpose was to apply the developed algorithm in numerical experiments to examine the effect of uncertainty on the level of long term capacity. Various types of uncertainty were applied, in order to examine the compliance between the performance of the algorithms compared to others findings and expectations derived analytically.

7.1 Summary and Contributions

To form a basis on which to perform the desired experiments, some background literature was presented in Chapter 2. To enable comparison between theoretical expectations and numerical findings, a simple economic model for a business was set up in Chapter 3. Then, in response to the primary aim, two algorithms were developed in Chapter 4, one for a single learning object (capacity) and one for multiple learning objects (capacity, price and the slope of the demand curve). Both algorithms were based on the gradient method, and implemented a unified long term criterion, derived by applying the Kuhn-Tucker conditions to the agent's objectives. It was shown that the long term criterion was valid for all types of applied uncertainty, despite the differences in both objectives and constraints for the various cases considered in the conducted experiments.

To investigate the effect of uncertainty on the long term choice of capacity, numerical values were assigned to the parameters in the applied model. Subsequently, the experiments were conducted by using non-linear mathematical programming implemented in AMPL. Four types of uncertainty (random inflow, additive demand and two types of multiplicative demand) were applied in the experiments, and both algorithms were tested for a wide range of initial values for the selected learning objects.

The results presented in Chapter 5 showed that convergence was achieved in all cases and by both algorithms. Whatever the initial guess the agent had for the learning objects at the start of the learning process, he uncovered a good approximation of long term capacity in each case. Though, both the development process of capacity and the selected long term capacity levels varied, as expected, dependant on the type of uncertainty.

Chapter 6 discussed the compliance between the theory and numerical findings. It was shown that the selected levels of long term capacity were consistent with the theoretical expectations in each of the random demand cases. For the random inflow case, the discussion also showed that the long term capacity was selected such that the long term criterion converged towards zero in probability, which implied that marginal cost of capacity also converged towards marginal profit of capacity in probability. Thereby it was shown that all the selected long term capacities were in line with both economic intuition and theoretical expectations. In Chapter 6, the applicability of a proxy demand function were also discussed, and it were showed that a proxy demand could be applied if demand was unknown. When price and slope were a learning object, differences i long term capacity occurred in some of the cases, without this leading to significant changes in terms of the correspondence between results and expectations.

7.2 Conclusion

The conclusion based on our experiments is that adaptive learning provides insight into long term market conditions using fairly simple methods. A short answer to the key question is therefore that a good approximation for the optimal long term level of capacity can be determined by using an adaptive learning algorithm. The slightly longer answer is that an adaptive algorithm contributes to uncovering the long term market conditions. This is due to fact that the knowledge based on current observations accumulates in the agent's expectations in the long term. Thereby, the unknown long term market conditions are gradually demystified, and eventually confirmed in expectation when convergence is obtained for the learning objects. The learning objects studied in this paper included capacity, price and the slope of the demand curve.

Neither special skills, nor statistics, are required to determine an (approximately) efficient long term capacity. In terms of capacity development all that is required of the agent is to select an arbitrary initial value to initiate the process, observe the current market conditions, select production such that profit is maximized under current conditions, and adjust the capacity at the end of each period according to the newly acquired knowledge. As long as the adjustments are based on the perceived marginal change in profit for a change in capacity, the long term capacity will eventually end up at an (close to) efficient level. The perceived marginal change in profit for a change in capacity can also be replaced by a finite difference, without this affecting the long term efficiency of the method.

Even when the actual demand curve is unfamiliar, and a proxy is applied, adaptive learning of the price and slope will provide the agent with the needed knowledge to select a long term capacity in line with the market conditions. Over time, the agent's prevailing belief about price and the slope of the demand curve will reflect the actual market conditions, inasmuch as his knowledge of the learning objects are updated as a moving average of current beliefs and actual observations. Due to the properties of the adaptive algorithm, the long term perceptions of price and slope will approach an empirical average when the number of periods increases. The agent's choice of capacity will thereby reflect his best possible choice under current market conditions - Both within each period, and most importantly for efficient long term choices, in the long run.

7.3 Future Work

For the simple uncertainty model applied in this thesis, we find a consistent compliance between the numerical results and (mostly all) the theoretical expectations. The results indicates that the algorithms might also be applicable for more advanced economic business models. This inspires to further development and testing of adaptive learning algorithms, and their applicability in more complex stochastic problems.

Some avenues for further research on adaptive learning under uncertainty could be to:

- Introduce an irreversible capacity decision, and allow for disequilibrium between production and demand. This would resemble the numerical experiments conducted by Driver et al. (1996). It would be interesting to see whether or not an adaptive algorithm would give similar results as found by Driver et al.

- Examine the effect of random factors with assymmetric probability distributions on long term decisions. An interesting case would be to apply a random factor with a fat-tailed distribution, as they imply high risk in the region of the fat tail, and otherwise a small risk. (For example a scaled Beta distribution, for $\alpha \neq \beta$)

- Include inter-temporal relations in a utility maximizing model, such that future capital availability become influenced by the short term choices. A division between short term choices versus long term decisions would make it possible to examine how uncertainty affects both the short and long term decisions, and the interrelation between them.

- Increase the numbers of market participants, for example by modelling an oligopoly. Flåm and Sandmark (2000, p.7-9) showed, using a similar approach as applied in *Algorithm 2*, that a Cournot equilibrium in production was achieved for the market's participants. It would be interesting to examine the effect of uncertainty on each of the producer's long term decisions in an oligopoly, in similar experiments as conducted in this thesis.

The suggested avenues for further research given above, are by no means exhaustive. Since the number of necessary assumptions is limited, and non-restrictive, the algorithm will most probably be applicable for mostly all cases where optimal choices are to be approximated in expectation. The suggestions spans a wide range of interesting areas to investigate numerically, although some of the suggestions possibly violate the method's strength, namely the simplicity in both implementation and applicability.

Appendix A

Programming in AMPL

To model, and solve, the different experiments in this thesis a student-version of AMPL (A modeling language for mathematical programming) have been used. AMPL is chosen due to an intuitive syntax, flexible data input and the possibillity of interactive dialog with pre-defined solvers to perform optimization. The program is available from www.ampl.com (AMPL). A comprehensive guide to AMPL is written by Fourer, Gay, and Kernighan (2003). The programs consists of the following seperate files:

1. Model-file (*.mod): Defines the abstract mathematical model.
2. Data-file (*.dat): Define numeric values for the definitions in the model.
3. Run-file (*.run): Commands used to solve, and resolve, a defined problem.

The complete model-, data- and run-files are given in seperate appendixes to assure reproducible results for each case considered. Tables of excerpts of the numerical results are given at the end of each appendix. All of the results are summarized in Table 5.3 at the end of Chapter 5.

- AMPL programs for *Algorithm 1* separately for each case: Appendix B
- AMPL programs for *Algorithm 2* separately for each case: Appendix C

The command environment are provided by the SW window (Technologies, 2000). Lines of text in a AMPL program which starts with `#` are not interpreted by the solver since they are treated as comments only. Text given in `typewriter font` in Appendix B and C are commands which is sent to the selected solver. All of the commands used in the AMPL language must be ended by a semicolon `;`

Keywords

A program in AMPL have the following setup of keywords, in the given order, with a predefined interpretation within AMPL:

- `set`: defines a set. May be numeric (`set T 1..5;`) or alphabetic (`set T a,b,c;`). A set can be empty when initiated (`set T default;`) then values can be added to the set at any later stage (`let T := T union b;`).
- `param`: defines a parameter, either as a single value by `param H;` or indexed over a set: `param H {t in T} default 0;` The latter give a vector with norm T, all set to 0 by default.
- `var`: defines the control variable. A non-negative variable is coded by `var Q >= 0;` Functions can be also defined by the var command: `var P = a - b * Q;`
- `maximize`: Tells AMPL to maximize the objective function, similarly `minimize` will minimize the objective function.
- `subject to`: defines the constraints of the objective function. The keyword `subject to` can be omitted as AMPL assume any logic conditions as constraints if they appear without any of the other keywords.

The order of appearance of the keywords does matter in the AMPL language as a value can not be used before it is defined. Also note that AMPL is case sensitive so the commands: `set H;` and `set h;` are interpreted as two different sets.

Programming of DetModel.mod

The sets are usually defined first and named by capital letters. By defining a parameter (`H`) first it is easy to manipulate the length of the time-frame for the iterative process by changing only one value in the data-file:

- `param H > 0;`
- `set T default 1..H;`

Based on the definition of capacity in eq. (3.5) the bounds of K are defined as two positive parameters:

- `param K_under >= 0;`
- `param K_over >= 0;`

Similarly all the parameters for demand, production- and capacity-cost must be defined. As the complete model is given in Appendix B and C not all definitions are repeated here. Though it is not needed for the applied algorithm, construction of a parameter indexed over the set T give a vector which store results from the iterations. Vectors are given for optimal production, step-size, profit, capacity, dual value and gradient in each iteration. Below the construction for the capacity K and optimal production Q ve are given:

- `param K{t in T} default 0;`
- `param Q_ve{t in T} default 0;`

When all parameters are given, the control variable is defined by:

- `var q >= 0;`

To facilitate iterative recalculations of price and cost for each new value of the control variable they are defined as variables:

- `var Cost = A*q + B*q*q;`
- `param Q_ve{t in T} default 0;`

The objective function and the constraint, both given in (3.17) are defined in AMPL by the following lines of code:

- `maximize profit: Price*q - Cost - (r+d+f)*K[n];`
- `subject to CONST: q <= K[n];`

By assigning the problem a name, repeated solving in a loop is simplified:

- `problem DET: q, profit, CONST;`

The complete deterministic model file is given in Section B.1.

Programming of DetData.dat

The DetData.dat-file in Section B.1 consist of a list of numerical values, all consistent with the parameters given in the model. The syntax for a single numeric parameter is:

- `param a := 200;`

Programming of DetRun.run

By typing: `include DETrun.run;` in the SW-window the run-file is executed[1]. To read model and data from a local folder named **foldername**, apply the change directory command (`cd;`) and change directory again after the files are loaded to select and load the preferred solver.

- `cd ''C:\Users \...\foldername'';`
- `model DetModel.mod;`
- `data DetData.dat;`

[1]Assuming the run file is saved under the current working directory (Fourer et al., 2003, p. 489-490)

- `cd ''C:\ Program Files \ amplcml'';`
- `option solver cplex; option cplex options 'sensitivity';`

The commands `for`, `if-then-else`, and `break-continue` are used to update the iterative step and to control the stop criteria at the end of each period. A for-loop has the following syntax:

- `for {t in T} {...};`

where the statements contained in the body {...} are executed for each `t` in the set `T`. Inside the body of the `for`-loop a series of assignments are given. First a counting parameter `t` and step-size λ_t in eq. (4.6) are updated. Then, the problem is solved subject to current market conditions by the command `solve DET;` and all results are saved in vectors by the `let`-command. Only selected examples are given below:

- `let n := t;`
- `let delta[n] := 1 / (1 + n);`
- `solve DET;`
- `let Q ve[n] := q;`
- `let Profit ve[n] := profit;`

Single learning algorithm include the iterative step:

- `let K[n+1] := K[n] + delta[n]*gradient[n];`

When the update of capacity is done, the bounds are checked. If the new values are outside the allowed range, they are projected back to the allowed range by:

- `if K[n+1] < K_under then let K[n+1] := K_under;`
- `if K[n+1] > K_over then let K[n+1] := K_over;`

To control the repeated calculations and make sure that the loop will terminate at some point, a logical condition in a break-continue command is applied. For example:

- `if n >= PreStep`
- `then if abs(K[n+1]-K[n]) <= stopIF then break;`
- `else continue`

The above states that for the iteration number defined by parameter PreStep, or higher, the absolute difference between new and old beliefs is checked to see if it is small enough, i.e. smaller than the stopIF criteria. If so, the for loop will be terminated and the calculations stopped. For a more thorough discussion of conditional calculations and termination of loops see Fourer et al. (2003, p. 258).

To display the results in the SW window, type:

- `display K, Q_ve, Profit_ve, gradient;`

Programming of the stochastic cases

In the stochastic cases there are additional parameters compared to the above deterministic codes. The construction of the uncertainty, as described in Section 3.1 is done by using the `option randseed (X);` and the predefined keyword `Uniform`. Below the Type 1 uncertainty is used as example. For the other cases, the codes are similar, while the data will change. In Type1Model.mod-file four lines of code are added to define the bounds of the uncertain demand parameter. As before, a vector is included to facilitate display, though it is not needed:

- `param s under;`
- `param s over;`
- `param s_ve{t in T} default 0;`
- `param s_nr default Uniform(s under, s over);`

In the Type1Data.dat-file the numerical values of the bounds are added:

- `param s under := -0.5;`
- `param s over := 0.5;`

In the stochastic cases the command `param X default Uniform(a, b);` generate pseudo-random values only, as discussed in Section 1.2. To control how these are generated, and to assure reproducible results, the command `option randseed (X);` is used to set the seed given to the computer equal to `X`. This is done by first defining a numeric value `l := 1` and then assign it to `X`.

- `let randseed[l] := X;`
- `option randseed (X);`

AMPL, as other programming languages, will assign the same value to `s_nr` unless the data for `s_nr` is reset in each iteration inside the for-loop. This will, together with the above `option randseed (X);` make sure that we get the `T` different psedo random values for `s_nr` in each trial, and that these values will be the same each time `l:=1`.

- `reset data s_nr;`
- `let s_ve[n] := s_nr;`

Solver Selection and Applied Option

There exists a wide range of solvers applicable for the AMPL system, and both CPLEX 11.2 (Technologies, 2010) and MINOS 5.5 (Murtagh and Saunders, 2003) have been used during the trial-and-error process of the thesis. The results resported are calculated by CPLEX 11.2, which both handles linear and non-linear constraints, and also provide suffixes for analysis as a part of the solver code (p.19-20). Note: CPLEX does not solve all types of non-linear problems, but can be applied on continuous quadratic objective functions (p.33).

The `suffix-option` provide an easy access to the numerical value of the dual value, or approximated shadow price, for a specific constraint after maximizing the objective function subject to the constraint at each iteration. For a constraint named `ConName` the numerical values of the shadow price is returned by simple adding a suffix to the name of the constraint such as `ConName.dual` when when using the option `option cplex_options 'sensitivity';`

For completeness note the solution method of CPLEX: namely the simplex algorithm. For quadratic problems, as the current objective function, CPLEX use either *the dual simplex* or *the primal simplex* algorithm. To fully explain these algorithms are beyond the scope of this thesis. Put in short the simplex algorithm search for a optimal solution along the edges of a polygon[2] defined by the constraints in the problem. First the problem is transformed to one of standard form by introducing artificial slack variables in the objective and the constraints[3]. After solving the problem CPLEX "translates" the findings back in to our notation.

[2] Polygon: Two dimensional region bounded by straight lines.(Adams, 2003, p. 309)
[3] Standard form: A optimization problem with equality in the constraints.

Appendix B

AMPL Programs for *Algorithm 1*

B.1 Deterministic Case - *Algorithm 1*

Model

```
# TIMEFRAME & SIZE OF SETS:
param H > 0;
set T default 0..H;
param n;
param PreStep;
set I default 1..H;
param delta{t in T} default 0;
param stopIF default 0;
# SIZE OF CAPACITY:
param K_under >= 0;
param K_over > 0;
param K{t in T} default 0;
# COST OF PRODUCTION (parameters):
param A > 0;
param B > 0;
# COST OF CAPACITY (parameters):
param r >= 0;
param d >= 0;
param f >= 0;
# DEMAND (parameters):
param a > 0;
param b  > 0;
```

VECTOR TO STORE VALUES:
```
param Profit_ve{t in T} default 0;
param Q_ve{t in T} default 0;
param gradient{t in T} default 0;
```
VARIABLE (short term choice):
```
var q>= 0;
```
DEMAND (function):
```
var Price = a - b*q;
```
COST (function):
```
var Cost = A*q + B*q*q;
```
DUAL VALUE (from AMPL):
```
param KCon_dual{t in T} default 0;
```
OBJECTIVE:
```
maximize profit: Price*q - Cost - (r+d+f)*K[n];
```
CONSTRAINTS:
```
subject to CONST: q <= K[n];
```
LABEL FOR PROBLEM (used in solve-command):
```
problem OBJ: q, profit, CONST;
```

Data

MAXIMAL NUMBER OF ITERATIONS
```
param H := 100;
```
CONVERGENCE REQUIREMENT:
```
param stopIF := 0.0002;
```
SIZE OF CAPACITY:
```
param K_under := 0;
param K_over := 30;
```
COST OF PRODUCTION (parameters):
```
param A := 6;
param B := 0.5;
```
COST OF CAPACITY (parameters):
```
param r := 0.15;
param d := 0.10;
param f := 3.75;
```
DEMAND (parameters):
```
param a := 150;
param b  := 2;
```
INITIAL CAPACITY
```
let K[0] := 0;
```

Run file

```
# RESET OF SYSTEM AND DATA:
reset data; reset options; reset;
# SETTINGS FOR LOG FILE AND OUTPUT DIRECTORY:
option log_file "C:\...\DETNL_LOG.tmp";
option outdir "C:\...\DET\";
cd "C:\...\DET\";
# READ IN MODEL AND DATA FROM FILES:
model DetNL.mod;
data DetNL.dat;
# SOLVER SELECTION AND SETTING OF OPTIONS:
cd "C:\Program Files\amplcml";
option solver cplex;
option cplex_options 'sensitivity';
option solver_msg 0;
suffix up OUT; suffix down OUT; suffix current OUT;
# FOR LOOP FOR REPEATED ITERATIONS:
for {t in T}  {  # Start of for-loop. # For each t in T until convergence, do:
  let n := t;
  let delta[n] := 1 / (5 + n);
  solve OBJ;
  let Q_ve[n] := q;
  let Profit_ve[n] := profit;
  let KCon_dual[n] := KCONST.dual;
  let gradient[n] := KCon_dual[n]-(r+d+f);
  let K[n+1] := K[n] + delta[n]*gradient[n];
  if K[n+1] < K_under then let K[n+1] := K_under;
  if K[n+1] > K_over then let K[n+1] := K_over;
  if n >= PreStep
     then if abs(K[n+1]-K[n]) <= stopIF then break;
     else continue
  } # End of for-loop.
```

B.2 Random Demand Cases - *Algorithm 1* (all types)

The AMPL program in this section is valid for the random demand case, that is for the case using the notation $\theta := s$ in eq. (3.4). The constraint in this case is given by eq. (3.6), denoted as constraint (1) in the stochastic objective summarized in eq. (3.21). Tree different types of demand uncertainty is modeled:

- Type 1 given by eq. (3.10) assuming $E(s) = 0$, that is $s \sim \mathcal{U}\,(-0.5, 0.5)$.
- Type 2 given by eq. (3.11) assuming $E(s) = 1.5$, that is $s \sim \mathcal{U}\,(0, 3)$.
- Type 3 given by eq. (3.12) assuming $E(s) = 1$, that is $s \sim \mathcal{U}\,(0.5, 1.5)$.

The AMPL code for the random demand cases is given below. The difference between the cases (in terms of the coding) is the definition of the demand function which is given as tree alternative formulations for `var Price :=`... in the file "RandDemMod.mod". Also the range for the random factor varies, and is obtained by changing the values for `param s_under :=`... & `param s_under :=`... in the file "RandDemData.dat".

Model

```
# TIMEFRAME & SIZE OF SETS:
param H > 0;
set T default 0..H;
param n;
param PreStep;
set I default 1..H;
param delta{t in T} default 0;
param stopIF default 0;
# SIZE OF CAPACITY:
param K_under >= 0;
param K_over > 0;
param K{t in T} default 0;
# COST OF PRODUCTION (parameters):
param A > 0;
param B > 0;
# COST OF CAPACITY (parameters):
param r >= 0;
param d >= 0;
param f >= 0;
```

```
# DEMAND (parameters):
param a > 0;
param b  > 0;
# SET SEED FOR RANDOM NUMBERS:
param loop > 0;
param X integer >= 0;
set LOOP = 1..loop;
param randseed{l in LOOP} default 0;
# RANDOM VALUE (parameters, vector and generation):
param s_under;
param s_over;
param s{t in T} default 0;
param s_nr default Uniform(s_under, s_over);
# VECTOR TO STORE VALUES:
param Profit_ve{t in T} default 0;
param Q_ve{t in T} default 0;
param gradient{t in T} default 0;
# VARIABLE (short term choice):
var q>= 0;
# DEMAND FUNCTION:
# Use ONE of the var Price := definitions according to type.
# For TYPE 1 uncertainty given by eq. (3.10) use:
var Price = a - b*q + s_nr;
# For TYPE 2 uncertainty given by eq. (3.11) use:
var Price = a - (b+s_nr)*q;
# For TYPE 3 uncertainty given by eq. (3.12) use.
var Price = (a - b*q)*s_nr;
# COST (function):
var Cost = A*q + B*q*q;
# DUAL VALUE (from AMPL):
param KCon_dual{t in T} default 0;
# OBJECTIVE:
maximize profit: Price*q - Cost - (r+d+f)*K[n];
# CONSTRAINTS:
subject to KCONST: q <= K[n];
# LABEL FOR PROBLEM (used in solve-command):
problem OBJ: q, profit, KCONST;
```

Data

MAXIMAL NUMBER OF ITERATIONS
```
param H := 5000;
```
NUMBER OF RANDOM SEEDS:
```
param loop := 5;
```
NUMBER OF STEPS PRIOR TO FIRST CHECK:
```
param PreStep := 10;
```
CONVERGENCE REQUIREMENT:
```
param stopIF := 0.002;
```
SIZE OF CAPACITY:
```
param K_under := 0;
param K_over := 30;
```
COST OF PRODUCTION (parameters):
```
param A := 6;
param B := 0.5;
```
COST OF CAPACITY (parameters):
```
param r := 0.15;
param d := 0.10;
param f := 3.75;
```
DEMAND (parameters):
```
param a := 150;
param b  := 2;
```
RANDOM DEMAND PARAMETERS:

Define **ONE PAIR** of `param s_under :=` & `param s_under :=` according to type.

For TYPE 1, eq. (3.10) assuming $s \sim \mathcal{U}(-1, 1)$ use:
```
param s_under := -1;
param s_over := 1;
```
For TYPE 2, eq. (3.11) assuming $s \sim \mathcal{U}(0, 3)$ use:
```
param s_under := 0;
param s_over := 3;
```
For TYPE 3, eq. (3.12) assuming $s \sim \mathcal{U}(0.5, 1.5)$ use:
```
param s_under := 0.5;
param s_over := 1.5;
```
INITIAL CAPACITY
```
let K[0] := 0;
```

Run-file

```
# RESET OF SYSTEM AND DATA:
reset data; reset options; reset;
# SETTINGS FOR LOG FILE AND OUTPUT DIRECTORY:
option log_file "C:\...\T1NL_LOG.tmp";
option outdir "C:\...\TYPE1\";
cd "C:\...\TYPE1\";
# READ IN MODEL AND DATA FROM FILES:
model RandDemMod.mod;
data RandDemData.dat;
# SOLVER SELECTION AND SETTING OF OPTIONS:
cd "C:\Program Files\amplcml";
option solver cplex; option cplex_options 'sensitivity';
option solver_msg 0;
suffix up OUT; suffix down OUT; suffix current OUT;
# SET SEED FOR RANDOM NUMBERS:
param l := 2;
let X := l;
let randseed[l] := X;
option randseed (X);
# FOR LOOP FOR REPEATED ITERATIONS:
for {t in T}  {
# For each t in T until convergence, do:
  let n := t;
  reset data s_nr;
  let s[n] := s_nr;
  let delta[n] := 1 / (5 + n);
  solve OBJ;
  let Q_ve[n] := q;
  let Profit_ve[n] := profit;
  let KCon_dual[n] := KCONST.dual;
  let gradient[n] := KCon_dual[n]-(r+d+f);
  let K[n+1] := K[n] + delta[n]*gradient[n];
  if K[n+1] < K_under then let K[n+1] := K_under;
  if K[n+1] > K_over then let K[n+1] := K_over;
  if n >= PreStep
     then if abs(K[n+1]-K[n]) <= stopIF then break;
     else continue
  } # End of for-loop.
```

B.3 Random Inflow Case - *Algorithm 1*

The AMPL program in this section is valid for the random inflow case, that is for the cases using the notation $\theta := R$ in eq. (3.4). The constraint in this case is given by eq. (3.7), denoted as constraint (2) in the stochastic objective summarized in eq. (3.21).

Model

```
# TIMEFRAME & SIZE OF SETS:
param H > 0;
set T default 0..H;
param n;
param PreStep;
set I default 1..H;
param delta{t in T} default 0;
param stopIF default 0;
# SIZE OF CAPACITY:
param K_under >= 0;
param K_over > 0;
param K{t in T} default 0;
# COST OF PRODUCTION (parameters):
param A > 0;
param B > 0;
# COST OF CAPACITY (parameters):
param r >= 0;
param d >= 0;
param f >= 0;
# DEMAND (parameters):
param a > 0;
param b  > 0;
# SET SEED FOR RANDOM NUMBERS:
param loop > 0;
param X integer >= 0;
set LOOP = 1..loop;
param randseed{l in LOOP} default 0;
# RANDOM VALUE (parameters, vector and generation):
param R_under;
param R_over;
```

```
param R{t in T} default 0;
param R_nr default Uniform(R_under, R_over);
# VECTOR TO STORE VALUES:
param Profit_ve{t in T} default 0;
param Q_ve{t in T} default 0;
param gradient{t in T} default 0;
param ACdp{t in T} default 0;
param minst{t in T} := min(R[t],K[t]);
# VARIABLE (short term choice):
var q>= 0;
# DEMAND (function):
var Price = a - b*q;
# COST (function):
var Cost = A*q + B*q*q;
# DUAL VALUES (from AMPL):
param RCon_dual{t in T} default 0;
param KCon_dual{t in T} default 0;
# OBJECTIVE:
maximize profit: Price*q - Cost - (r+d+f)*K[n];
# CONSTRAINTS:
subject to KCONST: q <= K[n];
subject to RCONST: q <= R[n];
# LABEL FOR PROBLEM (used in solve-command):
problem TYPE1: q, profit, RCONST, KCONST;
```

Data

```
# MAXIMAL NUMBER OF ITERATIONS
param H := 5000;
# NUMBER OF RANDOM SEEDS:
param loop := 5;
# NUMBER OF STEPS PRIOR TO FIRST CHECK:
param PreStep := 10;
# CONVERGENCE REQUIREMENT:
param stopIF := 0.002;
# SIZE OF CAPACITY:
param K_under := 0;
param K_over := 30;
```

```
# RANDOM INFLOW PARAMETERS:
param R_under := 0;
param R_over := 30;
# COST OF PRODUCTION (parameters):
param A := 6;
param B := 0.5;
# COST OF CAPACITY (parameters):
param r := 0.15;
param d := 0.10;
param f := 3.75;
# DEMAND (parameters):
param a := 150;
param b  := 2;
# INITIAL CAPACITY
let K[0] := 0;
```

Run-file

```
# RESET OF SYSTEM AND DATA:
reset data; reset options; reset;
# SETTINGS FOR LOG FILE AND OUTPUT DIRECTORY:
option log_file "C:\...\RDNL_LOG.tmp";
option outdir "C:\...\RDinput\";
cd "C:\...\RDinput\";
# READ IN MODEL AND DATA FROM FILES:
model RDinputMod.mod;
data RDinputDat.dat;
# SOLVER SELECTION AND SETTING OF OPTIONS:
cd "C:\Program Files\amplcml";
option solver cplex; option cplex_options 'sensitivity';
option solver_msg 0;
suffix up OUT; suffix down OUT; suffix current OUT;
# SET SEED FOR RANDOM NUMBERS:
param l := 2;
let X := l;
let randseed[l] := X;
option randseed (X);
```

```
# FOR LOOP FOR REPEATED ITERATIONS:
for {t in T}  { # Start of for-loop.
# For each t in T until convergence, do:
   let n := t;
   reset data R_nr;
   let R[n] := R_nr;
   let delta[n] := 1 / (5 + n);
   solve OBJ;
   let Q_ve[n] := q;
   let Profit_ve[n] := profit;
   let RCon_dual[n] := RCONST.dual;
   let KCon_dual[n] := KCONST.dual;
   let gradient[n] := KCon_dual[n]-(r+d+f);
   let K[n+1] := K[n] + delta[n]*gradient[n];
   if K[n+1] < K_under then let K[n+1] := K_under;
   if K[n+1] > K_over then let K[n+1] := K_over;
   if n >= PreStep
      then if abs(K[n+1]-K[n]) <= stopIF then break;
      else continue
  } # End of for-loop.
display K, Q_ve, R, Profit_ve, delta>($outdir&"RDNL"&X&".out");
display KCon_dual, RCon_dual, gradient>($outdir&"RDNL"&X&".out");
```

B.4 Tables of Results - *Algorithm 1*

Deterministic Result - *Algorithm 1*

Table B.1 results from the AMPL program in Section B.1 using data from Table 5.1 for the $Ex1$-experiment described in Table 5.1, $(K_0 = 0)$ for the deterministic case.

	Deterministic Results: $K_0 = 0$, $\delta_t = \frac{1}{10+t}$				
t	$\hat{K}_t$	q_t	λ_t	$\pi(q_t, \hat{K})$	$\frac{\Delta\pi(\hat{K})}{\Delta\hat{K}}$
0	0.0000	0.0000	144.0000	0.0000	140.0000
1	14.0000	14.0000	74.0000	1470.0000	70.0000
2	20.3636	20.3636	42.1818	1814.2149	38.1818
3	23.5455	23.5455	26.2727	1910.3926	22.2727
4	25.2587	25.2587	17.7063	1941.2137	13.7063
5	26.2378	26.2378	12.8112	1952.2363	8.8112
6	26.8252	26.8252	9.8741	1956.5495	5.8741
7	27.1923	27.1923	8.0385	1958.3691	4.0385
8	27.4299	27.4299	6.8507	1959.1874	2.8507
9	27.5882	27.5882	6.0588	1959.5761	2.0588
10	27.6966	27.6966	5.5170	1959.7699	1.5170
11	27.7724	27.7724	5.1378	1959.8705	1.1378
12	27.8266	27.8266	4.8669	1959.9249	0.8669
13	27.8660	27.8660	4.6699	1959.9551	0.6699
...	...	...	...	...	...
29	27.9930	27.9930	4.0351	1959.9999	0.0351
30	27.9939	27.9939	4.0306	1959.9999	0.0306
31	27.9946	27.9946	4.0268	1959.9999	0.0268
32	27.9953	27.9953	4.0235	1959.9999	0.0235
33	27.9959	27.9959	4.0207	1960.0000	0.0207
34	27.9963	27.9963	4.0183	1960.0000	0.0183
35	27.9968	27.9968	4.0162	1960.0000	0.0162
36	27.9971	27.9971	4.0144	1960.0000	0.0144
37	27.9974	27.9974	4.0129	1960.0000	0.0129
38	27.9977	27.9977	4.0115	1960.0000	0.0115
39	27.9979	27.9979	4.0103	1960.0000	0.0103
40	27.9982	27.9982	4.0093	1960.0000	0.0093
41	27.9983	.	.	.	.

Table B.1: *Algorithm 1* - Deterministic Numerical Results

Random Demand Type 1 Result - *Algorithm 1*

Table B.2 results from the AMPL program in Section B.2 using data from Table 5.1 for the $Ex1$-experiment described in Table 5.1, for the **Type 1** uncertainty.

	Type 1 Results: $K_0 = 0$, $\delta_t = \frac{1}{10+t}$ and seed:= 1				
t	$\hat{K}_t$	q_t	λ_t	s_t	$\frac{\Delta\pi(\hat{K})}{\Delta\hat{K}}$
0	0.0000	0.0000	144.2180	0.2184	140.2180
1	14.0218	14.0218	73.2705	-0.6203	69.2705
2	20.3192	20.3192	43.2480	0.8438	39.2480
3	23.5898	23.5898	26.9652	0.9143	22.9652
4	25.3564	25.3564	16.4296	-0.7885	12.4296
5	26.2442	26.2442	13.2072	0.4282	9.2072
6	26.8580	26.8580	9.8130	0.1031	5.8130
7	27.2213	27.2213	7.4196	-0.4737	3.4196
8	27.4225	27.4225	6.5868	-0.3008	2.5868
9	27.5662	27.5662	5.9835	-0.1855	1.9835
10	27.6706	27.6706	5.9775	0.3304	1.9775
11	27.7695	27.7695	5.3043	0.1516	1.3043
12	27.8316	27.8316	5.7262	0.8840	1.7262
13	27.9100	27.9100	4.1769	-0.2730	0.1769
14	27.9177	27.9177	3.4176	-0.9938	-0.5825
...	...	...	...	...	...
45	27.9849	27.9849	4.0949	0.0195	0.0949
46	27.9866	27.9866	3.5965	-0.4703	-0.4035
47	27.9794	27.9794	4.6754	0.5726	0.6754
48	27.9913	27.9913	4.3948	0.3512	0.3948
49	27.9981	27.9981	3.5291	-0.4805	-0.4709
50	27.9901	27.9901	4.2877	0.2383	0.2877
51	27.9949	27.9949	3.8659	-0.1596	-0.1341
52	27.9927	27.9927	3.2092	-0.8273	-0.7908
53	27.9800	27.9800	4.7695	0.6692	0.7695
54	27.9922	27.9922	4.5336	0.4945	0.5336
55	28.0005	28.0005	4.6892	0.6917	0.6892
56	28.0111	28.0111	3.7459	-0.1986	-0.2541
57	28.0073	28.0073	4.0130	0.0493	0.0130
58	28.0075	.	.	.	.

Table B.2: *Algorithm 1* - Random Demand Type 1 Numerical Results

Random Demand Type 2 Result - *Algorithm 1*

Table B.3 results from the AMPL program in Section B.2 using data from Table 5.1 for the $Ex1$-experiment described in Table 5.1, for the **Type 2** uncertainty defined by eq. (3.11) for $s \sim \mathcal{U}(0,3)$.

	Type 2 Results: $K_0 = 0$, $\delta_t = \frac{1}{5+t}$ and seed:= 1				
t	$\hat{K}_t$	q_t	λ_t	s_t	$\frac{\Delta\pi(\hat{K})}{\Delta\hat{K}}$
0	0.0000	0.0000	144.0000	1.8276	140.0000
1	28.0000	23.4557	0.0000	0.5696	-4.0000
2	27.3333	13.6735	0.0000	2.7657	-4.0000
3	26.7619	13.4042	0.0000	2.8715	-4.0000
4	26.2619	25.5575	0.0000	0.3172	-4.0000
5	25.8175	15.5095	0.0000	2.1423	-4.0000
6	25.4175	17.3302	0.0000	1.6546	-4.0000
7	25.0538	21.8885	0.0000	0.7894	-4.0000
8	24.7205	20.2885	0.0000	1.0488	-4.0000
9	24.4128	19.3458	0.0000	1.2217	-4.0000
10	24.1271	16.0155	0.0000	1.9956	-4.0000
11	23.8604	17.0317	0.0000	1.7274	-4.0000
12	23.6104	13.5184	0.0000	2.8261	-4.0000
13	23.3751	20.0525	0.0000	1.0906	-4.0000
14	23.1529	23.1529	27.8064	0.0093	23.8064
...	...	...	...	...	...
666	22.2670	15.5880	0.0000	2.1190	-4.0000
667	22.2611	19.2242	0.0000	1.2453	-4.0000
668	22.2551	15.2941	0.0000	2.2077	-4.0000
669	22.2492	14.4254	0.0000	2.4912	-4.0000
670	22.2432	20.7275	0.0000	0.9736	-4.0000
671	22.2373	22.2373	6.9722	0.5810	2.9722
672	22.2417	15.9228	0.0000	2.0218	-4.0000
673	22.2358	17.2709	0.0000	1.6689	-4.0000
674	22.2299	22.1654	0.0000	0.7483	-4.0000
675	22.2240	15.3677	0.0000	2.1851	-4.0000
676	22.2181	18.7671	0.0000	1.3365	-4.0000
677	22.2122	16.2381	0.0000	1.9340	-4.0000
678	22.2064	22.2064	4.0890	0.6502	0.0890
679	22.2065	.	.	.	.

Table B.3: *Algorithm 1* - Random Demand Type 2 Numerical Results

Random Demand Type 3 Result - *Algorithm 1*

Table B.4 results from the AMPL program in Section B.2 using data from Table 5.1 for the $Ex1$-experiment described in Table 5.1, for the **Type 3** uncertainty defined by eq. (3.12) for $s \sim \mathcal{U}\,(0.5, 1.5)$.

	Type 3 Results: $K_0 = 0$, $\delta_t = \frac{1}{10+t}$ and seed:$= 1$				
t	$\hat{K}_t$	q_t	λ_t	s_t	$\frac{\Delta\pi(\hat{K})}{\Delta\hat{K}}$
0	0.0000	0.0000	160.3810	1.1092	156.3810
1	30.0000	25.9293	0.0000	0.6899	-4.0000
2	29.3333	29.3333	11.1151	1.4219	7.1151
3	30.0000	30.0000	7.7147	1.4572	3.7147
4	30.0000	24.7915	0.0000	0.6057	-4.0000
5	29.5556	29.5556	3.0260	1.2141	-0.9740
6	29.4582	29.1445	0.0000	1.0515	-4.0000
7	29.0945	26.7660	0.0000	0.7631	-4.0000
8	28.7612	27.6101	0.0000	0.8496	-4.0000
9	28.4535	28.1027	0.0000	0.9072	-4.0000
10	28.1678	28.1678	9.3283	1.1652	5.3283
11	28.5230	28.5230	4.1071	1.0758	0.1071
12	28.5297	28.5297	17.2118	1.4420	13.2118
13	29.3069	27.7337	0.0000	0.8635	-4.0000
14	29.0846	23.0595	0.0000	0.5031	-4.0000
...	...	...	...	...	...
274	28.8816	28.4431	0.0000	0.9507	-4.0000
275	28.8672	28.8672	0.8232	1.0336	-3.1768
276	28.8559	26.4722	0.0000	0.7361	-4.0000
277	28.8416	26.5732	0.0000	0.7453	-4.0000
278	28.8275	24.0471	0.0000	0.5584	-4.0000
279	28.8133	28.8133	2.5135	1.0743	-1.4865
280	28.8081	23.8511	0.0000	0.5468	-4.0000
281	28.7941	28.7941	8.6701	1.2481	4.6701
282	28.8104	24.2925	0.0000	0.5734	-4.0000
283	28.7965	24.8603	0.0000	0.6104	-4.0000
284	28.7826	23.1953	0.0000	0.5102	-4.0000
285	28.7687	28.7687	14.6010	1.4136	10.6010
286	28.8053	28.8053	3.9553	1.1145	-0.0447
287	28.8051	.	.	.	.

Table B.4: *Algorithm 1* - Random Demand Type 3 Numerical Results

Random Inflow Result - *Algorithm 1*

Table B.5 results from the AMPL program in Section B.3 using data from Table 5.1 for the $Ex1$-experiment described in Table 5.1, for demand in eq. (3.8) for $R \sim \mathcal{U}(0, 30)$ and for the constraint in eq. (3.7).

	Random Inflow results: $K_0 = 0$, $\delta_t = \frac{1}{10+t}$ & seed:= 1				
t	$\hat{K}_t$	q_t	λ_t	R_t	$\frac{\Delta\pi(\hat{K})}{\Delta\hat{K}}$
0	0.0000	0.0000	144.0000	18.2763	140.0000
1	14.0000	5.6962	0.0000	5.6962	-4.0000
2	13.6364	13.6364	75.8182	27.6568	71.8182
3	19.6212	19.6212	45.8939	28.7147	41.8939
4	22.8438	3.1718	0.0000	3.1718	-4.0000
5	22.5581	21.4232	0.0000	21.4232	-4.0000
6	22.2914	16.5460	0.0000	16.5460	-4.0000
7	22.0414	7.8941	0.0000	7.8941	-4.0000
8	21.8061	10.4881	0.0000	10.4881	-4.0000
9	21.5839	12.2174	0.0000	12.2174	-4.0000
10	21.3734	19.9564	0.0000	19.9564	-4.0000
11	21.1734	17.2742	0.0000	17.2742	-4.0000
12	20.9829	20.9829	39.0854	28.2606	35.0854
13	22.5777	10.9057	0.0000	10.9057	-4.0000
14	22.4038	0.0927	0.0000	0.0927	-4.0000
...	...	...	...	...	...
1978	24.4990	10.6175	0.0000	10.6175	-4.0000
1979	24.4970	6.4970	0.0000	6.4970	-4.0000
1980	24.4950	24.4950	21.5252	25.3370	17.5252
1981	24.5038	18.4450	0.0000	18.4450	-4.0000
1982	24.5018	11.7213	0.0000	11.7213	-4.0000
1983	24.4998	3.2222	0.0000	3.2222	-4.0000
1984	24.4977	20.2267	0.0000	20.2267	-4.0000
1985	24.4957	12.6624	0.0000	12.6624	-4.0000
1986	24.4937	23.5631	0.0000	23.5631	-4.0000
1987	24.4917	19.7862	0.0000	19.7862	-4.0000
1988	24.4897	14.2602	0.0000	14.2602	-4.0000
1989	24.4877	11.2491	0.0000	11.2491	-4.0000
1990	24.4857	17.1967	0.0000	17.1967	-4.0000
1991	24.4837	.	.	.	.

Table B.5: *Algorithm 1* - Random Inflow Numerical Results

Appendix C

AMPL Programs for *Algorithm 2*

C.1 Model-file Changes - *Algorithm 2* (all cases)

Parameters to separate between *expectation* and *observation* of both price and slope must be **added** to the original models given in Appendix B. (Section B.1, B.2 and B.3 respectively.) Also the demand function should be changed according to the agent's beliefs about the price he believe he obtain.

ADD ADDITIONAL PARAMETERS FOR EXPECTATION.
Define p_t, the expected price

```
param phat{t in T}>= 0;
```

Define dp_t, expected change in price.

```
param dphat{t in T}>=0;
```

ADD ADDITIONAL PARAMETERS FOR ACTUAL OBSERVATIONS.
Define $\hat{p}_t$, actual observed price i period t:

```
param ACp{t in T} default 0;
```

Define $\hat{dp}_t$, actual observed change in price within period t:

```
param ACdp{t in T} default 0;
```

CHANGE THE DEMAND FUNCTION TO REFLECT AGENT'S BELIEVED PRICE:
CHANGE `var Price` in the original models of Section B.1, B.2 and B.3 to:

```
var Price = phat[n]-0.5*dphat[n]*q;
```

C.2 Data-file Changes - *Algorithm 2* (all cases)

In **addition** to the data given in Appendix B initial values for the added parameters accounting for the agent's expectations must be given. The following parameter values is therefore **added** to the data-files in Section B.1, Section B.2 and Section B.3.

```
# ADDED: ADDITIONAL PARAMETERS FOR INITIAL EXPECTATION.
# Initial expectation about price: p0
let phat[0] := 150;
# Initial expectation about slope: dp0
let dphat[0] := 12;
```

Several combinations of initial values for price, slope and capacity have been tested during the experiments. The random seed also varies over the experiments where random values are included. The combinations of initial values and seeds for the pseudo random numbers are listed in Table 5.1 given in Section 5.1 of Chapter 5.

For example, the experiment $Ex3$ given by the red dotted line in Figure 5.7 is simulated by running the model, data and run-file consistent with a **Type 1 uncertainty** for initial capacity $\hat{K}_0 := 20$, with the initial believe about price and slope set to $p_0 := 150$ and $dp_0 := 12$ respectively. The random demand generated in this particular experiment results from pseudo random numbers from the set seed command `param l:=3`.

In each of the experiments, the values for the added parameters defined above is set according to Table 5.1 in the AMPL program. This give rise to four different experiments in each case for each of the two algorithms, and hence a total of 40 experiments have been conducted. The results from all of the 40 experiments are summarized in Table 5.3 at the end of Chapter 5.

Tables of the numerical results for the following *Algorithm 2*-experiments (for $Ex1$) is also included:

- Deterministic Case: Table C.1
- Random Demand Type 1: Table C.2
- Random Demand Type 2: Table C.3
- Random Demand Type 3: Table C.4
- Random Inflow: Table C.5

C.3 Run-file Changes - *Algorithm 2*

To accommodate for the agent to learn both price and slope of price **additional** iterative components must be **added inside the for-loops**. The original run-files are given in Section B.1, Section B.2 or Section B.3 respectively. **Note:** Only the for-loop change, and hence all other components of the original run-files should be as in the original files.

Deterministic Run-file Changes

In Section B.1 the original run-file for the deterministic case is given. The modified for-loop is given below. Only the part between the dots (...) is changed.

```
# FOR LOOP FOR REPEATED ITERATIONS:
for {t in T}  {  # Start of for-loop.
  let n := t;
  let delta[n] := 1 / (5 + n);
  solve OBJ;
  let Q_ve[n] := q;
  let Profit_ve[n] := profit;
...
# ADDED: UPDATE ACTUAL PRICE & SLOPE.
  let ACp[n] := a-b*Q_ve[n];
  let ACdp[n] := b;
# ADDED: UPDATE AGENT'S BELIEFS.
  let phat[n+1] := (1-delta[n])*phat[n] + delta[n]*ACp[n];
  let dphat[n+1] := (1-delta[n])*dphat[n] + delta[n]*ACdp[n];
...
  let KCon_dual[n] := KCONST.dual;
  let gradient[n] := KCon_dual[n]-(r+d+f);
  let K[n+1] := K[n] + delta[n]*gradient[n];
  if K[n+1] < K_under then let K[n+1] := K_under;
  if K[n+1] > K_over then let K[n+1] := K_over;
  if n >= PreStep
   then if abs(K[n+1]-K[n]) <= stopIF
   and abs(phat[n+1]-phat[n]) <= stopIF
   and abs(dphat[n+1]-dphat[n]) <= stopIF
  then break; else continue
  } # End of for-loop.
```

Random demand Run-file Changes

In Section B.2 the original run-file for the random demand cases is given. The modified for-loop is given below. Only the part between the dots (...) is changed.

```
# FOR LOOP FOR REPEATED ITERATIONS:
for {t in T}  {  # Start of for-loop
  let n := t;
  reset data s_nr;
  let s[n] := s_nr;
  let delta[n] := 1 / (5 + n);
  solve OBJ;
  let Q_ve[n] := q;
  let Profit_ve[n] := profit;
...
# ADDED: UPDATE ACTUAL PRICE ACCORDING TO TYPE. (Use one of the below)
  let ACp[n] := a-b*Q_ve[n]+s_nr;  # For Type 1
  let ACp[n] := a-(b+s_nr)*Q_ve[n]; # For Type 2
  let ACp[n] := (a-b*Q_ve[n])*s_nr); # For Type 3
# ADDED: UPDATE ACTUAL SLOPE ACCORDING TO TYPE. (Use one of the below)
  let ACdp[n] := b;  # For Type 1
  let ACdp[n] := b+s_nr; # For Type 2
  let ACdp[n] := b*s_nr; # For Type 3
# ADDED: UPDATE AGENT'S BELIEFS (all Types).
  let phat[n+1] := (1-delta[n])*phat[n] + delta[n]*ACp[n];
  let dphat[n+1] := (1-delta[n])*dphat[n] + delta[n]*ACdp[n];
...
  let KCon_dual[n] := KCONST.dual;
  let gradient[n] := KCon_dual[n]-(r+d+f);
  let K[n+1] := K[n] + delta[n]*gradient[n];
  if K[n+1] < K_under then let K[n+1] := K_under;
  if K[n+1] > K_over then let K[n+1] := K_over;
  if n >= PreStep
   then if abs(K[n+1]-K[n]) <= stopIF
   and abs(phat[n+1]-phat[n]) <= stopIF
   and abs(dphat[n+1]-dphat[n]) <= stopIF
  then break; else continue
  } # End of for-loop.
```

Random inflow Run-file Changes

In Section B.3 the original run-file for the random inflow cases is given. The new for-loop for learning price and slope is given below. The part between the markers . . . is changed, the rest of the code should be as it is given in the original file.

```
# FOR LOOP FOR REPEATED ITERATIONS:
for {t in T}  { # Start of for-loop.
  let n := t;
  reset data R_nr;
  let R[n] := R_nr;
  let delta[n] := 1 / (5 + n);
  solve OBJ;
  let Q_ve[n] := q;
  let Profit_ve[n] := profit;
...
# ADDED: UPDATE ACTUAL PRICE & SLOPE.
  let ACp[n] := a-b*Q_ve[n];
  let ACdp[n] := b;
# ADDED: UPDATE AGENT'S BELIEFS.
  let phat[n+1] := (1-delta[n])*phat[n] + delta[n]*ACp[n];
  let dphat[n+1] := (1-delta[n])*dphat[n] + delta[n]*ACdp[n];
...
  let RCon_dual[n] := RCONST.dual;
  let KCon_dual[n] := KCONST.dual;
  let gradient[n] := KCon_dual[n]-(r+d+f);
  let K[n+1] := K[n] + delta[n]*gradient[n];
  if K[n+1] < K_under then let K[n+1] := K_under;
  if K[n+1] > K_over then let K[n+1] := K_over;
  if n >= PreStep
   then if abs(K[n+1]-K[n]) <= stopIF
   and abs(phat[n+1]-phat[n]) <= stopIF
   and abs(dphat[n+1]-dphat[n]) <= stopIF
  then break; else continue
  } # End of for-loop.
```

C.4 Tables of Results - *Algorithm 2*

Deterministic Result - *Algorithm 2*

Table C.1 results from the run-file in Section B.1 modified with the changes described in Section C.3 using data from Section B.1 and C.2, for $Ex1$ in Table 5.1.

	Deterministic Result (*Alg.2*): $K_0 = 0$, $\delta_t = \frac{1}{5+t}$					
t	$\hat{K}_t$	q_t	λ_t	$\frac{\Delta\pi(\hat{K})}{\Delta\hat{K}}$	p_t	dp_t
0	0.00000	0.00000	44.00000	40.00000	50.0000	1.0000
1	8.00000	8.00000	46.40000	42.40000	70.0000	1.2000
2	15.06667	15.06667	39.51111	35.51111	80.6667	1.3333
3	20.13968	20.13968	31.35601	27.35601	86.2667	1.4286
4	23.55918	23.55918	24.30045	20.30045	89.1984	1.5000
5	25.81479	25.81479	18.74764	14.74764	90.7188	1.5556
6	27.28955	27.28955	14.53110	10.53110	91.4839	1.6000
7	28.24693	28.24693	11.37267	7.37267	91.8418	1.6364
8	28.86132	28.86132	9.01703	5.01703	91.9805	1.6667
9	29.24724	29.24724	7.26080	3.26080	92.0034	1.6923
10	29.48015	29.48015	5.95024	1.95024	91.9678	1.7143
11	29.61017	29.61017	4.97147	0.97147	91.9059	1.7333
12	29.67089	29.67089	4.24060	0.24060	91.8355	1.7500
13	29.68504	29.68504	3.69588	-0.30412	91.7663	1.7647
14	29.66814	29.66814	3.29164	-0.70836	91.7032	1.7778
…	…	…	…	…	…	…
746	27.99822	27.99822	4.00340	0.00340	93.8487	1.9947
747	27.99822	27.99822	4.00339	0.00339	93.8489	1.9947
748	27.99823	27.99823	4.00338	0.00338	93.8491	1.9947
749	27.99823	27.99823	4.00338	0.00338	93.8493	1.9947
750	27.99824	27.99824	4.00337	0.00337	93.8495	1.9947
751	27.99824	27.99824	4.00337	0.00337	93.8498	1.9947
752	27.99825	27.99825	4.00336	0.00336	93.8500	1.9947
753	27.99825	27.99825	4.00335	0.00335	93.8502	1.9947
754	27.99825	27.99825	4.00335	0.00335	93.8504	1.9947
755	27.99826	27.99826	4.00334	0.00334	93.8506	1.9947
756	27.99826	27.99826	4.00334	0.00334	93.8508	1.9947
757	27.99827	27.99827	4.00333	0.00333	93.8510	1.9947
758	27.99827	27.99827	4.00332	0.00332	93.8512	1.9948
759	27.99828	.	.	.	93.8514	1.9948

Table C.1: *Algorithm 2* - Deterministic Numerical Results

Random Demand Type 1 Result - *Algorithm 2*

Table C.2 results from the run-file in Section B.2 modified with the changes described in Section C.3 using data from Section B.2 and C.2,for Type 1 uncertainty. Initial values is given by $Ex1$ in Table 5.1.

	Type 1 results: $K_0 = 0$, $\delta_t = \frac{1}{5+t}$, seed:= 1, $p_0 = 50$, $dp_0 = 1$					
t	$\hat{K}_t$	q_t	λ_t	$\frac{\Delta\pi(\hat{K})}{\Delta\hat{K}}$	p_t	dp_t
0	0.00000	0.00000	44.00000	40.00000	50.0000	1.0000
1	8.00000	8.00000	49.32680	45.32680	73.2763	1.2437
2	15.55450	15.55450	35.21870	31.21870	76.4707	1.2664
3	20.01430	20.01430	33.82670	29.82670	89.6964	1.4917
4	23.74260	23.74260	29.13330	25.13330	98.5150	1.6695
5	26.53520	26.53520	18.98240	14.98240	94.4684	1.6186
6	28.03340	28.03340	15.11100	11.11100	96.7899	1.6996
7	29.04350	29.04350	11.49920	7.49921	96.9702	1.7363
8	29.66850	29.66850	8.07274	4.07274	94.7345	1.7188
9	29.98180	29.98180	5.90403	1.90403	93.3725	1.7173
10	30.00000	30.00000	4.81133	0.81133	92.5377	1.7242
11	30.00000	30.00000	4.42100	0.42100	93.3598	1.7646
12	30.00000	30.00000	3.91183	-0.08817	93.5762	1.7888
13	29.99480	29.99480	4.12361	0.12361	95.7059	1.8532
14	30.00000	30.00000	3.32025	-0.67975	94.7071	1.8462
...	...	...	...	...	...	...
516	27.93420	27.93420	4.04968	0.04968	93.4502	1.9856
517	27.93430	27.93430	4.07308	0.07308	93.5085	1.9869
518	27.93440	27.93440	4.06587	0.06587	93.4918	1.9865
519	27.93450	27.93450	4.04492	0.04492	93.4411	1.9854
520	27.93460	27.93460	4.06584	0.06584	93.4932	1.9865
521	27.93470	27.93470	4.04686	0.04686	93.4475	1.9856
522	27.93480	27.93480	4.04032	0.04032	93.4320	1.9852
523	27.93490	27.93490	4.01832	0.01832	93.3784	1.9841
524	27.93490	27.93490	4.03436	0.03436	93.4181	1.9849
525	27.93500	27.93500	4.01695	0.01695	93.3758	1.9840
526	27.93500	27.93500	4.03185	0.03185	93.4127	1.9848
527	27.93510	27.93510	4.02020	0.02020	93.3844	1.9842
528	27.93510	27.93510	3.99741	-0.00259	93.3286	1.9830
529	27.93510	.	.	.	93.3285	1.9830

Table C.2: *Algorithm 2* - Random Demand Type 1 Numerical Results

Random Demand Type 2 Result - *Algorithm 2*

Table C.3 results from the run-file in Section B.2 modified with the changes described in Section C.3 using data from Section B.2 and C.2, for Type 2 uncertainty. Initial values is given by $Ex1$ in Table 5.1.

	Type 2 results: $K_0 = 0$, $\delta_t = \frac{1}{5+t}$, seed:= 1, $p_0 = 50$, $dp_0 = 1$					
t	$\hat{K}_t$	q_t	λ_t	$\frac{\Delta\pi(\hat{K})}{\Delta\hat{K}}$	p_t	dp_t
0	0.00000	0.00000	44.00000	40.00000	50.0000	1.0000
1	8.00000	8.00000	43.47580	39.47580	70.0000	1.5655
2	14.57930	14.57930	34.06380	30.06380	79.9072	1.7329
3	18.87410	18.87410	14.23670	10.23670	79.9947	2.1661
4	20.15370	20.15370	0.62763	-3.37237	77.2523	2.5043
5	19.77900	19.77900	5.24617	1.24617	80.1465	2.4835
6	19.90360	19.90360	0.30269	-3.69731	78.9387	2.6494
7	19.56750	19.45750	0.00000	-4.00000	78.7862	2.7408
8	19.23420	19.23420	2.16915	-1.83085	80.1977	2.7448
9	19.09330	19.09330	3.10861	-0.89140	81.0563	2.7682
10	19.02970	19.02970	3.26277	-0.73723	81.5870	2.8006
11	18.98050	18.98050	1.42924	-2.57076	81.0788	2.8803
12	18.81980	18.81980	0.94204	-3.05796	80.9646	2.9332
13	18.64000	18.21770	0.00000	-4.00000	79.6828	3.0446
14	18.41770	18.39860	0.00000	-4.00000	80.4614	3.0471
...	...	...	...	...	...	...
828	17.55170	17.55170	3.76428	-0.23572	88.4752	3.4845
829	17.55140	17.55140	3.78045	-0.21955	88.4828	3.4841
830	17.55110	17.55110	3.80768	-0.19232	88.4961	3.4834
831	17.55090	17.55090	3.77246	-0.22754	88.4782	3.4844
832	17.55060	17.55060	3.77850	-0.22150	88.4808	3.4843
833	17.55040	17.55040	3.83716	-0.16284	88.5098	3.4827
834	17.55020	17.55020	3.88483	-0.11517	88.5334	3.4814
835	17.55000	17.55000	3.94673	-0.05327	88.5642	3.4796
836	17.55000	17.55000	3.94268	-0.05732	88.5623	3.4798
837	17.54990	17.54990	4.00380	0.00380	88.5929	3.4780
838	17.54990	17.54990	3.96201	-0.03799	88.5722	3.4792
839	17.54990	17.54990	3.93239	-0.06761	88.5576	3.4801
840	17.54980	17.54980	3.93565	-0.06435	88.5592	3.4800
841	17.54970	.	.	.	88.5593	3.4801

Table C.3: *Algorithm 2* - Random Demand Type 2 Numerical Results

Random Demand Type 3 Result - *Algorithm 2*

Table C.4 results from the run-file in Section B.2 modified with the changes described in Section C.3 using data from Section B.2 and C.2, for Type 3 uncertainty. Initial values is given by $Ex1$ in Table 5.1.

	Type 3 results: $K_0 = 0$, $\delta_t = \frac{1}{5+t}$, seed:= 1, $p_0 = 50$, $dp_0 = 1$					
t	$\hat{K}_t$	q_t	λ_t	$\frac{\Delta\pi(\hat{K})}{\Delta\hat{K}}$	p_t	dp_t
0	0.00000	0.00000	44.00000	40.00000	50.0000	1.0000
1	8.00000	8.00000	49.32680	45.32680	73.2763	1.2437
2	15.55450	15.55450	35.21870	31.21870	76.4707	1.2664
3	20.01430	20.01430	33.82670	29.82670	89.6964	1.4917
4	23.74260	23.74260	29.13330	25.13330	98.5150	1.6695
5	26.53520	26.53520	18.98240	14.98240	94.4684	1.6186
6	28.03340	28.03340	15.11100	11.11100	96.7899	1.6996
7	29.04350	29.04350	11.49920	7.49921	96.9702	1.7363
8	29.66850	29.66850	8.07274	4.07274	94.7345	1.7188
9	29.98180	29.98180	5.90403	1.90403	93.3725	1.7173
10	30.00000	30.00000	4.81133	0.81133	92.5377	1.7242
11	30.00000	30.00000	4.42100	0.42100	93.3598	1.7646
12	30.00000	30.00000	3.91183	-0.08817	93.5762	1.7888
13	29.99480	29.99480	4.12361	0.12361	95.7059	1.8532
14	30.00000	30.00000	3.32025	-0.67975	94.7071	1.8462
...	...	...	...	...	...	...
516	27.93420	27.93420	4.04968	0.04968	93.4502	1.9856
517	27.93430	27.93430	4.07308	0.07308	93.5085	1.9869
518	27.93440	27.93440	4.06587	0.06587	93.4918	1.9865
519	27.93450	27.93450	4.04492	0.04492	93.4411	1.9854
520	27.93460	27.93460	4.06584	0.06584	93.4932	1.9865
521	27.93470	27.93470	4.04686	0.04686	93.4475	1.9856
522	27.93480	27.93480	4.04032	0.04032	93.4320	1.9852
523	27.93490	27.93490	4.01832	0.01832	93.3784	1.9841
524	27.93490	27.93490	4.03436	0.03436	93.4181	1.9849
525	27.93500	27.93500	4.01695	0.01695	93.3758	1.9840
526	27.93500	27.93500	4.03185	0.03185	93.4127	1.9848
527	27.93510	27.93510	4.02020	0.02020	93.3844	1.9842
528	27.93510	27.93510	3.99741	-0.00259	93.3286	1.9830
529	27.93510	.	.	.	93.3285	1.9830

Table C.4: *Algorithm 2* - Random Demand Type 3 Numerical Results

Random Inflow - *Algorithm 2*

Table C.5 results from the run-file in Section B.3 modified with the changes described in Section C.3 using data from Section B.3 and C.2, for random inflow. Initial values is given by $Ex1$ in Table 5.1.

	Inflow results: $K_0 = 0$, $\delta_t = \frac{1}{5+t}$, seed:= 1, $p_0 = 50$, $dp_0 = 1$						
t	$\hat{K}_t$	q_t	λ_t	R_t	$\frac{\Delta\pi(\hat{K})}{\Delta\hat{K}}$	p_t	dp_t
0	0.00000	0.00000	44.000	18.27630	40.000	50.000	150.000
1	4.00000	4.00000	45.600	5.69619	41.600	60.000	142.000
2	7.78182	7.78182	44.476	27.65680	40.476	67.455	134.436
3	11.15480	11.15480	41.938	28.71470	37.938	73.036	127.690
4	14.07310	3.17177	0.000	3.17177	-4.000	77.241	143.656
5	13.78740	13.78740	43.486	21.42320	39.486	81.985	122.425
6	16.41980	16.41980	39.273	16.54600	35.273	84.681	117.160
7	18.62440	7.89405	0.000	7.89405	-4.000	86.711	134.212
8	18.38910	10.48810	0.000	10.48810	-4.000	89.505	129.024
9	18.16680	12.21740	0.000	12.21740	-4.000	91.700	125.565
10	17.95630	17.95630	42.119	19.95640	38.119	93.483	114.087
11	19.86230	17.27420	0.000	17.27420	-4.000	94.513	115.452
12	19.67180	19.67180	38.925	28.26060	34.925	95.510	110.656
13	21.25930	10.90570	0.000	10.90570	-4.000	96.198	128.189
14	21.08540	0.09266	0.000	0.09266	-4.000	97.589	149.815
...	...	...	...	...	...	...	
1979	26.59200	6.49704	0.000	6.49704	-4.000	120.393	137.006
1980	26.59000	25.33700	0.000	25.33700	-4.000	120.401	99.326
1981	26.58800	18.44500	0.000	18.44500	-4.000	120.391	113.110
1982	26.58590	11.72130	0.000	11.72130	-4.000	120.387	126.557
1983	26.58390	3.22223	0.000	3.22223	-4.000	120.390	143.556
1984	26.58190	20.22670	0.000	20.22670	-4.000	120.402	109.547
1985	26.57990	12.66240	0.000	12.66240	-4.000	120.396	124.675
1986	26.57790	23.56310	0.000	23.56310	-4.000	120.398	102.874
1987	26.57590	19.78620	0.000	19.78620	-4.000	120.390	110.428
1988	26.57390	14.26020	0.000	14.26020	-4.000	120.385	121.480
1989	26.57190	11.24910	0.000	11.24910	-4.000	120.385	127.502
1990	26.56990	17.19670	0.000	17.19670	-4.000	120.389	115.607
1991	26.56790	14.03950	0.000	14.03950	-4.000	120.386	121.921
1992	26.56590	.	.	.	.	120.387	.

Table C.5: *Algorithm 2* - Random Inflow Numerical Results

Appendix D

Calculus

Calculations - Random Demand Type 1

Random Demand Type 1 is given by $P(q,s) = a - bq + s$, and is defined by eq. (3.10). Assume $s \sim \mathcal{U}(-0.5, 0.5)$ such that $E(s) = 0$. The expected demand in this case is thus:

$$
\begin{aligned}
E[P(q)] &= \int_{\underline{s}}^{\overline{s}} [P(q,s)]\, f(s)\, \mathrm{d}s \\
&= \int_{\underline{s}}^{\overline{s}} [a - bq + s]\, f(s)\, \mathrm{d}s \\
&= a - bq + \int_{\overline{s}}^{3} s f(s)\, \mathrm{d}s \\
&= a - bq + E(s) \\
&= P(q) \\
\text{for } E(s) &= 0
\end{aligned}
$$

Assume $a, b > 0$ and given numeric values, also assume q and s is uncorrelated, and apply definition 31.32 from Sydsæter et al. (2002, p. 182).

For any given level of production, quantity can be given as a function of price since the functions are invertible. Let $\tilde{P}$ denote the price corresponding to quantity $\tilde{q}$, and let s denote uncertainty. Then the following apply:

$$\tilde{P} = a - b\tilde{q} + s \Leftrightarrow \tilde{q} = \frac{1}{b}(a - \tilde{P} + s)$$

To find marginal change in quantity subject to a change in uncertainty, differentiate the inverse expression for quantity with respect to uncertainty (s). Similarly, the marginal change in price subject to a change in uncertainty can also be derived from the above.

$$\tilde{q}_s = \frac{\partial \tilde{q}}{\partial s} = \begin{cases} \frac{1}{b} & \text{if } s > 0 \\ 0 & \text{if } s = 0 \\ -\frac{1}{b} & \text{if } s < 0 \end{cases}$$

$$\tilde{P}_s = \frac{\partial \tilde{P}}{\partial s} = \begin{cases} 1 & \text{if } s > 0 \\ 0 & \text{if } s = 0 \\ -1 & \text{if } s < 0 \end{cases}$$

Use the cost function as given in eq. (3.13) and let $q = \tilde{q}$ from the inverse relation to find the cost expressed in terms of uncertainty, and $MC(q, s)$ and derive MC_s.

$$\begin{aligned} C(q, s) &= A\tilde{q} + Bq^2 \\ MC(q, s) &= \frac{\partial C}{\partial q} = A + 2Bq \\ MC_s &= \frac{\partial}{\partial s} MC(q, s) = \frac{2B}{b} \gtreqless 0 \text{ if } s \gtreqless 0 \\ MC_{ss} &= \frac{\partial}{\partial s} MC_s = 0 \end{aligned}$$

Revenue is $= P(q, s)q = (a - bq + s)\, q$. Let $q = \tilde{q} = \frac{1}{b}\left(a - \tilde{P} + s\right)$. Then marginal revenue and change in marginal revenue subject to a change in uncertainty is given by $MR(q, s)$ and MR_s respectively:

$$\begin{aligned} MR(q, s) &= P(q, s) + \frac{\partial P(q, s)}{\partial q} q \\ &= a - 2b\left[\frac{1}{b}\left(a - \tilde{P} + s\right)\right] + s \\ &= 2\tilde{p} - a - s \end{aligned}$$

$$MR_s = \frac{\partial}{\partial s} MR(q, s) = \begin{cases} -1 & \text{if } s > 0 \\ 0 & \text{if } s = 0 \\ 1 & \text{if } s < 0 \end{cases}$$

Expected marginal revenue for demand in eq. (3.10), in Figure 6.1a denoted EMR[eq.(3.10)] is:

$$MR[\text{eq.(3.10)}] = a - 2bq + s = a - 2bq + 0$$

Marginal revenue for the proxy in eq. (5.3) is $MR[\text{eq.(5.3)}] = p^* - dp^*q$. In Figure 6.1a for average price p^* and slope dp^* obtained by *Algorithm 2* (Table 5.3, rows **T1 *Alg.2***).

Calculations - Random Demand Type 2

Random Demand Type 2 is given by $P(q, s) = a - (b + s)q$, and is defined by eq. (3.11). Assume $s \sim \mathcal{U}(0, 3)$ such that $E(s) > 0$. The expected demand in this case is thus:

$$\begin{aligned} E[P(q,s)] &= \int_{\underline{s}}^{\overline{s}} [P(q,s)]\, f(s)\, \mathrm{d}s \\ &= \int_{\underline{s}}^{\overline{s}} [a - (b+s)\, q]\, f(s)\, \mathrm{d}s \\ &= a - \left(b + \int_{\underline{s}}^{\overline{s}} s f(s)\, \mathrm{d}s \right) q \\ &= a - (b + E(s))\, q \\ &< P(q) \text{ for } E(s) > 0 \end{aligned}$$

Assume $a, b > 0$ and given numeric values, also assume q and s is uncorrelated, and apply definition 31.32 from Sydsæter et al. (2002, p. 182).

Let $\tilde{P}$ denote the price corresponding to quantity $\tilde{q}$, and let s denote uncertainty. Then the following apply:

$$\tilde{P} = a - (b+s)\tilde{q} \Leftrightarrow \tilde{q} = \frac{a - \tilde{P}}{b + s}$$

To find marginal change in quantity subject to a change in uncertainty, differentiate the inverse expression for quantity with respect to uncertainty (s). Similarly, the marginal change in price subject to a change in uncertainty can also be derived from the above.

$$\tilde{q}_s = \frac{\partial \tilde{q}}{\partial s} = \frac{\partial}{\partial s}\left(\frac{a - \tilde{P}}{s + b} \right) = -\frac{a - \tilde{P}}{(s+b)^2} \leqslant 0 \,\forall s \geqslant 0$$

$$\tilde{P}_s = \frac{\partial \tilde{P}}{\partial s} = -\tilde{q} \leqslant 0 \,\forall s, q \geqslant 0$$

Use the cost function in eq. (3.13) and let $q = \tilde{q}$ from the inverse relation to derive marginal cost $MC(q, s) = \tilde{C}_q$, and derive the second and third derivatives MC_s and MC_{ss}

$$\begin{aligned} C(q,s) &= A\tilde{q} + Bq^2 \\ MC(q,s) &= A + 2B\tilde{q} = A + 2B\left[\frac{a - \tilde{P}}{b+s}\right] \\ MC_s &= \frac{\partial}{\partial s} MC(q,s) = -2B\left[\frac{a - \tilde{P}}{(s+b)^2}\right] \quad < 0 \,\forall\, s \\ MC_{ss} &= \frac{\partial}{\partial s} MC_s = 4B\frac{a - \tilde{P}}{(s+b)^3} \quad > 0 \,\forall\, s \end{aligned}$$

Revenue is $= P(q,s)q = (a-(b+s)q)\,q$. Let $q = \tilde{q} = \frac{1}{b}\left(a - \tilde{P} + s\right)$. Then marginal revenue and change in marginal revenue subject to a change in uncertainty is given by $MR(q,s)$ and MR_s respectively:

$$\begin{aligned} MR(q,s) &= P(q,s) + \frac{\partial P(q,s)}{\partial q} q = a - 2(b+s)\tilde{q} \\ MR_s &= \frac{\partial}{\partial s} MR(q,s) = \frac{\partial \tilde{q}}{\partial s}\left[-2\tilde{q}\right] = 2\left[\frac{a-\tilde{P}}{(b+s)^2}\right] \; > 0 \,\forall\, s \\ MR_{ss} &= \frac{\partial \tilde{q}}{\partial s} MR_s = -4\left[\frac{a-\tilde{P}}{(b+s)^3}\right] \; < 0 \,\forall\, s \end{aligned}$$

Similarly the invers relation $\tilde{q} = \frac{a-\tilde{P}}{b+s}$ is used to derive Π_q, Π_{qs} and Π_{qss}:

$$\begin{aligned} \Pi(q,s,K) &= P(q,s)q - C(q,s) - \Psi(K) \\ \Pi_q &= \frac{\partial \Pi(q,s,K)}{\partial q} = \left(a - 2\left[b+s\right]\tilde{q}\right) - (A + 2B\tilde{q}) \\ \Pi_{qs} &= \frac{\partial}{\partial s}\Pi_q = \frac{\partial \tilde{q}}{\partial s}\left[-2B\tilde{q}\right] = 2\left[\frac{a-\tilde{P}}{(b+s)^2}\right] > 0 \,\forall\, s \end{aligned}$$

Expected marginal revenue for demand in eq. (3.11), in Figure 6.1b denoted EMR[eq.(3.11),s=1.5] is:

$$MR[\text{eq.(3.11)}] = a - 2(b+s)q = a - 2(b+1.5)q$$

Marginal revenue for the proxy in eq. (5.3) is $MR[\text{eq.(5.3)}] = p^* - dp^*q$. In Figure 6.1b for average price p^* and slope dp^* obtained by *Algorithm 2* (Table 5.3, rows **T2 *Alg.2***).

Calculations - Random Demand Type 3

Random Demand Type 3 is given by $P(q,s) = (a-bq)s$, and is defined by eq. (3.12). Assume $s \sim \mathcal{U}(0,2)$ such that $E(s) = 1$. The expected demand in this case is thus:

$$\begin{aligned} E[P(q,s)] &= \int_{\underline{s}}^{\overline{s}} \left[P(q,s)\right] f(s)\,\mathrm{d}s \\ &= \int_{\underline{s}}^{\overline{s}} \left[a - bq\right] sqf(s)\,\mathrm{d}s \\ &= \int_{\underline{s}}^{\overline{s}} \left[as\right] f(s)\,\mathrm{d}s - \int_{\underline{s}}^{\overline{s}} \left[bsq\right] f(s)\,\mathrm{d}s \\ &= aE(s) - bE(s)q \\ &= P(q) \\ \text{for } E(s) &= 1 \end{aligned}$$

For any given level of production, quantity can be given as a function of price since the functions are invertible. Let $\tilde{P}$ denote the price corresponding to quantity $\tilde{q}$, and let s denote uncertainty. Then the following apply:

$$\tilde{P} = (a - b\tilde{q})s \ \Leftrightarrow \ \tilde{q} = \frac{1}{b}\left(a - \tilde{P}\frac{1}{s}\right)$$

To find marginal change in quantity subject to a change in uncertainty, differentiate the inverse expression for quantity with respect to uncertainty (s). Similarly, the marginal change in price subject to a change in uncertainty can also be derived from the above.

$$\tilde{q}_s = \frac{\partial \tilde{q}}{\partial s} = \frac{1}{b}\tilde{P}\frac{1}{s^2} \geqslant 0 \ \forall s \geqslant 0$$

$$\tilde{P}_s = \frac{\partial \tilde{P}}{\partial s} = (a - b\tilde{q}) \geqslant 0 \ \forall \tilde{q} \geqslant 0$$

Use the cost function in eq. (3.13) and let $q = \tilde{q}$ from the inverse relation to derive marginal cost $MC(q,s) = \tilde{C}_q$, and derive the second and third derivatives MC_s and MC_{ss}

$$\begin{aligned}
C(q,s) &= A\tilde{q} + Bq^2 \\
MC(q,s) &= A + 2B\tilde{q} = A + 2B\left[\frac{1}{b}\left(a - \tilde{P}\frac{1}{s}\right)\right] = A + \frac{2Ba}{b} - \frac{2B\tilde{P}}{b}\frac{1}{s} \\
MC_s &= \frac{\partial}{\partial s}MC(q,s) = \frac{2B\tilde{P}}{b}\frac{1}{s^2} \ > 0 \ \forall \ s \\
MC_{ss} &= \frac{\partial}{\partial s}MC_s = -\frac{4B\tilde{P}}{b}\frac{1}{s^3} \ < 0 \ \forall \ s
\end{aligned}$$

Revenue is $= P(q,s)q = [(a - bq)\,q]\,s$. Let $q = \frac{1}{b}\left(a - \tilde{P}\frac{1}{s}\right)$. Then marginal revenue and change in marginal revenue subject to a change in uncertainty is given by $MR(q,s)$ and MR_s respectively:

$$\begin{aligned}
MR(q,s) &= (a - 2b\tilde{q})\,s = \left(a - 2b\left[\frac{1}{b}\left(a - \tilde{P}\frac{1}{s}\right)\right]\right)s = 2p - as \\
MR_s &= \frac{\partial}{\partial s}MR(q,s) = -a \ < 0 \ \forall \ s \\
MR_{ss} &= \frac{\partial}{\partial s}MR_s = 0
\end{aligned}$$

Expected marginal revenue for demand in eq. (3.12) is EMR[eq.(3.12)] (In Figure 6.1c):

$$EMR[\text{eq.(3.12)}] = (a - 2bq)s = (a - 2bq) \cdot 1$$

Marginal revenue for the proxy in eq. (5.3) is $MR[\text{eq.(5.3)}] = p^* - dp^*q$. In Figure 6.1c for average price p^* and slope dp^* obtained by *Algorithm 2* (Table 5.3, rows **T3 *Alg.2***).

Necessary & Sufficient Conditions:

Definition 4 *A function $\pi(K)$ is continuous on a closed, finite interval $[\underline{K}, \overline{K}]$ if it is continuous at each point of the interval, and the function $\pi(K)$ is a continuous function if $\pi(K)$ is continuous at every point of its domain. (Adams, 2003, p. 80)*

Definition 5 *The first order condition for K^* being a stationary point of the function $\pi(K)$ is defined by: $\pi'(K^*) = 0$. (Sydsæter et al., 2002, p. 87)*

Definition 6 *A function $\pi(K)$ is concave if $\pi\left(\alpha K_2 + (1-\alpha)K_1\right) \geqslant \alpha\pi\left(K_2\right) + (1-\alpha)\pi\left(K_1\right)$ for $K_1 < K_2$ and $\alpha \in (0,1)$. (Sydsæter et al., 2002, p. 80)*

Divergence of Proxy Demand

Proxy demand is given by eq. (5.3). The agent update his knowledge of both learning subjects p_t and dp_t according to eq. (4.2). All repeated below for reference.

$$\widetilde{P}(q) = p_t - \frac{dp_t}{2} q_t \qquad \text{Proxy Demand, eq. (5.3)}$$
$$p_{t+1} \leftarrow (1-\delta_t)p_t + \delta_t \hat{p}_t \qquad \text{Learning Price, eq. (4.2)}$$
$$dp_{t+1} \leftarrow (1-\delta_t)dp_t + \delta_t \hat{dp}_t \qquad \text{Learning Slope, eq. (4.2)}$$

Let $F(P, \theta) := [P(q), \frac{\partial}{\partial q}P(q)]$ be a function of the actual price and slope, and let the expectation function of price and slope, by assumption unknown to the agent, be defined by

$$f(p, dp) = EF(P, \theta) := [EP(q, \theta), \frac{\partial}{\partial q}EP(q, \theta)]$$

Assume the expected change in price for a change in quantity is such that $\frac{\partial}{\partial q}E\widetilde{P}(q) \leqslant 0$ and $\frac{\partial}{\partial dp}E\widetilde{P}(q) \leqslant 0$. Then the divergence of the expectation function $f(p, dp)$ is:

$$div f(p, dp) = \frac{\partial}{\partial q}EP(q, \theta) + \frac{\partial}{\partial dp}EP(q, \theta) < 0$$

List of References

Robert A. Adams. *Calculus: A Complete Course*. Pearson Addison Wesley, 5nd ed. edition, 2003. ISBN 0-201-79131-5.

Karl Aiginger. The impact of risk attitude, uncertainty and disequilibria on optimal production and inventory. *Theory and Decision*, 19(1):51–75, 1985. ISSN 0040-5833. doi: 10.1007/BF00134354. URL http://dx.doi.org/10.1007/BF00134354. (Accessed 08.08.13).

Karl Aiginger. *Production and decision theory under uncertainty*, pages 1–66, 171–178. Basil Blackwell Oxford, 1987. ISBN 0-631-14792-6. (Accessed 03.04.12) Available as Aiginger's homepage: http://karl.aiginger.wifo.ac.at[1].

AMPL. Student version for windows. URL http://www.ampl.com/DOWNLOADS/. (Downloaded 09.09.13).

BibTEX. -Bibliography Management System. 2009. URL http://www.bibtex.org/. (Downloaded 09.05.12).

Léon Bottou. Stochastic learning. In Olivier Bousquet and Ulrike von Luxburg, editors, *Advanced Lectures on Machine Learning*, Lecture Notes in Artificial Intelligence, LNAI 3176, pages 146–168. Springer Verlag, Berlin, 2004. URL http://leon.bottou.org/papers/bottou-mlss-2004. (Accessed 21.05.13).

John G. Cross. A stochastic learning model of economic behavior. *The Quarterly Journal of Economics*, 87(2):p. 239–266, 1973. ISSN 00335533. doi: 10.2307/1882186. URL http://www.jstor.org/stable/1882186. (Accessed 16.05.13).

Ciaran Driver, Sonia Abubacker, and Georgios Argiris. Capacity choice under monopoly, flexible price and demand uncertainty. *Southern Economic Journal*, 63(2):pp. 526–532, 1996. ISSN 00384038. URL http://www.jstor.org/stable/1061186. (Accessed 12.04.13).

Michal Feckan. A generalization of bendixson's criterion. *Proceedings of the American Mathematical Society*, 129(11):pp. 3395–3399, 2001. ISSN 00029939. URL http://www.jstor.org/stable/2668886. (Accessed 11.10.13).

Sjur Didrik Flåm and Maria Sandmark. Learning to face stochastic demand. *International Game Theory Review*, 02(04):259–271, 2000. doi: 10.1142/S0219198900000238. (Accessed 12.05.12).

[1] Full path: karl.aiginger.wifo.ac.at/fileadmin/Publications/files_aiginger/Production_and_Decision.PDF

R. Fourer, David M. Gay, and Brian W. Kernighan. *AMPL:A modeling language for mathematical programming.* Pacific Grove, Calif. : Thomson/Brooks/Cole, 2nd ed. edition, 2003. ISBN 0534388094.

Arthur Hau. Theory of the firm facing uncertain demand revisited. *Economic Theory*, 24 (2):pp. 457–464, 2004. ISSN 09382259. URL http://www.jstor.org/stable/25055824. A comment to Leland (1972) and Korkie (1975). (Accessed 05.04.13).

Robert V. Hogg and Elliot A. Tanis. *Probability and Statistical Inference*. Pearson Prentice Hall, 7th ed. edition, 2005. ISBN 0131293826.

Trefor Jones. *Business Economics and Managerial Decision Making*. J. Wiley, 2004. ISBN 0-470-02112-8. E-book available via University of Bergen Library. (Accessed 03.08.12).

Thorsten Karrer. i10 Master/Diploma/Bachelor/PhD Thesis LaTeX Template. *Media Computing Group at RWTH Aachen University*, 2012. URL http://hci.rwth-aachen.de/karrer_thesistemplate. (Downloaded 28.09.12).

John Maynard Keynes. *The General Theory of Employment, Interest and Money*. London: Macmillan, 1936. ISBN 0-333-07223-5. (Accessed 03.10.13) Available as web edition from The University of Adelaid Library: http://ebooks.adelaide.edu.au/.

J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):pp. 462–466, 1952. ISSN 00034851. URL http://www.jstor.org/stable/2236690. (Accessed 05.10.13).

Robert M. Korkie. Theory of the firm facing uncertain demand: Comment. *The American Economic Review*, 65(1):pp. 245–247, 1975. ISSN 00028282. URL http://www.jstor.org/stable/1806418. Discussed by Hau (2004, p. 457) (Accessed 01.11.13).

H. W. Kuhn and A. W. Tucker. *Nonlinear programming*. University of California Press, Berkeley and Los Angeles, 5nd ed. edition, 1951. Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950.

LaTEX. A Documentation Preparation System. 2009. URL http://latex-project.org/. (Downloaded 09.09.13).

Hayne E. Leland. Theory of the firm facing uncertain demand. *The American Economic Review*, 62(3):pp. 278–291, 1972. ISSN 00028282. URL http://www.jstor.org/stable/1803376. (Accessed 02.04.13).

J. Mula, R. Poler, J.P. García-Sabater, and F.C. Lario. Models for production planning under uncertainty: A review. *International Journal of Production Economics*, 103(1):271 – 285, 2006. ISSN 0925-5273. doi: 10.1016/j.ijpe.2005.09.001. URL http://www.sciencedirect.com/science/article/pii/S0925527306000041.

Bruce A. Murtagh and Michal A. Saunders. MINOS 5.51 user's guide. 2003. URL http://www.stanford.edu/group/SOL/guides/minos551.pdf. (Accessed 21.08.2012).

NVE. Norwegian water resources and energy directorate. URL http://www.nve.no/realtimedata. Real time data from national measurement stations.(In Norwegian only.).

John Pallister and Jonathan Law. *A Dictionary of Business and Management*. Oxford University Press, 4th ed. edition, 2006. ISBN 9780192806482. Definition: Adaptive expectations hypothesis.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):pp. 400–407, 1951. ISSN 00034851. URL http://www.jstor.org/stable/2236626. (Accessed 02.11.10).

K. Sydsæter, A. Strøm, and P. Berck. *Matematisk Formelsamling for Økonomer*. Gyldendals Norsk Forlag AS, 3th ed. edition, 2002. ISBN 8200127974.

Lucent Technologies. SW – scrolling window program, 2000. URL http://www.netlib.org/ampl/student/mswin/readme.sw. Accessed 05.12.2012.

Lucent Technologies. ILOG AMPL CPLEX System: User's guide, Version 12.2, 2010. URL http://www.ampl.com/BOOKLETS/amplcplex122userguide.pdf. (Accessed 09.09.2013).

TeXnicCenter. LaTeX for Windows. 2013. URL http://www.texniccenter.org/. TXC 2.0 Beta 2. (Downloaded 09.09.13).

V.N. Vapnik. An overview of statistical learning theory. *Neural Networks, IEEE Transactions on*, 10(5):988–999, 1999. ISSN 1045-9227. doi: 10.1109/72.788640. (Accessed 11.09.13).