UiT
THE ARCTIC
UNIVERSITY
OF NORWAY

School of Business and Economics

Production Planning in Fisheries Under Uncertainty

Stochastic Optimization and Scenario Generation - A Case Study

—

Marius Runningen Larsson

Master thesis in Economics – June 2016

Abstract

Uncertainty in the value chain of fisheries exists and as a result, a production model taking uncertainty into account is important for such a business to be efficient. A way to better plan under uncertain conditions is through the use of stochastic programming. The literature for which this is applied to fisheries is limited. Constructing, testing, and evaluating such a model's applications for a fishery processing plant producing dried and salted, and fresh fish is the purpose of the thesis. This is done through a case study where the method of stochastic optimization is applied. Initially, scenarios representing the underlying distributions are generated through time series models for the variables and parameters exhibiting uncertain behaviour. These scenarios are used as input values for the mathematical program representing the value chain of the fishery. The results yielded in this thesis indicate that it is indeed an increase in efficiency to be had applying such a model, although the low value of the stochastic solution (VSS) estimates makes it difficult to conclude with certainty. Consequently, it is suggested to increase the complexity of the model to better represent the whole value chain in greater detail which is expected to increase the VSS. Furthermore should different scenario generating methods be evaluated for both harvest and price to compare the stability of the results as per now they are suspected to be to somewhat unstable as indicated by their dispersion and central tendency results.

Acknowledgements

I would like to thank my supervisor Professor Øystein Myrland and my co-supervisor Professor Terje Vassdal for their feedback, ideas, and thoughts on the subject presented in this thesis. Thank you to Capia AS for the initial talks and for being the initial intermediaries between me and the case study company. Thank you to Nergård AS and Geir Nilsen for supplying both data and great knowledge with regards to fisheries for this thesis.

I would further like to thank my parents, May-Britt and Lars Henry Larsson, for all their support. And Lilli Minh Nguyen for everything.

A shout-out to Jarl Fagerli for introducing me to programming.

Contents

List of Figures

List of Tables

List of Abbreviations

MA	**M**oving **A**verage
AR	**A**uto**r**egressive
ARIMA	**A**uto**r**egressive **I**ntegrated **M**oving **A**verage
SARIMA	**S**easonal **A**uto**r**egressive **I**ntegrated **M**oving **A**verage
LP	**L**inear **P**ogram
SLP	**S**tochastic **L**inear **P**rogram
AIC	**A**kaike **I**nformation **C**riterion
ADF	**A**ugmented **D**ickey-**F**uller
KPSS	**K**wiatkowski, **P**hillips, **S**chmidt, **S**hin
WAPM	**W**eak **A**xiom of **P**rofit **M**aximization
GARCH	**G**eneral **A**uto**r**egressive **C**onditional **H**eteroscedasticity
KG	Kilograms
SLUH	Gutted without head (**Sl**øyd **U**ten **H**ode)
PRNG	**P**seudo **R**andom **N**umber **G**enerator
RRD	**R**elative **R**ange **D**evation
RSD	**R**elative **S**tandard **D**eviation
VSS	**V**alue of **S**tochastic **S**olution
TAC	**T**otal **A**llowable **C**atch

List of Symbols

$\mathcal{N}$	Set of nodes, indexed by i
$\mathcal{S}$	Family of scenario sets, indexed by s
$\mathcal{T}$	Family of time period sets, indexed by t
T_t	Subset of time periods
S_s	Subset of scenarios
R	General revenue function
C	General cost function
$\mathbf{a}$	Product vector
$\mathbf{y}$	Time series vector
y_t	Value at time t of time series
B	Backshift operator
p	Order of the $AR(p)$-model
d	Order of the $I(d)$-model
q	Order of the $MA(q)$-model
n	Scenario fan node
Δ	Difference operator
γ	Test coefficient ADF
α	Intercept ADF
ν	Random error ADF
σ	Standard deviation
ϵ	White-noise or stochastic process $\sim N(0, \sigma_\epsilon^2)$
Φ	Seasonal AR-coefficient
ϕ	AR-coefficient
Θ	Seasonal MA-coefficient
θ	MA-coefficient
π_s	Probability of scenario s
τ	Tau statistics ADF
τ^c	Critical Tau statistics ADF
ξ	Vector of Stochastic Variables
λ	Shadow Price
$\Pi(\mathbf{x})$	Objective Function Underlying Deterministic Program
$\Pi^S(\mathbf{x})$	Objective Function Deterministic Equivalence Problem

Symbols for the mathematical programs are defined in chapter 5.

Chapter 1

Introduction

Norwegian exports of fish and seafood is an important source to both revenue and employment in Norway. Consequently, it is necessary for a domestic producer to be efficient in the production process to be able to compete in a globalized market with disadvantageous production costs. Cod, haddock, and similar types are disembarked and processed by autonomous fishing vessels and processing plants, and in addition handled by integrated fisheries. A typical integrated fishery manages the whole value chain from start to finish by having its own fleet of fishing vessels, processing the raw material when disembarked, and finally supplies and distributes it to the market. They are faced by uncertainty throughout this chain, and to mention some, both quality and the quantity of raw material is of importance in planning production. The operational analysis of fisheries has been developed and built upon by the help of mathematical optimization for some decades now, where the objective of the method is to increase efficiency in the production leading to reduced costs and greater revenue, ultimately increasing the profits. Although a deterministic method such as linear programming (LP) might help solve both smaller and larger problems given a determined situation, the lack of being able to account for random events such as price fluctuation and uncertainty of raw material procurement limits its accuracy, although it is a readily available and practical tool for optimization.

The absence of including uncertainty in a linear programming model can decrease its chance of yielding an accurate result (Kali and Wallace, 1994). The problem of not considering uncertainty, and thus risk, is the chance of overestimating profits and/or underestimating costs of the process. This is due to the naive certainty in the model. Consider the example where a production facility produces a mix of several commodities. One these products would indeed yield a higher gross profit margin per unit than the rest, albeit this might be marginal. A deterministic optimization program could overvalue the importance of this product as it only considers one possible outcome of its input variables, e.g. price and quality of raw materials. A stochastic approach i.e. an optimization model with uncertainty does a better job in evaluating several likely and/or unlikely scenarios than the deterministic. This method can assist the planning of production yielding more efficient operations and ultimately lead to higher profits.

While the history and different thoughts of profit maximization will not be touched upon in this thesis, some fundamentals surrounding this concept must be mentioned. In mainstream economics, no doubt the largest following in economics and closely related today to the neoclassical synthesis (Dequech, 2007), certain assumptions underlies the models and the economic thought. Some of these, e.g. perfect competition, intertemporal optimization, and rational participants in the economy (Blanchard, 1997), tells us that in equilibrium in the economy, profit maximizing activity happens only when the marginal cost is equal to the marginal revenue. This is due to the simple fact that if a company is able to extract a profit in any given market, another participant in the economy will acknowledge this and enter it. Consequently, the prices will be reduced due to the newly established competition. This will continue until equilibrium where no new agents will find an incentive to enter. The question whether this is true hinges on several of the assumptions. It's highly unlikely there exist such a market with full perfect competition. Even highly functional and transparent financial markets will have at least transaction costs. And while the knowledge of the (ir)rational consumer has been greatly increasing together with the increasing prevalence of experimental economics, especially the assumption with regards to perfect information is greatly explored in the literature. The prevalence of asymmetric and costly information does indeed affect firms behaviour making them more risk averse (Greenwald and Stiglitz, 1990, Stiglitz and Weiss, 1983). As a direct result of this and other market failures, resources can be spent to exploit these. Either through the use of new tools previously not available, or through reduced cost and availability of already existing ones. And even if one were to operate in a completely perfect market, as long as the weak axiom of profit maximization is not satisfied, there are improvements to be made. An answer to why this axiom is not met might be the fact that uncertainty in the variables is interfering with the decision making (Dasgupta, 2009). These concepts will be more formally defined in chapter 3.

With the above mentioned in mind, the need for a model taking uncertainty into account is indeed important for a business to plan its production in an efficient way and being competitive. The literature for which stochastic programming is applied to fisheries is limited with only a few examples and applications. Constructing, testing, and evaluating a stochastic optimization model's applications to fisheries is the purpose of this master's thesis. Consequently, a relatively generic mathematical program is applied to a case study company producing both dried and salted, and fresh fish. The difficulties they are faced with are uncertainty throughout their value chain both with regards to the harvest, the prices, and the quality of the fish. These uncertainties, with the exception for the quality for which there is little data, are modelled through the use of time series models. The results of these statistical models are simulated and returned as scenarios. The scenarios are in turn used as input values for the mathematical program and allows for evaluation and testing of the stochastic program. The outline of the thesis follows.

The thesis will start of in chapter 2 by describing the value chain of a fishery and some of it's logistical difficulties. Further, some of the research and previous works in the field of optimization in fisheries will be discussed. In chapter 3 the theoretical background of profit maximization and mathematical programming will be explained. Chapter 3 will also elaborate on the theoretical background for the time series scenario generating process and its foundations. Chapter 4 presents the method of optimization and scenario generation for the analysis done in this thesis, and the tools for which this is done. In chapter 5 the case study is presented and the results of applying the method to the case study is evaluated. chapter 6 contains discussion, conclusion, and suggestions for further work.

Chapter 2

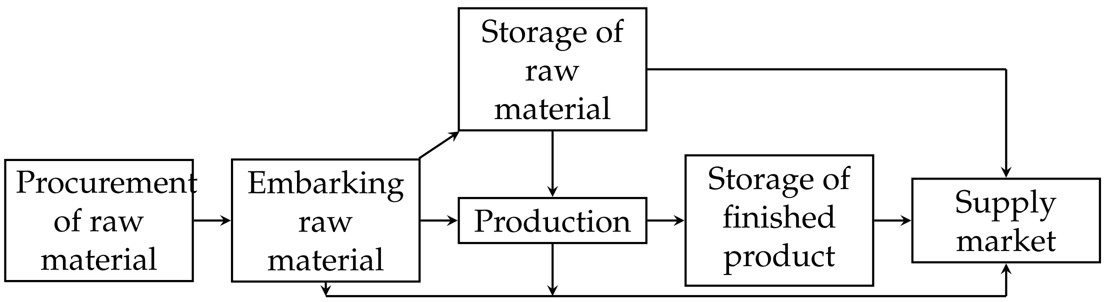
Fisheries and Optimization Research

In this chapter, the value chain of fisheries is presented, with variations. Furthermore a literature review follows which in no way is intended to be exhaustive, but should suffice as a solid enough background covering the evolution of mathematical optimization applied to fisheries since the early 1980's.

2.1 Value Chain Fisheries

The value chain of a fishery can be quite different depending on how the raw material is handled and what type of final products it is used for. In addition, different strategies can be applied by different fisheries. In Figure 2.1, a typical value chain is depicted, with variations.

FIGURE 2.1: Value Chain of an Integrated Fishery



First and foremost, the raw material is procured. For an integrated fishery, the fish is harvested by its own fleet of fishing vessels which can contain several different types depending on the scale of operations and type of fish. For typically larger integrated fisheries with larger production capabilities, trawlers are used. Trawlers applies the method of releasing a net dragging behind it either close to the sea bottom or at the surface, and is capable of being at sea for weeks on end due to their size. The size allows for greater storage and freezing capabilities of the raw material. Other types of methods applied are longline fishing, which uses the more traditional way of releasing a line with several hooks. While this type has been a traditional way of harvesting for years and is still used at a smaller scale, the method have since been automatized generating large autoline vessels capable of around the clock operations at open waters releasing tens of thousands of hooks a day. Seine fishing is also a common way of harvesting fish, especially herring and other types which operates in shoals. There exists several ways of seine fishing, while the main concept behind this method is to deploy a net surrounding the shoal, and when surrounded the harvesting is done through suction by using a pump pumping the fish from the ocean to the fishing vessel. Depending on the type of fish and the products sold by the fishery, much of the pre work can be done at the vessels already at sea. Often both the head and guts are removed making the fish SLUH, which is a Norwegian acronym for gutted without head. On larger vessels the filleting process can start.

The next step is to embark the raw material from the vessel to the landing facility. This can both be done at independent landing facilities, or facilities in conjunction with processing plants which is typical for integrated fisheries. The landing of fish is strictly regulated. Both in the quality, quantity, size, and other factors regulated by laws and regulations determining quotas, area of allowed harvest defined by the latitude, size of fishing vessels, *et cetera*, as stated by the *Directive for Harvesting Cod, Haddock and Saithe north of* $62°N$, the *Directive for Prohibiting Landing of Fish and Other Special Measures to Combat Illegal, Unreported and Unregulated Fishing Activities*, and the *Directive for Landings*. The trade in Northern Norway is maintained by The Norwegian Fishermen's Sales Organization. The organization makes sure fishermen gets paid fairly, and assists in controlling the sustainability of the industry. They operate a marketplace where fish is sold and bought. This is also true for the integrated fisheries. Quality assurance is also handled by the organization.

When embarked, the production process can start. Typically, the fish is sent through the process of filleting. This process can produce several types and classes of fillets from a single fish, ranging in the quality of the meat from the different parts of the fish. These products are sold either fresh or frozen. In addition, several different types of dried fish is produced from typically whitefish like cod and haddock, while pelagic fish such as saithe is also used. Depending on the method of drying the fish, different produce is made e.g. stockfish and clipfish. In addition, fisheries may supply finished SLUH or dried fish to other companies for re-branding or further processing. Consequently, the fish might be shipped directly to buyer from the embarking area after packaging without any further processing of the raw material.

There exists several different storage strategies for smoothing out the production of the fish. These can be deployed either to smooth out production for seasonal fluctuation in the harvest. In addition, this allows potentially to store both raw material and finished products in high supply seasons with low product price, and transfer the sales to low supply season with higher product prices. The latter method is true both for fresh, frozen, and dried fish, and is made possible for fresh fish due to more efficient ways of freezing and double freezing the fish.

The fisheries encounter uncertainty in several parts of their value chain, and this uncertainty makes it more difficult to be as efficient as possible. With regards to the amount i.e. the quantity of raw material, this is mostly determined by seasonal fluctuations and quotas, which determines a lot of the harvest for most of the different species procured. Despite this, quite high fluctuations can occur even during the busiest season due to the behaviour of the fish, spawn rates, and the potential difficulties subject to weather conditions and other random factors. Further, there exists uncertainty in the prices. Both the prices paid for the raw material when embarked, and in the prices for the final products. The former is controlled by The Norwegian Fishermen's Sales Organization, and requires a good knowledge of the current trends and regulations set by official entities to plan for. The latter is determined by the marked and can be considered a more traditional market price in which it is determined by the supply and demand of the respective products. The quality of the raw material is also a stochastic variable which must be planned for. Traditionally the fish with the highest quality is sold as fresh and frozen fillets, while the lower quality fish is processed as stock- or clipfish. The quality assurance and requirements are also regulated by The Norwegian Fishermen's Sales Organization for the regions in Northern Norway.

2.2 Litterature Review

There has been done quite a significant amount of research on operational planning and optimization, and while linear programs have been popular for several years, the stochastic approach have been ever increasing. One reason is the increased computer power which allows for more complex problems to be solved in a reasonable time, and where the cost of solving it does not exceed the benefit. Optimization such as different variants of LP has been applied to fisheries (Millar and Gunn, 1992; Randhawa and Bjarnason, 1995). Both integrated and stand-alone processing plants. As for application of linear programming as to do operational planning Mikalsen and Vassdal (1981) suggest a multi period LP-model where the objective is to increase the profitability of the fish manufacturing sector with the focus on storage management. This model is mainly constructed around a stand-alone producing plant which acquires raw material from the market. Consequently, an integrated fishing fleet and how this affects the product mix is not directly considered in this model. Shadow prices can subsequently be found to do a sensitivity analysis where the user of such a model, even though the model is deterministic, can derive decision rules *ex post* in contrast to the SLP where it is based on *ex ante* analysis. The model constructed in this thesis will be loosely based on the model of Mikalsen and Vassdal. A similar approach is done by Gunn et al. (1991). They suggest a multi period LP-model where maximization of net revenue is done subject to harvesting, production, and marketing. This model is constructed for an integrated fishery, and consequently the effect of its own fleet is of importance in optimizing the production. Furthermore, they touch on the fact that uncertainty is not taken into account in their model. They argue that this is already done by *corporate personnel* and a sensitivity analysis is sufficient. Further, they argue that a model such as theirs can be updated a long they way such as policies are updated in response to new information. As this is all true, it can be argued that this is just as true for a stochastic LP. A better question would perhaps be if the extra work of implementing uncertainty is in fact worth it. As far as their argument for not including uncertainty, this seems to have been the consensus for quite some time in fish processing optimization.

Begen and Puterman (2003) develops a LP-model for a salmon producer with the desired result to increase profitability, reduce decision making time, and over all streamline the production. While this model is considered for salmon and thus is not directly transferable to cod production, the overall idea is quite similar. Further, this model focuses on the allocation of harvest to their various processing plants. This is in contrast to the model being developed in this thesis, as this will only focus on the production value chain in one plant. They do however make a good point with regards to modelling uncertainty. They suggests an extension to their model making it a SLP-model where catch size is considered the random variable. Begen and Puterman were having a hard time modelling the uncertainty as the producer they made the model for failed to supply enough data. They further concluded that despite the few observations supplied, the data does in fact reflect the producers buying preferences. This makes it necessary to user other methods than time series models. With regards to the method of stochastic programming applied for fisheries, not much literature exists.

Bakhrankova et al. (2014) creates an integral stochastic programming model for optimization of operational production planning for fisheries. And as they state, research in supply chain management under uncertainty has indeed been done before (Dabbene et al., 2008; Schütz and Tomasgard, 2011), they point out that a stochastic modelling of optimization has yet to be applied to fish processing. The focus of their model is to determine whether a storage system based on super-chilled storage is beneficial. This is analysed through the use of uncertainty in prices and incoming raw material (quantity). Through the use of their SLP, they conclude that implementing super-chilled storage indeed may increase profits. They state that it indeed is necessary to account for uncertainty in determining whether this is true or not. As far as uncertainty in quality is concerned, they don't touch on this except stating that further research should be done by incorporating this as a stochastic variable as well. In contrast to the model being developed for the thesis, this focuses mainly on the effect of the super-chilling technology. This will not be done in this thesis. Furthermore, the scenario generation techniques in their paper is unknown and seem purely to be based on the assumption that after the second stage, each scenario considered is constant for the duration of the planning horizon. It's unclear why this is assumed as there's no elaboration with regards to the process of scenario generation besides the stated five different scenarios, which seem arbitrary. This might be due to the fact that while their case study is a real company, the original data might be withheld for privacy reasons and thus the scenarios implemented are just applied for illustration purposes of the model's functionality. The lack of focus on scenario generation in also seen in Simbolon et al. (2014) which presents a stochastic programming model for inventory management and meeting demand subject to uncertainty in quality. A chance constraint model is constructed to account for this uncertainty, while the other input factors are simulated through scenarios similar to Bakhrankova et al. (2014). In addition Naibaho and Mawengkang (2016) applies a nonlinear mixed integer stochastic programming model subject to environmental restrictions with the aim of increasing the efficiency and sustainability of a production process where a single processing plant distributes its final products to several different distribution centres.

Chapter 3

Theory

The prospect of profits is widely considered a fundamental incentive for establishing new business and to keep existing from closing down. Both mainstream economics and other heterodox schools of thought accept this, albeit under some different assumptions. While how one best can achieve profits is different depending on situation and sector, there exists some general fundamentals.

The idea of profit maximization has been a perpetual concept since the early days of economics, and certainly even before it was defined in mathematical terms. The concept is quite fundamental, and states that if one can increase ones profits by selling more, it's necessary to do so to be able to possess profit maximizing behaviour. This can be stated more formally by considering a general cost and revenue function. The cost function can be defined as $C(\mathbf{a})$ and represents the corresponding cost of producing the products contained in the product vector $\mathbf{a} = \{a_1, a_2, ..., a_i\}$. The revenue function $R(\mathbf{a})$ represents the revenue generated by selling the corresponding product in the product vector. As long as the revenue is greater than the cost, profits are increased. This can be showed more formally as stated by Varian (1992), and while he was most certainly not the first to state this relationship, two basic principles for maximizing profits follows such that $\max_{\mathbf{a}}(R(\mathbf{a}) - C(\mathbf{a}))$, and for the optimal solution vector $\mathbf{a}^*$, $\frac{\partial R(\mathbf{a}^*)}{\partial a_i} = \frac{\partial C(\mathbf{a}^*)}{\partial a_i} \forall i \in N$, which states that if marginal revenue exceeds the marginal cost, the activity should be increased to further increase profits. In equilibrium in a perfect competitive market, these conditions will always hold and consequently no further profits can be extracted. While it is quite unlikely every business goes through such a routine as constructing their own profit function, it is implicitly done by weighing revenue against cost. Consequently, the principle is the same. Furthermore, the Weak Axiom of Profit Maximization (WAPM) (Samuelson, 1948) must hold, by definition, for a business to be profit maximizing. The WAPM states that $\mathbf{p}^t\mathbf{y}^t \geq \mathbf{p}^t\mathbf{y}^{t'}\ \forall\, t$ and $t' \in T$, and $t \neq t'$, i.e. as long as there exists a profit maximizing production set for a given price today, no other output mix could generate a greater revenue than this. While measuring whether the WAPM holds or not is certainly not an easy task, especially due to measuring errors, the idea behind this axiom is still important and self explanatory: If there exists a better product mix than the one you already produce, why not adjust for it? As mentioned initially in the introduction, this might be because of uncertainty (Dasgupta, 2009). As a way of being able to plan production and sales better subject to uncertainty, a stochastic optimization approach can be applied. In this chapter, the theoretical background of stochastic optimization will be explained, both in terms of the mathematical program, and the process of scenario generation.

3.1 Mathematical Programming and optimization

As discussed in chapter 2, mathematical programming and operational planning is indeed a well established and well tested method of doing optimization in fisheries and other closely related production processes. The method of mathematical programming allows for a relatively good and not too inaccurate way of evaluating the operational processes and its efficiency. And while mathematical programming will give good approximations for complex processes, the decision whether to apply a linear or a non-linear method is a decision where costs must be weighted against the benefits, and consequently the question is whether or not the more close to reality non-linear program is indeed suited better for the problem at hand rather than the linear one. In addition, benefits of including dynamic variables must be evaluated. There exists several reasons for why such a program can perform better in certain circumstances than a static one. The two-stage one period suffers under the fact that it is incapable of treating different production processes with different time horizons correctly (Kali and Wallace, 1994). This is especially true in operations research on fisheries. Further the dynamic property is necessary to be able to model storage from one period to the next and production over a certain time horizon (Bakhrankova et al., 2014, Mikalsen and Vassdal, 1981).

3.1.1 Linear Programming

When constructing a stochastic programming model, one usually starts with an underlying deterministic linear program (Griva et al., 2009)

$$\begin{gathered} \min\, c^T x \\ s.t.\, Ax \leq b, x \geq 0 \end{gathered} \tag{3.1}$$

where $c^T x$ is the objective function, x is the vector of variables which is to be determined. c and b are known vectors of coefficients e.g. cost of producing product x and capacity restrictions, respectively. A LP-model can either be in a general or canonical form such that for the general form, the constraint can be greater or equal, less or equal, or equal to. In canonical form, only less or equal is permitted. While the differences are important, simple mathematical steps can be made to make them interchangeable. To be solvable, both the requirements for feasibility and boundedness must be satisfied. Consequently, the constraints must define a bounded convex polyhedron i.e. a convex set in n-dimensions. For simple models, observation is enough to find the complete vector x for which the problem is minimized under a convex curve. When increasing in complexity and moving the problem from the eucledian plane to the space for n-dimensions, different algorithms can be applied to solve the problems. The Simplex algorithm is a search algorithm allowing for solving LP-models given feasibility and boundedness. The method is divided into two step, and as stated by Dantzig (1998), the first step consists of, starting at a random extreme point, searching for a feasible solution. Should no such solution exist, the problem will be defined as infeasible and no solution will be returned. For a feasible solution, the next step is initiated starting at the optimal feasible point. The second search will determine whether the problem has a basic feasible solution, or if the problem is not downwards unbounded.

As discussed earlier, a LP-model can indeed be applied to complex problems. On the other hand, the lack of taking uncertainty into account reduces its accuracy. To expand on this, the LP can be expanded into a stochastic program by including uncertainty in the variables and/or the parameters.

3.1.2 Two-Stage Stochastic Program

An example of a non-deterministic model is a two-stage stochastic linear program with recourse and can be stated as (Kali and Wallace, 1994)

$$\begin{aligned} \min\ & c^T x + E[Q(x,\tilde{\xi})] \\ \text{s.t. } & Ax = b, x \geq 0 \end{aligned} \tag{3.2}$$

where

$$E[Q(x,\xi)] = \sum_j p^j Q(x,\xi^j) \tag{3.3}$$

and

$$Q(x,\xi) = \min\{q(\xi)^T y | W(\xi)y = h(\xi) - T(\xi)x, y \geq 0\} \tag{3.4}$$

Where $\xi \in \Xi$ on the probability space $(\Xi, \mathcal{F}, P)$, and the probability distribution P on $\mathcal{F}$ is given. Consequently, for every subset $A \subset \Xi$ that is an event, $A \in \mathcal{F}$, the probability $P(A)$ is known. The function $Q(x,\xi)$ is the recourse function, and $Q(x,\tilde{\xi})$ is the expected recourse function. The objective function Equation 3.2 is the first stage optimization, where the uncertainty is not yet realized and the first decision must be made. Next, the optimization of the recourse function Equation 3.4 is done and the proper adjustments are made. The overall goal is to decide on an optimal set *here and now* to optimize the two stages given the information we have. Thus, the recourse action is not done in practice, as will be shown in the explicit representation that follows in chapter 4.

Several variations of the stochastic programming method exists. The Chance-Constrained way, as first introduced in Charnes and Cooper (1959), considers situation where penalties for not abiding the constraints in the original program is difficult to define. The chance-constrained method allows for defining probabilities for such constraints holding. Consequently a program can be defined for which a feasible solution for a given probability will exists, making the program more flexible. Furthermore, the two-stage stochastic program can be extended to a multistage stochastic program. A multistage stochastic program expands on the two-stage such that more complex and more realistic optimizations can be done. The decision whether to apply a two-stage or multi-stage approach is consequently a decision where the cost versus the benefits must be considered as a multi-stage program far exceeds the need for computational capacity in comparison to the two-stage method.

For stochastic programs, Benders Decomposition (Benders, 1962) can be applied which is a method suggested for large linear programming models which exhibit block structures, i.e. partitioned matrices where the matrix can be subdivided into smaller matrices defined by the rows and columns in the original matrix (Anton and Rorres, 2011), which are typical for stochastic programs. The prevalence of this in stochastic programming is due to the fact that variables and parameters often are given as scenarios such that for a matrix of random variables and parameters, Ξ, this can be partitioned into vectors such that

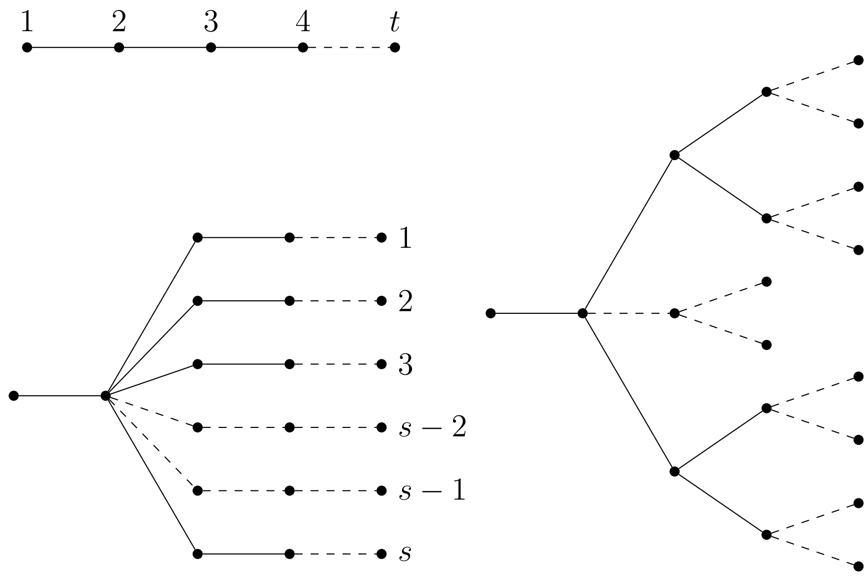
$$\Xi = \left[\begin{array}{cccc} \xi_{11} & \xi_{12} & \cdots & \xi_{1n} \\ \hline \xi_{21} & \xi_{22} & \cdots & \xi_{2n} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \xi_{m1} & \xi_{m2} & \cdots & \xi_{mn} \end{array}\right] = \begin{bmatrix} \boldsymbol{\xi_1} \\ \boldsymbol{\xi_2} \\ \vdots \\ \boldsymbol{\xi_m} \end{bmatrix}$$

Thus the Benders decomposition first solves for the main program without reducing it, and if encountering infeasible solutions, row generation is applied and the iterative search for a feasible solution is repeated until found or returned infeasible. This process is know as Benders Cut. While the method of Benders Decomposition should be explained in greater detail, it is beyond the scope of this thesis and it is noted that it is a widely applied method of solving stochastic programs with relative high efficiency.

3.2 Scenario Generation

Generating accurate scenarios is of great importance when constructing a stochastic program (Kali and Wallace, 1994; Di Domenica et al., 2009) Scenarios consists of probabilities of realisations on the underlying distribution of the random variable. This can be depicted as a scenario tree. Examples of scenario trees, which in the context of a two-stage optimization problem is referred to a scenario fan, can be seen in figure Figure 3.1 for s scenarios and t time periods. Such trees and fans consists of the set of nodes $n_i \in \mathcal{N}$ which represents events where a decision is made. The probability of each node is $p(n_t)$, where $\sum_{t \in T_i} p(n_t) = 1$, where $T_i \in \mathcal{T}$ and $\mathcal{T}$ is a family of sets over $\mathcal{N}$ which contains the sets for each period t. Furthermore, the trees consists of scenarios over two or multiple stages, where the first stage is over one time period and is known with certainty, while the next stages branches out. The general formulation of the probability of a scenario s is $P_s = \prod_{s \in S_j} p(n_s)$, where $S_j \in \mathcal{S}$ and $\mathcal{S}$ is another family of sets over $\mathcal{N}$ which contains the sets of scenarios s. Consequently, a scenario represents how likely, or unlikely, a vector of realizations on the variable evaluated are. Combining several such scenarios yields a scenario fan or a scenario tree where the combined probability over all scenarios is unity.

FIGURE 3.1: Visual Scenario Representation



Top left: Deterministic behaviour for t periods. Bottom left: Scenario fan for two-stage stochastic program for s scenarios. Right: Multistage scenario tree

To generate a suitable scenario tree Di Domenica et al. (2009) runs through several different methods in doing so. To mention some, they suggest several econometric and time series models including autoregressive models (AR(p), ARIMA(p,d,q)) and vector autoregressive models (VAR). Other statistical approaches are moment matching (Høyland et al., 2003) and discretization (non-parametric methods), and discrete sampling (random sampling). Different forms of *Monte Carlo* simulations and bootstrapping are used to generate scenarios. While the theoretical background of several of them should be mentioned in greater detail, the shear amount of suitable methods makes this unreasonable. Consequently only the theoretical background for the time series approach is presented, which will be further expanded upon in chapter 4.

3.2.1 Econometric and Time Series Analysis

Time series methods and econometric analysis is a widely applied technique for generating scenarios, especially in modelling supply chains, electricity price and demand, hydro and wind power, and in financial markets. With regards to examples in these fields of research and application there exists too many to discuss even a fraction, but to mention some, Zhou et al. (2009) applies the Autoregressive Moving Average (ARMA) model to simulate price scenarios based on historical data through two stages. First the demand is determined and is subsequently fitted to polynomials between the demand and price. Through this method they conclude that even on their limited time series they are able to generate realistic price scenarios. In Sharma et al. (2013) an algorithm is presented to generate and reduce scenarios through the use of the ARMA-model and the probability distance based scenario reduction method, respectively. It is suggested that the use of the ARMA in their algorithm successfully can be used for different types of planning and operations. Why is the ARMA and similar econometric models such as the GARCH model so widely applied in these types of analysis? The answer lies in the combination of the stochastic properties of the data combined with the repeating patterns e.g. time of day where electricity demand is higher, and other seasonal effects, such that next periods unrealized value is dependant on the value of previous periods and today.

As a closer look into the time series approach (Hill et al., 2008; Shumway and Stoffer, 2010; Hull, 2006), consider first a k-period time series which can be denoted as

$$\mathbf{y}_k = \{y_{t1}, y_{t2}, ..., y_{tk}\}$$

consequently a time series is a collection of discrete observed values for a given data measure object. For statistical analysis of time series, the variables should exhibit stationary behaviour such that the results doesn't suffer from spurious regression, which more often than not will give significant results when there are none. For the time series to be stationary both the mean and the variance must be identical regardless of the time period they are observed. More formally, it can be stated that the probabilistic behaviour of the observed variables, $\mathbf{y}_k$, is identical to that of the time shifted set $\mathbf{y}_{k+h}$ such that $P\{\mathbf{y}_k \leq \mathbf{c}_k\} = P\{\mathbf{y}_{k+h} \leq \mathbf{c}_{k+h}\} \forall k, h \in \mathbb{Z}^+$ which implies that the mean must be constant over all time periods. To investigate whether or not the time series is stationary often a visual test on the plot will suffice. And while this might be true for the most obvious examples, there exists several tests which can be applied to determine to a more certain degree the stationary, or the non-stationarity, in the series. To mention some, both the Augmented Dicky-Fuller and the KPSS test can be done.

Unit Root Tests

The Augmented Dicky-Fuller (Dickey and Fuller, 1979) is a unit root test where, $\Delta y_t = \alpha + \gamma y_{t-1} + \sum_{s=1}^{m} a_s \Delta y_{t-s} + v_t$ for $H_0 : \gamma = 0$, and $H_1 : \gamma < 0$. Reject the null if $\tau \leq \tau^c$. If H_0 is not rejected, the time series exhibits non-stationary properties and the time series y_t is integrated of order one $I(1)$.

As a supplement to the unit root test, the KPSS (Kwiatkowski et al., 1992) test was developed and can be used as a complementary test to the DF to check for stationarity in the time series. Its intended use is to assist the tester where other unit roots tests fail to give sufficient information. This stationary test tests for the null hypothesis that the time series are integrated of order zero $I(0)$ i.e. is stationary. Consequently, for a decomposition of the series into a deterministic trend, a random walk, and a stationary error (in the authors notation), $y_t = \xi_t + r_t + \epsilon_t$, respectively, where for the random walk $r_t = r_{t-1} + u_t$ and $u_t \sim iid(0, \sigma_u^2)$. Thus for the situation where the variance is zero $\sigma_u^2 = 0$ the random walk can be stated as $r_t = r_{t-1} \forall t$ and the time series y_t can be concluded to be stationary. Thus the null and alternative hypothesis can be stated as $H_0 : \sigma_u^2 = 0$ and $H_1 : \sigma_u^2 > 0$, respectively.

Stochastic Processes

For unobserved variables of a time series, i.e. variables which has not yet been realized, and unless perfectly predictable, are random. This process is called a stochastic process such that observed values of the time series are the realized stochastic process. This is a common feature in economic time series. While the time series observed in economics are discrete, the values the stochastic variables can contain are usually continuous state space processes i.e. the values of the stochastic variable can be anywhere on the real number line such that for a random variable, $y_1 \in \mathbb{R}$. This allows for the application of models such autoregressive and moving average models..

Autoregressive and Moving Average Models

More often than not, and despite the stochastic properties of the unrealized variables, there exists patterns in the time series. Some of these patterns can be evaluated through the use of autoregressive and moving average models. For an autoregressive model, $AR(p)$, the current value of the series, y_t, depend on the previous ones such that $y_t = \sum_{j+1}^{p} \phi_j y_{t-j} + \epsilon_t$ for a p amount of lagged variables for a stationary series, $\mathbf{y}_k$. For a moving average model, $MA(q)$, the current realization of the variable, y_t, depends on the previous values of the white noise such that $y_t = \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \epsilon_t$ for a stationary series. Both of these effects can occur in time series, and consequently the $ARMA(p, q)$ model can be applied which combines the two effects for stationary series. For a non-stationary series, which is quite typical for economic time series, a n-th order differenced version of the model is used, namely the $ARIMA(p, d, q)$-model, which will further explored in subsection 4.2.1. Due to the nature of the AR and MA processes, and their combined versions, they are able to give information about upcoming periods based on information given in the past. This is often referred to as forecasting, and is an essential part of being able to generating scenarios through the use of these models. That is, e.g., given an $AR(p)$ model $y_t = \sum_{j+1}^{p} \phi_j y_{t-j} + \epsilon_t$, the next period value of the time series, y_{t+1}, can be estimated by evaluating the observed time series and its values for y_t and e_t. These features will be exploited to generate scenarios and will further be expanded upon in section 5.2

3.3 Scenario Reduction

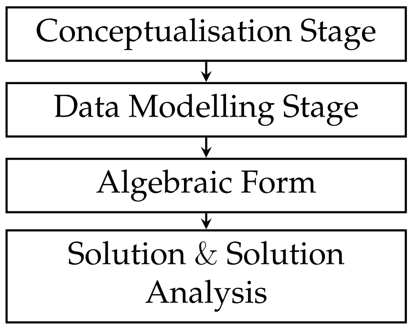
There exist several different ways of scenario reduction. For stochastic programs, the shear number of scenarios necessary to yield as good results as possible might be overwhelming even for powerful computers. To be able to account for this, i.e. the cost of operating complex mathematical programs, scenario reduction can be applied (Römisch (2009), Heitsch and Römisch (2003)). This is allows to greatly reduce the number of scenarios needed to estimate good results, while making the scenarios as accurate and as close to the observed time series properties as possible.

Chapter 4

Method

Dominguez-Ballesteros et al. (1999) (as cited by Valente et al. (2001)) suggests the following modelling process of a mathematical programming problem as depicted in figure 4.1. The conceptualisation stage

FIGURE 4.1: Modelling process



consists of collecting and assessing real world information and develop a mathematical formulation of the problem. During the data modelling stage one extracts data of the random processes and generate scenarios for the random variables. An algebraic form is then formulated to make the problem readable for a computer solver. Finally, the model is processed and the results are produced and analysed. In this chapter a more specific description of the method for the research question at hand follows.

4.1 Conceptualisation Stage

The model which is to be applied will be loosely based on the short term production planning model developed by Mikalsen and Vassdal (1981). Their model is developed to assist decision makers in planing how to most efficient store raw materials as to postpone production, decide product mix, and take advantage of seasonal fluctuating prices. Their model is a deterministic linear program and is intended for a processing plant which acquires raw material in the market.

The fish processing value chain of an integrated fishery must be evaluated. Some of the questions which has to be answered are whether or not procurement of raw material solely is from their own fleet. Is the uncertainty in quantity and quality of fish a problem - What about fluctuation in prices? Are products only delivered on demand, or does there exist contractual agreements? The plants processing cod are faced by decisions whether to process and sell it fresh, frozen, or dried. This is based on raw material quality. Consequently, the product mix together with how to produce the clipfish optimally through different periods will be the main focus of the model. Both of these factors are greatly impacted by the expected intake of raw materials and the market price of the finished products. Thus both price, and the amount of procured raw material will be modelled as uncertain variables. The model will be constructed to be used on a single processing plant, such that if a business is running several the optimization model must be modified and repeated for each plant. This is not unreasonable as difference in both technology and location exists which affects production and storage and a model such as this can easily be modified to accommodate for such differences. It can also be advantageous as different processing plants in the enterprise can be compared. After reviewing these and several other factors, a mathematical model is constructed.

As mentioned in chapter 3, when constructing a stochastic programming model, the underlying deterministic program is first evaluated

$$\begin{aligned} \min\ & c^T x \\ s.t.\ & Ax \leq b, x \geq 0 \end{aligned} \tag{4.1}$$

which is defined in subsection 3.1.1. After construction this, and with the knowledge in mind of the assessment of the integrated fishery, the LP is expanded to a non-deterministic model.

To expand on the idea of the two-stage modelling, a scenario formulation of the program can be a more practical way in solving an optimization problem. As suggested by Higle (2005) the explicit representation, or the deterministic equivalent problem (DTE), can be stated as

$$\begin{aligned} \min \sum_{\xi \in \Xi} & p_\xi (c x_\xi + g_\xi y_\xi) \\ \text{s.t.}\ T_\xi x_\xi + W_\xi y_\xi & \geq r_\xi \\ x_\xi - x = 0\ & \forall \xi \in \Xi \\ x_\xi, y_\xi & \geq 0. \end{aligned}$$

which now contains the non-anticipativity constraints $x_\xi = x$ such that in contrast to the recourse formulation, this allows for the program to omit the recourse action all together. Consequently the second stage function is maximized instantaneously. This is due to the fact that for each $\xi \in \Xi$, $p_\xi = P\{\tilde{\xi} = \xi\}$, i.e. for the expected value in Equation 3.2, the objective in the above mentioned DTE represents the same expected value. With this formulation in mind the optimization problem for an integrated fishery producing fresh cod and stockfish is presented in chapter 5.

4.2 Data Modelling Stage

After observing the random variables during the conceptualisation stage, data must be extracted to generate scenarios, and as mentioned in chapter chapter 3, there exists several methods of generating scenarios for a stochastic optimization problem, and preferable several should be applied and compared to evaluate which yields the most accurate and best result. In this thesis two different, but in a sense somewhat similar, *Monte Carlo* approaches has been applied through the use of ARIMA-modelling (Box et al., 2015; Whitle, 1951) and its extensions. As the data set for price is quite small and inhibits a more stochastic behaviour, regular ARIMA-*Monte Carlo* simulations on the white noise is done. As for the intake of the raw material, this is much more determined by seasonal fluctuations and consequently conditional simulations is done by the use of a seasonal ARIMA model (SARIMA). Both models are explained in more detail below, while the explicit way of generating the scenarios are presented in chapter 5

4.2.1 ARIMA

The ARIMA(p,d,q) model can in backshift operator notation be expressed as

$$\phi(B)\nabla^d y_t = \theta(B)\epsilon_t + \mu \tag{4.2}$$

Which expanded can be stated as

$$\Delta y_t = \sum_{j+1}^{p} \phi_j y_{t-j} + \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} \tag{4.3}$$

for $\Delta y_t = y_t - y_{t-1}$ and the error term, ϵ_t, i.e. the stochastic process or the white-noise which follows $cor(y, \epsilon) = 0$ and $\epsilon_t \sim N(0, \sigma_\epsilon^2)$. Consequently the ARIMA(p,d,q) model is a model combining both autoregressive and moving average parameters, for an amount p and q, respectively, such that for $p, \{\phi_j\}_{j=1}^{p}$ and for $q, \{\theta_j\}_{j=1}^{q}$, at a d-th order of difference.

The method applied for estimating the $ARIMA(p, d, q)$ model is the standard method supplied by `forecast::arima` in R, which is a log likelihood method that applies Kalman filtering (Gardner et al., 1980), an algorithm used for measurements over time which goal is to increase accuracy of the estimated coefficients and their relationship. To evaluate the fit and accuracy of the returned model, the Akaike's Information Criterion (AIC) is used (Akaike, 1998). Thus the model with the lowest AIC value is the ARIMA(p,d,q) model for which a given configuration of p, d, and q minimizes the sum of the squared errors, or in this case the likelihood of the data. The AIC can be defined as $AIC = \ln(\frac{SSE}{T}) + \frac{2K}{T}$, or for the log likelihood estimation, $AIC = -2log(L) + 2(p + q + k + 1)$, and is increasing in an increase in parameters added to the model for which the SSE, or the likelihood, is not reduced. Consequently the model with the lowest AIC is kept while the others are rejected.

4.2.2 Seasonal ARIMA

The seasonal ARIMA (SARIMA), $ARIMA(p, d, q) \times (P, D, Q)_s$, can in backshift operator notation be expressed as

$$\Phi(B^s)\phi(B)\nabla^d\nabla_s^D y_t = \Theta(B^s)\theta(B)\epsilon_t + \mu \tag{4.4}$$

Where in contrast to Equation 4.2, a seasonal difference of order D, $\nabla_s^D y_t = (1-B^s)^D y_t$, has been included such that for a seasonal effect s in the time series, the additional parameters Φ and Θ for the seasonal autoregressive and moving average effects are calculated. Further, the seasonal difference ∇_s^D is estimated. The seasonal ARIMA is calculated in the same fashion as the non-seasonal one.

4.2.3 Random Number Generator

A pseudo random number generator (PRNG) is used to generate the random errors for the *Monte Carlo* simulations. The algorithm implemented is the default generator in R, the Mersenne Twister (Matsumoto and Nishimura, 1998b). This algorithm inhibits the properties of high speed and efficient use of memory. As with most PRNGs, the method depends on its initial seed provided either by the user, or by the default value assigned to the seed. It has been shown that PRNGs of this type based on a linear recurrence, e.g. linear difference equation, can see some repetition when applied for parallel simulations that require independent RNGs (Matsumoto and Nishimura, 1998a). In addition, it can exhibit slow performance such that it may need several runs before generation random numbers passing different randomness tests (Saito and Matsumoto, 2008). While for this master's thesis the Mersenne Twister should suffice, there exists several other PRNGs which produce closer to true random behaviour and should be examined in further trials of scenario generation. As an example, Gülpınar et al. (2004) suggests the low-discrepancy Sobol sequences (Sobol', 1967) for simulation and optimization approaches to scenario tree generation.

4.2.4 Scenario Reduction

The scenario reduction algorithm implemented is a moment-matching algorithm (Zhou et al., 2009) which considers the first four moments where the four moments are defined as $\sigma = \sqrt{\frac{1}{S}\sum_{t\in S}(y_t - \mu)^2}$, $\mu = \frac{1}{S}\sum_{t\in S} y_t$, $skew = \frac{1}{S}\sum_{t\in S}\left(\frac{y_t-\mu}{\sigma}\right)^3$, and $kurt = \left(\frac{y_t-\mu}{\sigma}\right)^4$, for the standard deviation, mean, skewness, and kurtosis, respectively. These moments are first found for the historical data, which will be referred to as the *control*, and compared to the same moments for each generated scenario which were generated by the scenario generating algorithm. This moment matching method allows for a quick and simple way to reduce the numbers of scenarios greatly while keeping the accuracy of the distribution. However, to perform at the highest level, the true underlying distribution must be known. Because the true underlying distribution is unknown, valuable information can easily be lost by applying this method.

4.2.5 Software for Scenario Generation and Reduction

In this thesis, the scenario generation and reduction is applied using the object-oriented programming language R (R Core Team, 2015) which allows for the use of several different packages for easy and quick implementation of algorithms and functions. The `tseries` (Trapletti and Hornik, 2015) package gives the user tools allowing for easy data handling and manipulation of time series. In addition, tests for determining non-stationarity such as ADF and KPSS are available. The `Forecast` package (Hyndman, 2015, Hyndman and Khandakar, 2008) allows for calculations of the $ARIMA$-models and methods of automatic forecasting. A more detailed explanation of the implementation of the scenario generating and reduction algorithms follows in chapter 5.

4.3 Algebraic Form and Solution & Solution Analysis

Just a few decades ago one usually had to define an algebraic modelling language form of the problem and forward it to an institution which had a powerful enough machine to process the model. These days, even complex problems can be solved by home computers. Larger problems still benefit from increased computational power, especially when considering large multi dimensional scenario trees. There exists several different algebraic modelling languages which is constructed to solve mathematical optimization problems. Some of these are GAMS, AIMMS, AMPL, and Xpress-Mosel. The programming language R is also able to do mathematical optimization. In addition, MATLAB has the possibility of scenario generation and processing SLP-models. In this thesis, both the underlying deterministic and the stochastic program will be solved by the use of SAMPL, which is an adapted version of AMPL for stochastic programming containing the SAMPL engine FortSP (Valente et al., 2001). The solver system FortSP allows for quick and simple ways in solving both deterministic and stochastic models through the use of solvers such as CPLEX which for larger LP problems applies, among other, the Simplex algorithm. In addition, the solver FortMP (Neumaier and Shcherbina, 2004) is available for LP problems. For stochastic programs Benders decomposition (Benders, 1962) is used.

Chapter 5

Analysis - The Case Study

The company used for the case study in this thesis is a larger integrated fishery located in Northern Norway. The company consists of their own fishing fleet, several production plants in Troms county, and their own sales department. For this case study, a single production plant is chosen where the fish species cod, saithe, and haddock are both embarked and processed. The final products which are to be evaluated in the mathematical program are fresh cod, and a type of cod which are both dried and salted, in Norwegian called *klippfisk*. For the remainder of this thesis, the product type of cod which are dried and salted will be refered to as *clipfish*, and is not to be mistaken with stockfish which is only dried and not salted. The case study company mainly delivers the whole fish, usually without head and guts (in Norway the status of this type of raw material and product is denoted SLUH, which is an acronym for "gutted without head" i.e. the entrails and heads are removed). The raw material is mainly embarked in the SLUH state as this is already done at sea on the fishing vessels. Consequently the workload at the processing plants is reduced. This means that the fish sold as fresh is directly sent from the boat, through packaging, to the market with no more processing. Thus the variable costs of producing the fresh is far less than the cost of producing the dried and salted fish. The case study company deliver the fish in boxes of 20 and 25 KG, for three different size categories. For this mathematical program, only the packaging of 25 KG is considered. Further, only cod as raw material is considered.

The clipfish is processed further through the more time consuming process of drying and salting. During this process, the fish is contained in salt for a total of 21 days to dehydrate it, and further dried for 2-4 days. This can vary hugely depending on the size of the raw material and how fast the fish matures due to the salt process. To further increase the quality of the process, the fish could be salted and dried for two months before shipping it. For the mathematical program at hand, the time is set to four weeks. During this process, the weight of the product is reduced by around 50%. The clipfish is shipped in both pallets of 1000 KG, and boxes of 25 KG. For the this mathematical program, only the packaging of the 25 KG is considered. Both cod, haddock, and saithe can be used as raw material to produce different types of clipfish. For this model, only cod

FIGURE 5.1: Location of Processing Plant

Map package in R by Becker et al. (2016)

is considered.

With regards to frozen products, the processing plant at hand does not supply much of this and is consequently omitted for this case study. While the mathematical program that follows is a simplified one, it can easily be adjusted for smaller time periods, different packaging types, different products i.e. frozen products, and for different species of fish. Further more, the program can easily be adjusted for different types of technology e.g. technology which could reduce the cost of drying and salting, or reducing the time it takes to go through such a process.

5.1 Mathematical Program

5.1.1 Deterministic Program

First, the underlying deterministic program is defined. The variables and coefficients are as follows

Sets

$\mathcal{T}$	Family of sets of time periods, indexed by T_i
	For T_1, production period, excluding incoming, indexed by t
	For T_2, production period with incoming, indexed by t
$\mathcal{K}$	Set of the different types of raw materials, indexed by k
$\mathcal{F}$	Family of sets of finished products, indexed by F_i
	For F_1, fresh products, indexed by f
	for F_2, dried and salted products, indexed by f

Decision variables

X_{tk}	Product k in time period t
Z_{tf}	Sales of product f in time period t

Parameters and known variables

Y_{tk}	Incoming raw material k at time period t

p_{tf}	Market price of product f at time period t
p_{tk}	Procurement price of raw material k at time t
v_k	Variable costs for type k
w_{kf}	Amount of raw material k needed for finished product f
j	Amount of time to produce clipfish in weeks.
NT_{T_i}	Amount of time periods in set T_i
D_{tf}	Exogenously defined demand and strategic variable.
C_{tf}	Production capacity.
a_{kf}	Loss of weight by drying and salting.

The objective equation, $\Pi(\mathbf{x})$, can be stated as

$$\max(\sum_{t\in T_1}\sum_{f\in\mathcal{F}} p_{tf}Z_{tf} - \sum_{t\in T_1}\sum_{f\in\mathcal{F}} v_f X_{tf} - \sum_{t\in T_1}\sum_{k\in\mathcal{K}} p_{tk}Y_{tk}) \tag{5.1}$$

To operate properly, the program must contain several restrictions, both to define upper and lower bounds for the decision variables, and conversion equations to direct the raw material to different use and storage for finished products.

First define the balance equation such that for incoming raw material Y_{tk} product X_{tk} is produced. As this is an integrated fishery, Y_{tk} is defined as a time-dynamic variable and is not decided by the model as the total amount of raw materials procured are given by the harvest of the fishing vessels at time t. The conversion equation can be stated as

$$\sum_{f\in F_1} w_{kf}X_{tkf} = Y_{tk},\ t\in T_2,\ k\in\mathcal{K} \tag{$\star$}$$

Consequently, the procured raw material is converted to the finished fresh product. The parameter w_{kf} defines the amount needed of raw materials in KG for one unit of finished product.

Next it's necessary to add to the balance equation the activity of producing the clipfish. The clipfish production consists of a a much more time consuming process in which the cod is salted for a longer period, then dried through the use of a dryer-system which objective is to remove the excess water from the product. This process can depend on technology used, and quality required and preferred. There is no discrimination between size of the fish in this model. The alternative to send the raw material to clipfish production is added to Equation $\star$ such that

$$\sum_{f\in F_2} a_{kf}w_{kf}X_{tkf} + \sum_{f\in F_1} w_{kf}X_{tkf} = Y_{tk},\ t\in T_2,\ k\in\mathcal{K} \tag{5.2}$$

At the stage of salting and drying, a significant amount of weight is lost due to the process of removing the water. In addition, the time required for the process must be included. This is defined by the coefficient a_{kf} and signifies how much raw material is needed for one KG of finished clipfish product. Further the flow to the sales department must be defined. For fresh cod, this goes instantaneously and can be stated as

$$Z_{tf} = X_{tf},\ t \in T_1,\ k \in \mathcal{K},\ f \in F_1 \tag{5.3}$$

Consequently, the final product X_{tk} is both the same as the one sold, Z_{tk}, and is sold as soon at it has gone through the production line. This is especially true for fresh fish which is delivered almost directly from boat to market. Given a storage system where fish is sent to storage as a frozen product either to keep for periods with higher price or to fill the demand for frozen products, this could be defined here.

For the dried and salted products i.e. the clipfish, the production process is significantly higher than the one for fresh, thus a balance equation which allows for the product to be made and sold in different time periods must be defined such that

$$Z_{(t+j)f} = X_{tf},\ t \in \{1,..,(NT_{T_2} - j)\},\ k \in \mathcal{K},\ f \in F_2 \tag{5.4}$$

Where j is defined as the amount of time needed for production of the dried and salted products. This balance equation allows for the incoming raw material i.e. from the previous year, be considered in the evaluation of product mix as well while not exceeding the current one year period.

Now that the connection between the input, Y_{tk}, production, X_{tk}, and output Z_{tk} is constructed, upper and lower bounds must to be defined. First and foremost, the non-negative constraints must be included to make the problem downwards bounded. This can be stated as

$$X_{tf}, Z_{tf} \geq 0 \tag{5.5}$$

While the model by now is both bounded and feasible and thus no more theoretical restrictions is required, there might exist several contractual or strategic reasons to include upper and lower bound for the decision variables. Consequently, the constraint for the production variables, i.e. Z_{tf} for $f \in F_1$ and $f \in F_2$, can be stated as

$$D_{tf}^{min} \leq Z_{tf} \leq D_{tf}^{max},\ f \in \mathcal{F},\ t \in T_1 \tag{5.6}$$

where the parameter D_{tf} represent an exogenously given contractual demand and/or strategical decisions and allows for the production of the specific products to be constrained. In addition, constraints for upper and lower bounds on production capacity can be stated as

$$C_{tf}^{min} \leq X_{tf} \leq C_{tf}^{max},\ f \in \mathcal{F},\ t \in T_1 \tag{$\star\star$}$$

By this, and since $X_{tf} = Z_{tf}\, \forall f \in F_1$, it follows that for the program to be feasible it's necessary for $C_{tf}^{min} \ngeq D_{tf}^{max}$ and/or $D_{tf}^{min} \ngeq C_{tf}^{max}$ as this program does not allow for additional storage possibilities.

Thus for simplicity the exogenously given demand and production capabilities is contained in the same parameter such that $D_{tf} = C_{tf}$ in this program and Equation $\star\star$ is omitted.

To summarize, the full underlying deterministic program objective function $\Pi(\mathbf{x})$ can be stated as

$$\max(\sum_{t\in T_1}\sum_{f\in\mathcal{F}} p_{tf}Z_{tf} - \sum_{t\in T_1}\sum_{f\in\mathcal{F}} v_f X_{tf} - \sum_{t\in T_1}\sum_{k\in\mathcal{K}} p_{tk}Y_{tk}) \tag{4.2}$$

Subject to

$$\sum_{f\in F_2} a_{kf}w_{kf}X_{tkf} + \sum_{f\in F_1} w_{kf}X_{tkf} = Y_{tk},\ t\in T_2,\ k\in\mathcal{K} \tag{4.3}$$

$$Z_{tf} = X_{tf},\ t\in T_1,\ k\in\mathcal{K},\ f\in F_1 \tag{4.4}$$

$$Z_{(t+j)f} = X_{tf},\ t\in\{1,..,(NT_{T_2}-j)\},\ k\in\mathcal{K},\ f\in F_2 \tag{4.5}$$

$$D_{tf}^{min} \le Z_{tf} \le D_{tf}^{max},\ f\in\mathcal{F},\ t\in T_1 \tag{4.7}$$

$$X_{tf}, Z_{tf} \ge 0,\ f\in\mathcal{F},\ t\in T_2 \tag{4.6}$$

5.1.2 Stochastic Program

Some new variables, coefficients, and restrictions must be added to the underlying deterministic program such that the stochasticity can be included. This includes both making the decision variables X and Z be dependant on the different scenarios. Further, it's necessary to include a probability parameter π_s which allows for stating the probability of each scenario occurring. Furthermore, both the price and incoming raw material must be adjusted for the scenario set. For the two-stage explicit representation, i.e. the deterministic equivalent problem (DTE), these are

Sets

$\mathcal{S}$	Set of scenarios, indexed by s
Decision variables	
X_{tks}	Product k in time period t and scenario s
Z_{tfs}	Sales of product f in time period t and scenario s
Parameters and known variables	
Y_{tks}	Incoming raw material k at time period t and scenario s
p_{tfs}	Market price of product f at time period t and scenario s
π_s	Probability of scenario s occurring

And the non-anticipativity to ensure the two-stage formulation of the problem

$$Z_{1fs}, X_{1fs} = Z_{1f1}, X_{1fs} \tag{5.7}$$

Consequently, the complete DTE for the objective function $\Pi^S(\mathbf{x})$ can be stated as

$$\max \sum_{s\in S} \pi_s(\sum_{t\in T_1}\sum_{f\in\mathcal{F}} p_{tfs}Z_{tfs} - \sum_{t\in T_1}\sum_{f\in\mathcal{F}} v_f X_{tfs} - \sum_{t\in T_1}\sum_{k\in\mathcal{K}} p_{tk}Y_{tks}) \quad (5.8)$$

Subject to

$$\sum_{f\in F_2} a_{kf}w_{kf}X_{tkfs} + \sum_{f\in F_1} w_{kf}X_{tkfs} = Y_{tks},\ t\in T_2,\ k\in\mathcal{K},\ s\in\mathcal{S} \quad (5.9)$$

$$Z_{tfs} = X_{tfs},\ t\in T_1,\ k\in\mathcal{K},\ f\in F_1,\ s\in\mathcal{S} \quad (5.10)$$

$$Z_{(t+j)fs} = X_{tfs},\ t\in\{1,..,(NT_{T_2}-j)\},\ k\in\mathcal{K},\ f\in F_2,\ s\in\mathcal{S} \quad (5.11)$$

$$D_{tf}^{min} \le Z_{tfs} \le D_{tf}^{max},\ f\in\mathcal{F},\ t\in T_1,\ s\in\mathcal{S} \quad (5.12)$$

$$X_{tfs}, Z_{tfs} \ge 0,\ f\in\mathcal{F},\ t\in T_2,\ s\in\mathcal{S} \quad (5.13)$$

$$Z_{1fs}, X_{1fs} = Z_{1f1}, X_{1fs} \quad (4.8)$$

As a summary and to expand on the data used for the mathematical program, a quick run-through follows.

5.1.3 Parameters and Data

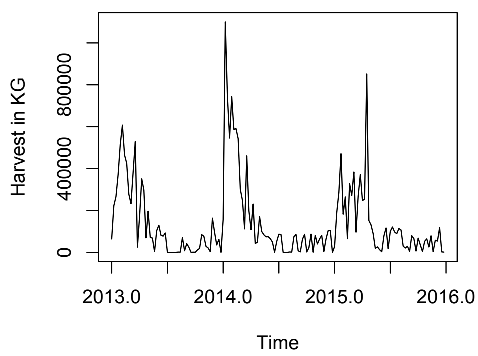
The numerical values for the parameters are set by cost estimates done by the case study company, with exception of the parameter for market price, p_{tfs}, and the variable for procured raw material, Y_{tks}, which will be estimated in section 5.2. The variable production cost estimates for the fresh and clipfish are $3.5\,NOK$ and $7.5\,NOK$, respectively, such that $v_1 = 3.5$ and $v_2 = 7.5$. The procurement price of the raw material, p_{tk}, is a dynamic deterministic variable in this model where the historical data is used as a proxy for the unrealized stochastic time series. The coefficient for depreciation of raw material weight through the salting and drying process, a_{kf}, is estimated to be 0.5 for the clipfish. The size of the products supplied the market, $w_k f$, are both set to 25 KG. The maximum capacity for producing clipfish, D_{t2}^{max}, is estimated to be 6000 KG per week, and the minimum requirement, D_{t2}^{min} is set to 200 KG, and for the fresh fish, D_{t1}^{min} is set to 500 KG. The model is maximized over the full year, i.e. $t = 52$, where the length of production of clipfish is set to 4 weeks, such that sales are available at the fifth week such that $j = 5$.

5.2 Scenario Generation

As mentioned in chapter 4, the applied methods for scenario generation is both the $ARIMA(p,d,q)$ and the seasonal version $ARIMA$ (p,d,q) $(P,D,Q)^s$ for price and quantity scenarios, respectively. The data used for the price scenarios can be seen to the right in Figure 5.2 and contains the mean price of the three different categories of fresh cold sold in packages of 25 kg products, i.e. 1-2 kg, 2-4 kg and 4-6, all in cases of 25 kg. While only a yearly time series of price doesn't do much in revealing seasonal fluctuations and other important information describing how the prices evolve, for this thesis only the price for one year is available. The time series should be expanded in the future to be able to model the price process more accurately. Despite of this, the price seems to be adjusting downwards for the periods where the supply is highest, which is in line with economic theory. None the less, it's difficult to conclude based on the short time series.

The data for the quantity scenarios can be seen to the left in Figure 5.2 and is the weekly SLUH quantity in KG embarked by the fleet. As this is the total quantity, the three size categories of cod has been added up and is assumed equal for the upcoming estimations. The historical data clearly shows a seasonal harvest, where the most activity is limited to the first 20 weeks of the year.

FIGURE 5.2: Historical Data for Harvest and Price of Cod



5.2.1 Price Scenarios

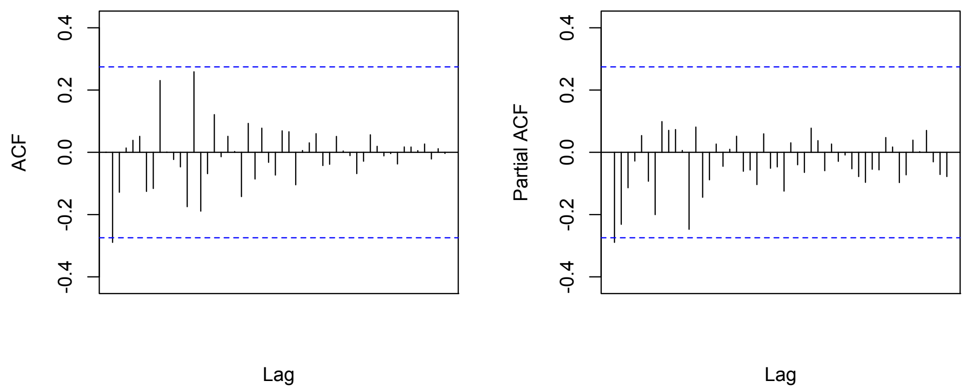
An $ARIMA(p,d,q)$ fit to the historical data can be made through the use of the `Forecast::auto.arima` algorithm in R which is an iterative process evaluating several different configurations of the ARIMA(p,d,q) model. The model with the lowest (AIC) is returned.

For the case study at hand, an ARIMA(0,1,1) is returned with coefficient $\theta = -0.4290$ and can be expressed as

$$\Delta y_t = \epsilon_t - 0.4290\epsilon_{t-1}$$

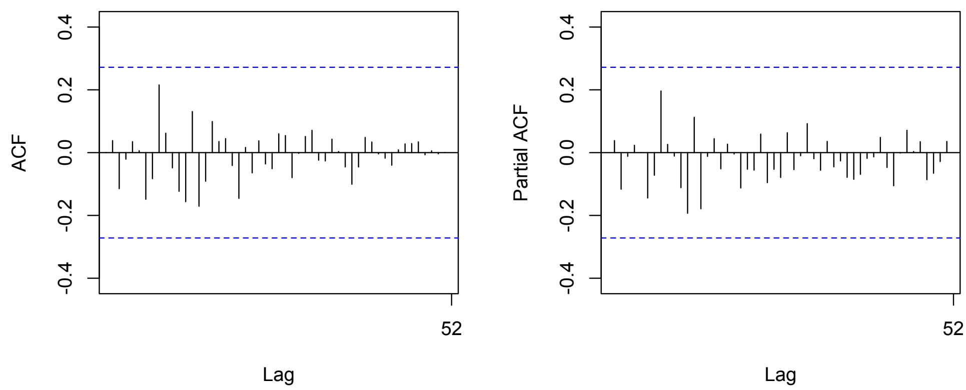
This result is further supported by the ACF and PACF plots as seen in Figure 5.3 which contains significant spikes at lag one which is suggestive of a $MA(1)$ model. The fitted model is seen in figure 5.5. The

FIGURE 5.3: ACF and PACF Plot for Price Time Series



mean of the residuals in the model is $\mu_\epsilon = 0.17$ and are close to a normal distribution with $\sigma = 3.64$. The correlation between the observed time series and the residuals is $cor(y, \epsilon) = 0.45$ and by inspection of the ACF the errors are not autocorrelated as seen in Figure 5.4. Based

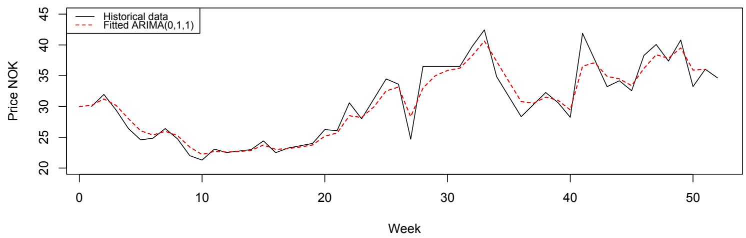
FIGURE 5.4: Residual ACF and PACF Plot for Price Time Series



on the low number of observations available it's difficult to conclude how close these estimates are to the true distribution. For the price simulations the errors will be assumed to contain the stochastic properties $cor(\epsilon_t, \epsilon_{t-1}) = 0$ and $\epsilon_t \sim N(0, \sigma_\epsilon^2)$.

The method applied for scenario generation is through the use of *Monte Carlo* simulations on the stochastic part of the model (Conejo

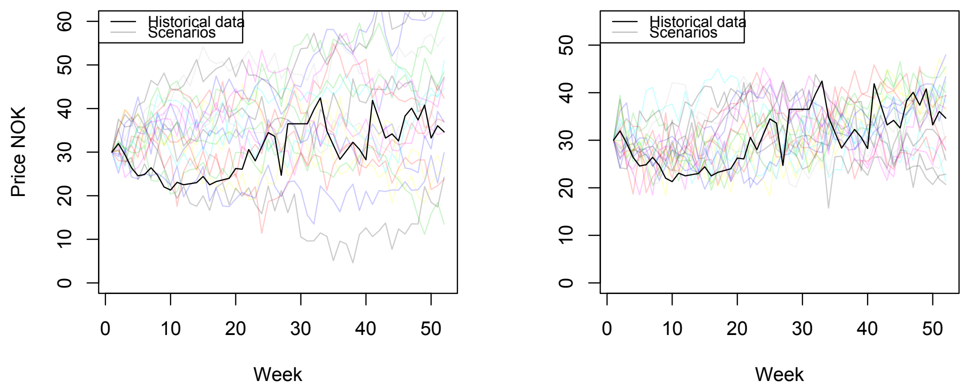
FIGURE 5.5: Fitted ARIMA(0,1,1) model to historical price data for cod 2015



et al., 2010), $\epsilon \sim N(0, \sigma_\epsilon^2)$, and the method can be illustrated through the algorithm seen in appendix B.1.1. The algorithm is implemented in R version 3.2.3. The code can be found in appendix A.1.1. An illustrative example can be seen to the left in Figure 5.6 for 20 generated scenarios.

The scenario reduction algorithm implemented is the moment matching (Zhou et al., 2009) algorithm presented in subsection 4.2.4 which considers the first four moments of the generated scenarios and compares them to the observed data (the *control*). Consequently the same moments for each generated scenario are calculated and matched with the *control*. The moment-matching scenario reduction algorithm is illustrated in appendix B.1.2. The algorithm is implemented in R and the code can be found in appendix A.1.2. For computational simplicity the last moment i.e. the kurtosis is omitted which may affect the results. An illustrative example can be seen in Figure 5.6 for 20 scenarios where out of initially 50 000 generated scenarios on average 90 moment-matched scenarios are returned.

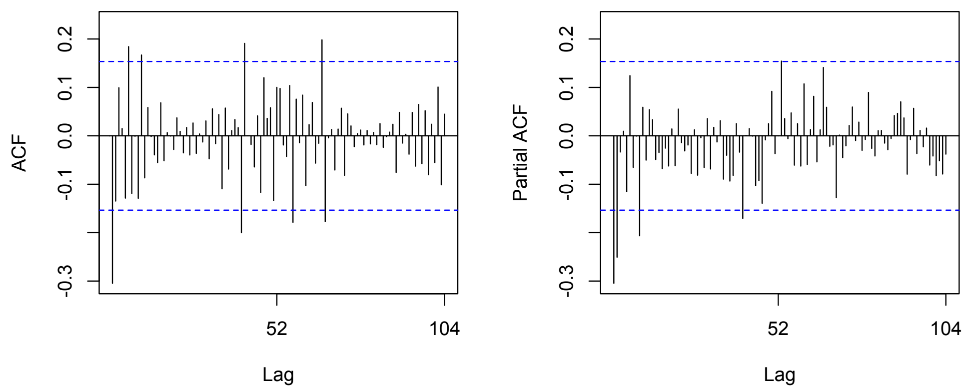
FIGURE 5.6: Raw generated price scenarios (left) and reduced price scenarios (right) for cod prices 2015



5.2.2 Quantity Scenarios

To construct conditional simulations Sak and Hörmann (2012) suggests an algorithm which they illustrate and implements in R as the function `arima.condsim`. Their approach is based on forecasting, while instead of basing the simulations on the mean-square prediction error, the recursive properties of the $ARIMA$ equation is used. Consequently a random realization on the time series, y_{t+1}, is created conditionally on the historical data with a random shock, $\epsilon_{t+1} \sim N(0, \sigma)$. This is repeated for each time period for as long as desired by the user where y_{t+2} is simulated conditionally on the observed historical time series and y_{t+1} and ϵ_{t+1}, *et cetera*. To use their method an ARIMA-model must be fitted. This is done in the same fashion as with the price scenarios in subsection 5.2.1. The returned best fit ARIMA with the lowest AIC is an $ARIMA(1, 0, 1) \times (0, 0, 1)^{52}$. When operating with an $ARIMA(p, d, q)$ where $p > 0$ and $q > 0$, its difficult in determining by inspection of the ACF and PACF plot what is the correct numbers for the respective processes, but the plots does indeed suggest a seasonal effect, as seen in Figure 5.7. Consequently, the returned model with the lowest AIC is accepted. However, by

FIGURE 5.7: ACF and PACF Plot for Quantity Time Series

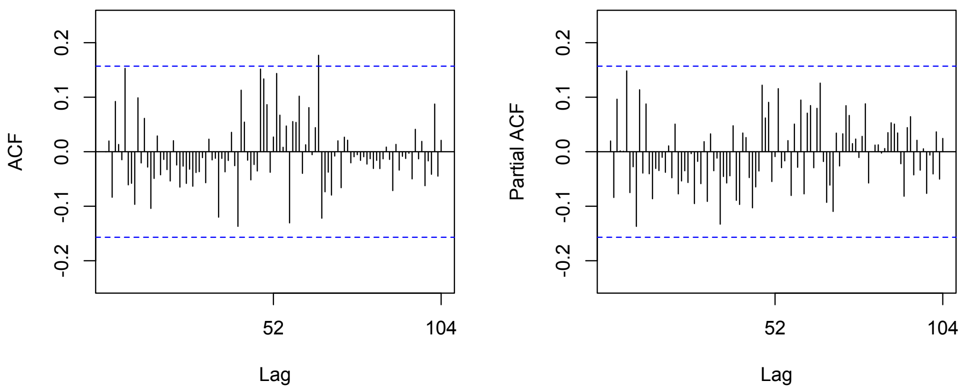


inspection the data doesn't seem stationary as the model suggests. This is confirmed by the Augmented Dicky-Fuller test where the null hypothesis $H_0 : \gamma = 0$ is failed to be rejected for the computed $\chi^2 = -2.875$. The time series is shown to be non-stationary and integrated of order one $I(1)$. This is further confirmed by the KPSS-test. Consequently the differenced model is used. The coefficients for the $ARIMA(1, 1, 1) \times (0, 1, 1)^{52}$ are $\phi = 0.4389$, $\theta = -0.9703$, and $\Theta = -0.9985$. In expanded form the seasonal ARIMA can be stated as

$$\Delta y_t = 0.4389 y_{t-1} + \epsilon_t - 0.9703 \epsilon_{t-1} - 0.9985 \epsilon_{t-52} + (-0.9985 \times -0.9703) \epsilon_{t-53}$$

The ACF and PACF for the residuals can be seen in Figure 5.8. And while most of the significant lags are eliminated, the ACF plot still suggest a barely significant lag at lag 60. The residuals are none the less assumed to behave like white noise for the upcoming simulations. The result of implementing the conditional simulation method

FIGURE 5.8: Residual ACF and PACF Plot for Quantity Time Series



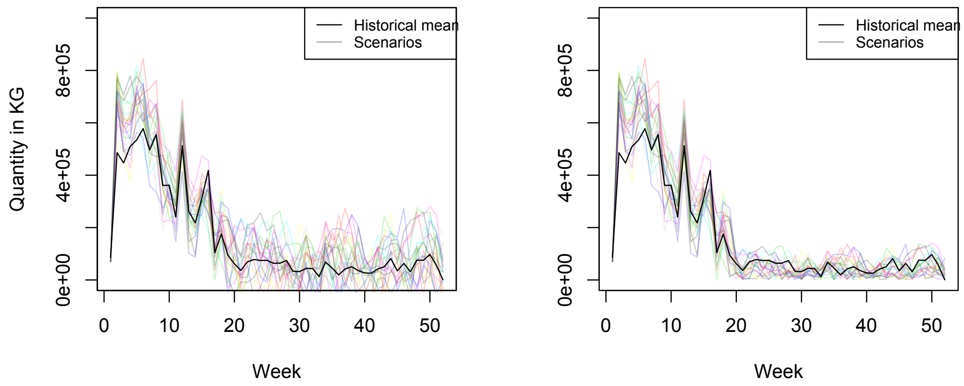
in R yields the scenarios depicted to the left in Figure 5.9. The mean of the historical data is superimposed for comparison, and it's clear the simulations are following the conditional path set during implementation.

Some *Ad hoc* modifications to the conditional simulation has been made. First, the returned generated scenarios has been changed to only contains positive numbers as there is no such thing as a negative harvest. Furthermore, as the busiest, and thus the most volatile, season for harvesting cod is mostly contained in the period between week 1 and week 20, the simulated scenarios has been smoothed and made less volatile for the remaining 32 weeks of the year to better reflect the reduction in variance for these periods. The results of this process can be seen to the right in Figure 5.9. The modified algorithm implementation can be found in appendix A.1.3. The original algorithm and implementation in R can be found in Sak and Hörmann (2012).

5.3 Scenario Stability

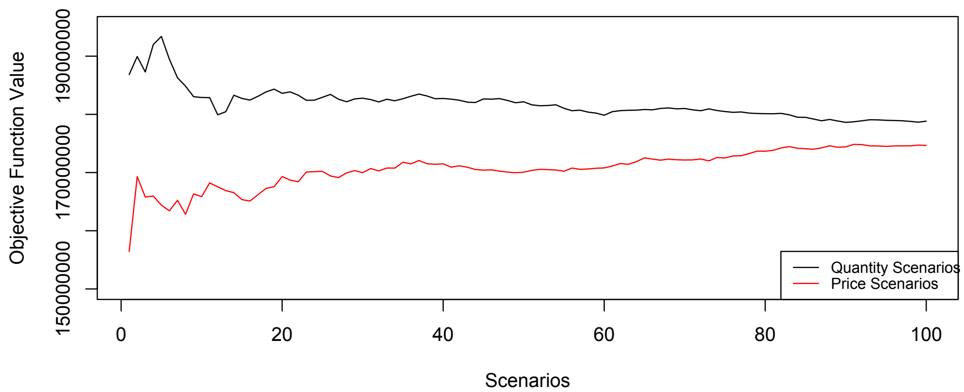
To determine whether or not the previously generated scenarios are somewhat yielding accurate and close to true estimates, various tests should be performed to establish their stability. This will also help establish the amount of scenarios necessary for relatively stable results. The amount of scenarios necessary can be approximated by observing the convergence of the objective value (Carrión et al., 2007), i.e. such that the value of the maximized objective function, $\max \Pi(\mathbf{x}, \xi)$,

FIGURE 5.9: Unmodified (left) and modified (right) conditional scenarios on quantity in KG



is iteratively generated in a progressively fashion for the same seeded Mersenne Twister starting with one scenario and ending in a predefined number of scenario. This allows investigating for what amount of scenarios stops changing the objective value in a critical way. This process of iteratively checking for a objective value stability can be seen in Figure 5.10 for both the quantity and price scenarios, in black and red, respectively. As seen in the figure, both are exhibiting quite

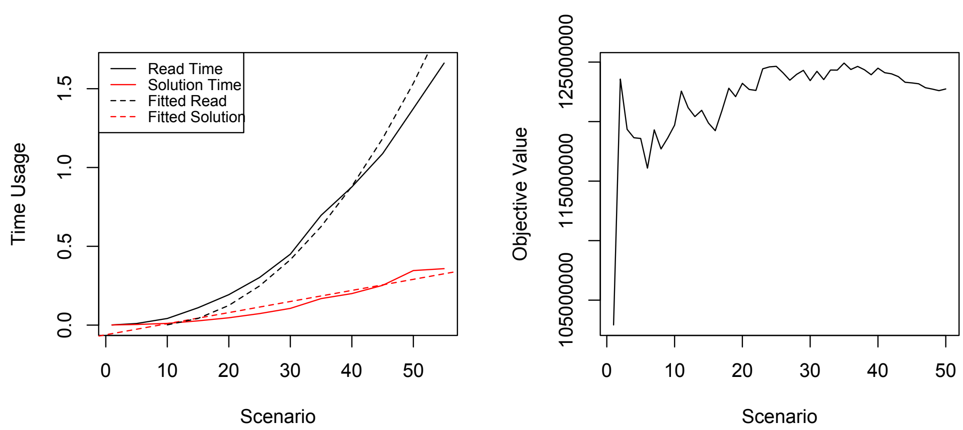
FIGURE 5.10: Objective Function Convergence



unstable behaviour for the first 40 scenarios or so, which is to be expected as for few scenarios, more extreme results will occur. When increasing the amount of scenarios to 100, both seems to be starting to converge to a more stable objective function value. The difference between the highest and the lowest value for the whole process, i.e. for the objective function value for all the different cases, is 11.77% and 8.27% for quantity and price, respectively. For the values between 80 and 100, and 90 and 100, the difference is 0.86% and 0.67%, and 0.25% and 0.24%, respectively, for quantity and price. Consequently it can be a fair assumption that the objective value is closing in on its true value for the respective seed for the PRNG already at 100 scenarios, and while more scenarios preferably should be used to asses the convergence, this amount is kept for computational simplicity, even though the solution time is close to instantaneous at this point.

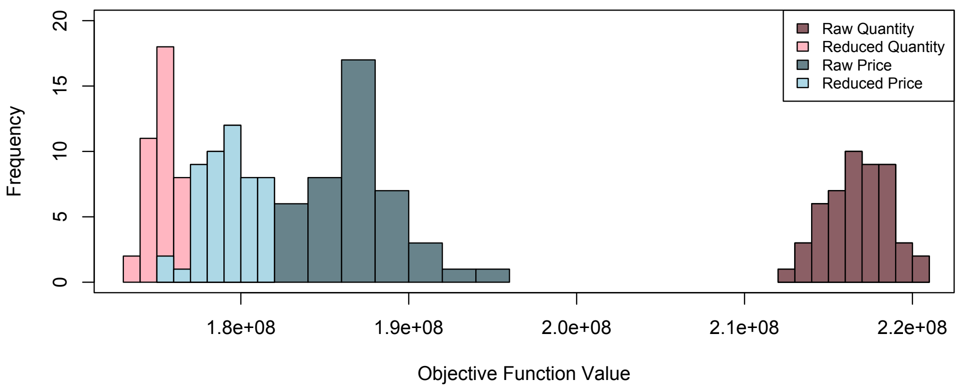
The combined stochastic program with stochastic variables both in price and quantity, the results can be found in Figure 5.11. For the current implementation of this model, the time it takes to solve is unreasonable high. Both the read time and solution time can be observed to the left in Figure 5.11 for a model with $t = 20$. For 100 scenarios, the program contains 800000 variables. As seen, the read time far exceeds the solution time. The dotted lines shows the ordinary least squares trends. For the solution time, its close to linear with an $O(n)$ efficiency. As for the read time, the efficiency is atleast $O(n^2)$ due to its nested implementation. While not a big increase for smaller problems, the difference can be great for large numbers of variables. The lower efficiency in the read time can suggest an inefficient implementation of the model by the user. This is further confirmed by testing the combined program for $t = 8$ with 100 scenarios. This program should consist of roughly the same amount of variables as the single stochastic programs, around 25000, and still suffers the unreasonable slow read time. Consequently, the convergence is only observed up to 50 scenarios, and can be seen to the right in Figure 5.11. As expected, it's still unstable at this point. Consequently, and due to its instability, the combined model will not be evaluated further in this thesis.

FIGURE 5.11: Combined Model Time and Convergence



To further test the stability of the generated scenarios, the in-sample and out-of-sample stability can be evaluated (Kaut and Wallace, 2003). The in-sample stability can be defined as $\Pi(\mathbf{x}_k^*; \hat{\xi_{tk}}) \approx \Pi(\mathbf{x}_l^*; \hat{\xi_{tl}})$, i.e. for K different scenario trees generated by the same scenario generating method where $k, l \in K$, and the optimal solution vector $\mathbf{x}_k^*$, for it to have in-sample stability, they should be fairly similar in their objective value. For the out-of-sample stability, the true distribution must be known such that $\Pi(\mathbf{x}_k^*; \tilde{\xi_t}) \approx \Pi(\mathbf{x}_l^*; \tilde{\xi_t})$. To observe the out-of-sample stability is difficult as the true distribution of the random variables are difficult to obtain, although it follows that if the scenarios are perfect in-sample stable, they must also exhibit out-of-stable stability. While the in-sample stability is a good measure even when evaluating stand alone scenario generating methods, the relative measures towards other methods increase the usefulness of the tests. Consequently both the original quantity and price scenarios are compared to their reduced versions, i.e. compared to the *ad hoc* and moment-matched versions, respectively. The in-sample stability tests are done by generating 50 separate situations containing 100 scenarios each for the 4 types of generating procedures for a total of $4 \times 50 \times 100$ scenarios. For each separate situation, a new seed is given to the Mersenne Twister such that different sets of random white noise is used. The objective value for each situation is then estimated and analysed. The distribution of the two scenario generating methods, and their reduced versions, can be seen in Figure 5.12.

FIGURE 5.12: In Sample Stability Distribution



The stability is measured through several different test statistics for central tendency and dispersion, most of them quite self explanatory and can be seen in Table 5.1 and Table 5.2 for quantity and price scenarios, respectively. The relative range deviation, RRD, and relative standard deviation (coefficient of variation), RSD, are defined as $RRD = \frac{Range}{\mu}$ and $RSD = \frac{\sigma}{\mu}$. Consequently they measure the distance of range and standard deviation from the mean as a fraction, and is expressed in percentages. Scenarios exhibiting stable behaviour will have low values of RRD and RSD, and further these measures of dispersion allows for an easy cross comparison of the raw and reduced scenarios.

TABLE 5.1: In-Sample Test Statistics for Quantity Scenarios

Test Statistics	Raw Scenarios	Reduced Scenarios
Min	212 898†	175 043
Max	220 586	181 965
Range	7 688	6 922
Mean	216 720	179 170
Standard Deviation	1 874	1 560
RRD	3.55%	3.86%
RSD	0.86%	0.87%

†Numbers in NOK thousands

The in-sample stability for the quantity scenarios can be found in Table 5.1. Both the RSD amd RMD are similar for both the raw and the reduced version, and consequently they exhibit close to similar stability. As for the range of the test in itself, depending on the situation, it can be both acceptable and too high. As the scenario generation is not compared to other fundamental different types of generators, possibility of improvements can not be stated with certainty, but it is likely that a different scenario generating procedure could yield more stable results. The mean value of the two deviates quite a lot. The reason behind this is the highly volatile periods after week 20 in the raw version, and consequently the harvest of cod is overestimated for the periods 20 to 52. Also, by examining the distributions in Figure 5.12, the objective function values of the raw quantity scenarios seems to be greatly overestimated.

TABLE 5.2: In-Sample Test Statistics for Price Scenarios

Test Statistics	Raw Scenarios	Reduced Scenarios
Min	177 508†	173 161
Max	195 338	179 375
Range	17 830	6 213
Mean	186 025	175 889
Standard Deviation	3 652	1 369
RRD	9.58%	3.53%
RSD	1.96%	0.78%

†Numbers in NOK thousands

The in-sample stability of the price scenarios can be seen in Table 5.2. Contrary to the quantity scenarios, for price the reduced scenarios greatly outperforms the raw, both in terms of RRD and RSD. While the same holds here with regards to the immediate interpretation of the isolated range of the reduced scenarios. Whether this is acceptable or not, depends on the situation for which the stochastic program is intended. It is also reasonable here to assume there exists scenario generating methods exhibiting greater stability, while at this time no further tests are conducted.

5.4 Results

Both the underlying and the stochastic program has been calculated. The results of these estimation can be found in Table 5.3. The value of the stochastic solution (VSS) (Birge, 1982) is also found. The VSS can be viewed as a measurement of the increased expected value of using the stochastic program instead of the of the deterministic one. The expected value problem (EEV) can be found by solving the deterministic program with the optimized values for the stochastic program, i.e. the $\arg\max \Pi^s(\mathbf{x})$. Consequently the objective value for this program will always be lower than for the stochastic program such that $\max \Pi^s(\mathbf{x}) > \max \Pi(\mathbf{x})_{EEV}$. The measured VSS for these programs are 3.71% and 1.29%, for price and quantity, respectively. The implication of this is that solving the stochastic program yields an increase in the expected profits of 3.71% and 1.29%, respectively. These values for VSS are quite low. The reason for this might be because of the simplicity of the mathematical program solved in this case, and is expected to increase in an increase in complexity in the mathematical program, as seen in Bakhrankova et al. (2014) where VSS for all variations of stochasticty are *super* 10%. With small values of the VSS, it's difficult to conclude whether or not the value of solving the stochastic program is sufficient, and must be evaluated at an *ad hoc* basis. In addition to the VSS, the volatility of the value of the objective function must be taken into consideration.

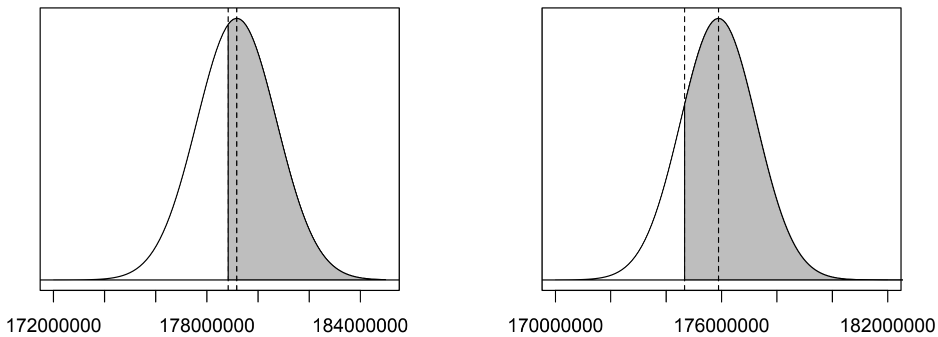
TABLE 5.3: Objective Values

Category	Stochastic Price	Stochastic Quantity
Deterministic Program	174 448†	183 468
Stochastic Program	174 665	178 831
EEV	168 424	176 558
VSS	6 241 (3.71%)	2 273 (1.29%)

†Numbers in NOK thousands

In most frameworks, there exists a need to know if the modified value chain is able to yield increased profits to at least a given level, for which the given level can be the depreciation rate of the real capital invested, or other recurring costs for which the profits must surpass to be a valuable investments. While in an ideal situation, the in-sample stability as discussed in the previous section should be such that when in-sample stability is found, so is the out of-sample stability as well. This is quite unlikely, and consequently the central limit theorem (Hill et al., 2008) argues that the distribution of the maximized objective values found in the sensitivity analysis will, for sufficiently high enough simulations, approximate the normal distribution. This theorem can be used to calculate the probability for the program to yield at least the required value to generate profits negating the investment cost, i.e. given a yearly cost of the new physical capital δ_c for some technology, c, then the probability $P(\Pi(\mathbf{x})^c < \Pi(\mathbf{x}))$ i.e. the probability to obtain a profit which is higher than the critical profit required to out earn the new cost of the newly invested physical capital or technology.

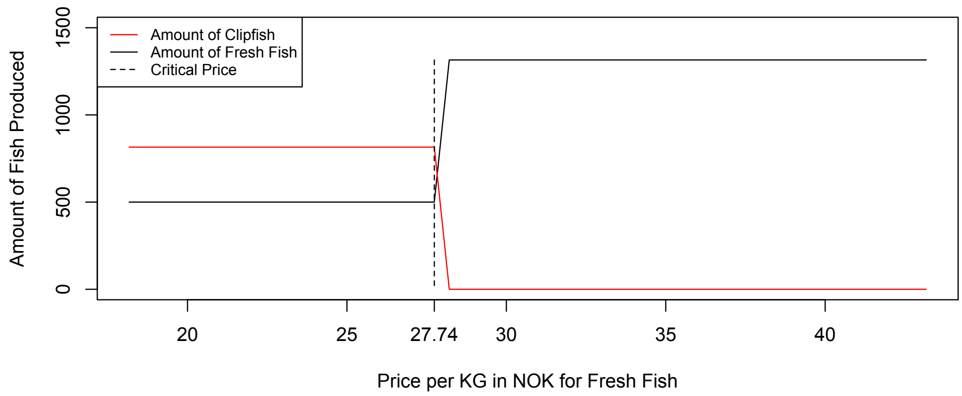
FIGURE 5.13: Normal Approximations of Stability Tests for Quantity on the left, Price on the Right



The normal approximations and their respective probability spaces for the stochastic model objective values can be seen in Figure 5.13 for quantity and price, respectively. Consequently the probability of obtaining atleast the calculated stochastic values presented in Table 5.3 are $P(\max \Pi(\mathbf{x})^s < \Pi(\mathbf{x})) = 58.6\%$ for quantity, and for price $P(\max \Pi(\mathbf{x})^s < \Pi(\mathbf{x})) = 81.42\%$. Furthermore, the confidence intervals can be calculated such that for the probability of obtaining the measured mean, $\overline{\Pi(\mathbf{x})} \pm t_{\alpha/2}\frac{\mu}{\sqrt{n}}$, such that for quantity, the confidence interval is $[178\,726', 179\,613']$ and for price, $[175\,499', 176\,277']$, for $\alpha = 0.05$. If these results are satisfactory highly depends on the dispersion of the results, and consequently the stability of the scenario generating method is of great importance when using a stochastic program for evaluating upgrades in technology, e.g. technology for reducing the time taken to dry the fish in a satisfactory way, or technology upgrades for freezing investigated by Bakhrankova et al. (2014). A comparison of their results is not very productive as their method of generating scenarios differs greatly from the method applied in this thesis, in addition the complexity and included variables of their program far exceeds the mathematical program here.

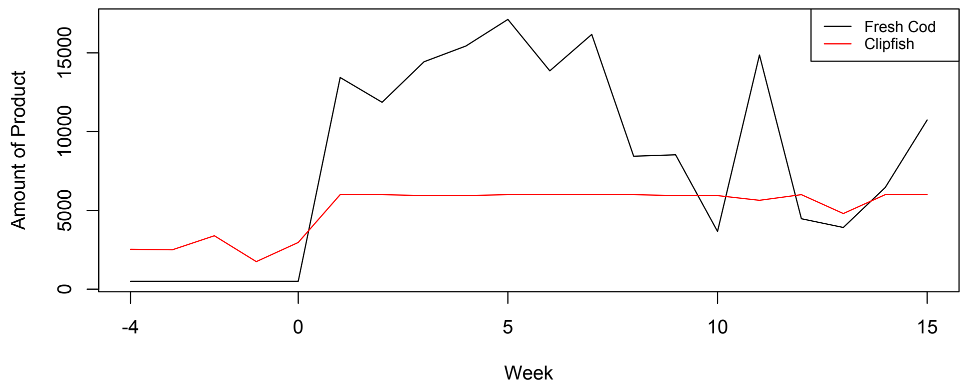
In addition to an analysis of upgrades and investments explained above, the model can be used for different types of sensitivity analysis. And while the complexity of such an analysis will indeed increase in the numbers of variables included in the model, extracting usable data is simple enough with a framework which allows for it. The difficulty lies in interpreting the results and the external effects outside the model which may affect the decisions. Due to the small number of decision variables and the deterministic behaviour of the price of clipfish, for this program, most of the time periods is optimized to be either-or in what to most optimally produce. There exists some periods though where a critical price is reached and the production efforts are allocated for a different product. As an example, consider Figure 5.14 for $t = 33$. The figure shows the critical price

FIGURE 5.14: Critical Price



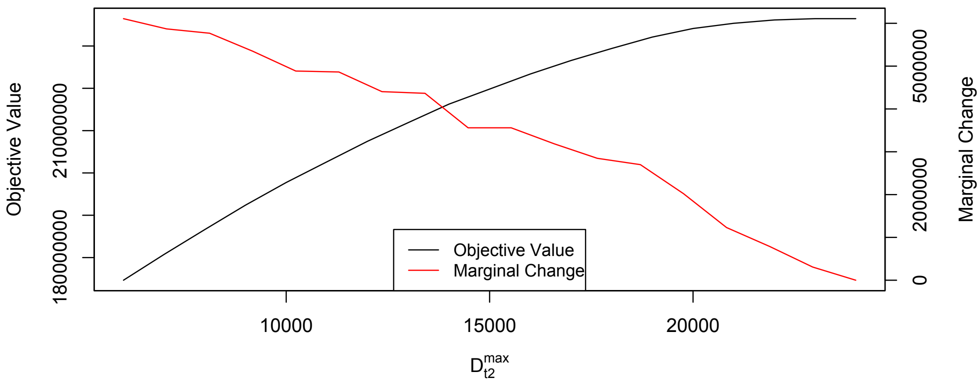
for which production should be shifted from full clipfish production, subject to the strategic restriction for production of fresh cod, to only production of fresh cod. However, these results must be used with caution. While the critical prices would show the optimal allocation of production for the model, the model considers the price and quantity path simulated by the scenario generator. Consequently, external effects such as policy change, bad harvest or low stock size of fish, bad weather, *et cetera*, is not considered by these paths unless it's repeating consistently on a yearly basis. For the mentioned external effects, this is unlikely. While announced policy changes are slow, they may take effect quite fast even before implemented (Lucas, 1976). This effect makes it difficult to adjust decisions, while it is crucial that it be taken into consideration using the stochastic program as a decision maker for production. For both the weather and other factors causing slow harvest e.g. low spawn rates, the other strategic decisions must be made on an *ad hoc* basis. Consequently, the stochastic program should be used as a supplement in decision making, and further must be updated on a regular basis to take trends and other factors into consideration.

FIGURE 5.15: Long Term Production



With regards to the long term production, for the whole time duration the model doesn't perform as well due to the low harvest during the last 30 weeks of the year and with only a single input factor, namely cod. For the most active period, the production of both the fresh and the clipfish is depicted in Figure 5.15. The plot includes the incoming harvest for the last 4 weeks of the previous year. As seen the production cap for clipfish is reached immediately when the harvest is sufficiently high. The production is kept at this level for the duration of the model, only declining when the access for raw material declines. The main reason behind these results are the deterministic behaviour of the clipfish prices, making them yield high enough profits to outperform the fresh according to the mathematical model. These results are expected to be different if randomness were to be added to these prices as well. If one were to consider the hypothetical case where the clipfish in fact should be produced at capacity every period, the fish processor should consider investing in higher production capabilities for the clipfish. A model like this is able in a quick and simple way to evaluate the payoff through its shadow price. The shadow prices, i.e. the marginal profit potential given a slack in the constraints, can be used to analyse the benefits of relaxing the constraints. The shadow price, also known as the Lagrangian multiplier, can be more formally defined as $\left.\frac{\partial \Pi(\mathbf{x})}{\partial a_i}\right|_{\mathbf{x}^*} = \lambda_i$ i.e. the marginal change in the objective function evaluated at the optimal production vector, $\mathbf{x}^*$, subject to a change in the constraint, a_i. Such a slack in the constraints is only possible if the capacity is expanded. For the model in Figure 5.15, the shadow price is found to be $\lambda = 6342$ NOK. Consequently for a marginal increase in capacity clipfish production, the company would be 6342 NOK better off per year after the first KG increase. The evolution of the objective value and the marginal change can be seen in Figure 5.16 where the objective value is measured on the left y-axis, and the marginal change on the right. The x-axis shows the capacity restraint, D_{t2}^{max} which as mentioned earlier is set to 6000 KG per week. For this program, the processing plant should continue to increase its capacity until the marginal cost of the increased capacity equals the marginal change in the objective value. As seen in the figure, the marginal change is positive for all capacity increases, although in a decreasing manner such that $\frac{\partial \Pi(\mathbf{x})}{\partial D_{t2,max}} > 0$ and $\frac{\partial^2 \Pi(\mathbf{x})}{\partial D_{t2,max}^2} < 0$. As a direct consequence of this, an increase in the capacity should be considered only when the marginal benefits intersects the marginal costs. Such an analysis is not conducted in this thesis.

FIGURE 5.16: Marginal Change in Objective Function



Chapter 6

Conclusion

In this thesis, a stochastic optimization approach through scenario generation and mathematical programming has been applied to an integrated fishery. The results presented in Bakhrankova et al. (2014) and in this thesis indicate that such a method indeed might be viable both for complex problems, and smaller ones. As for the specific results obtained in this thesis, it's difficult to conclude on this basis alone. Further tests must be conducted both in terms of scenario generation, and in terms of an increased complexity on the mathematical program combined with the stochastic variables. With regards to applicability, and due to its low complexity, the model is far from implementable in its current form. Furthermore, even with an increased and closer to reality complexity, such a method of optimizing and analysing the value chain might be problematic, as suggested by Mikalsen and Vassdal (1981). They underline the problem of being able to both understand the process in applying such a method, and even interpreting the results yielded by a complete model. While this was especially true back then, 20-30 years ago, when the computational power mostly were accessible through centralized data centres, this still applies today. And despite the more than doubling of computer power per year (Schaller, 1997; Mack, 2011) since then which allows for millions of variables to be solved on simple computer set-ups, the framework for doing mathematical optimization is still a complicated one. Sensitivity analysis and best/worst case analysis is difficult when the amount of parameters and decision variables increase in numbers (Huang and Loucks, 2000). Consequently a method such as this might not be suitable for smaller fisheries without a centralized organisation or without external assistance in implementation, framework development, and maintenance. This might make the benefits of applying such a model be outweighed by the costs. This is further underlined by the fact that the case study for this thesis were unable to supply detailed enough data for a more complex and accurate model in the time required. Despite this, taking uncertainty into account is indeed important for a fishery processing plant to be able to produce according to the WAPM. Stochastic programming has in this thesis been shown, albeit with some uncertainty, to be a tool allowing for production closer to ones profit maximizing production set. The results is expected to further be increase by increasing the complexity in the mathematical model and accuracy of the scenarios generated.

For increasing the accuracy of the scenarios, different scenario generating methods should be explored and compared to the ARIMA method applied in this thesis. While the in-sample stability might suggest good results in some cases, it is dangerous to conclude based on only these results alone. And as suggested by Di Domenica et al. (2009), *super* 3% RRD and the calculated RSDs turns out to be more unstable compared to the conditional moment matching and the bootstrapping methods they applied for an extended Newsvendor model. While the methods should not be compared across different mathematical programs, and certainly not based on the low complexity of the program for this thesis, the results show that comparisons of different methods should be done to be able to find the most stable way of generating scenarios. And while this is important, many methods of generating scenarios are heuristic in its nature which underlines the importance of developing a framework which allows for simple comparisons of methods. Furthermore, the time series approach for quantity is heavily reliant on the assumption that the specified quotas, i.e. the total allowable catch (TAC) for a defined vessel group, is relative stable over the horizon of the scenarios generated and the underlying characteristics of the data they are based upon. And while the TAC has been fairly stable for the time horizon used for scenario generation in this thesis, it saw a small decline in 2015, while set at 2015-levels for 2016. As these quotas are set by the Directorate of Fisheries subject to stock sizes, spawn rates, and other bilateral agreements, they are uncertain and must be taken into consideration when evaluating the scenarios subject to the scenario generating method (Millar and Gunn, 1992; Heen et al., 2014).

For the stochastic program to be applicable to a real setting, its complexity must be heavily expanded upon. Consequently, the true value chain must be reflected in the program in a much more realistic way. To do this, the variable costs must be divided into several subgroup to measure more precisely across the different sections of the value chain. Furthermore, the value chain must be divided into a more correct depiction of a real value chain. As for the model in this thesis, only the production part of the chain is considered for a predefined variable cost. Suggestions to expansion of the value chain should include, but not limited to, both work hours and time usage at landing site, both for landing to further process the raw material, and directly packaging and shipping. Further, the work hours and wage should be defined for the production processes at the different stages of drying and salting, which would further be increased in performance to take account of variable electricity cost. To be able to asses storage possibilities and system, this must be included as well where power cost must be estimated. In addition, and as a result of the difficult task of monitoring the quality of the fish, modelling this uncertainty through chance constraints (Simbolon et al., 2014) should be considered if no further efforts are done to improve on this.

More efficient implementation of the combined model must be carried out to better test a full size model. As for now, the implementation of the combined model is to slow and unstable to evaluate. This is most likely due to inefficient implementation by the author, and not the algorithm applied for solving the problem itself. Despite the inefficient implementation, more complex problems require a great deal more implementation efforts. This is further underlined in Bakhrankova et al. (2014) which for simplicity omitted a more realistic way of generating scenarios, and further states that to increase the realism in the scenarios would require much more work with regards to programming and implementation to the point where new algorithms must be defined in a more heuristic manner. Furthermore, and which would be true both for the model with stochastic income of raw material and the combined model, it's necessary to omit the lower constraint on the production i.e. D_{tk}^{min} as the last 32 weeks risk yielding fairly low estimates in some scenarios and time periods. This is indeed a weakness in the model, and it should be considered dividing it up in two parts such that one plans production for the most active weeks i.e. week 1 to week 20, and another one for the rest. While to divide the model into longer stretches for analysis is straight forward, the accuracy of the results might suffer. To adjust for this, its suggested to introduce the whole spectre of raw materials, i.e. cod, haddock, and herring, and their respective final products as well. This will allow for a full sized model. In addition, and as seen in Figure 5.16, and while there exists diminishing returns increasing the capacity, the potential gain is infinite. This is due to the lack of penalty restriction in the model e.g. overproduction and alternative costs not already included in the model. Consequently the marginal increase in reducing the constrictions of producing clipfish will be reduced and yield more accurate results.

Appendix A

Source Code

A.1 R Code

A.1.1 Price Scenario Generation $ARIMA(0,1,1)$

```
prisscens<-function(model, data, uker, scens)
{
  first.white <- residuals(model)[length(data)]
  e<-replicate(scens, rnorm(uker+1, 0, sqrt(model$sigma2)))
  e[1,1:scens] <- first.white
  ad<-diff(data)

  y <- matrix(0, uker, scens)

  mu<-mean(ad)
  theta<-coef(model)[1]
  i<-0
  for(i in 1:scens)
  {
    i<-i+1

    y[,i] <- e[2:(uker+1),i]
    y[,i] <- y[,i] + theta*e[1:uker,i]
    y[,i] <- y[,i] + mu

  }

  rawscens<-diffinv(y, differences = 1, xi =
      matrix(data[1], 1, scens))
  rawscens<-rawscens[-1,];rawscens[1,]<-data[1]
  return(rawscens)

}
```

A.1.2 Scenario Reduction

```
scenred<-function(scengenmatrix, control, int)
{
  bestatt<-matrix(,nrow(scengenmatrix),ncol(scengenmatrix))
  i<-0
  upper<-(1+int)
  lower<-(1-int)
  for(i in 1:ncol(scengenmatrix))
  {
    i=i+1

    if(mean(scengenmatrix[,i])<=mean(control)*upper &&
        mean(scengenmatrix[,i])>=mean(control)*lower &&
        skewness(scengenmatrix[,i])<=skewness(control)*upper &&
        skewness(scengenmatrix[,i])>=skewness(control)*lower &&
        sd(scengenmatrix[,i])<=sd(control)*upper &&
        sd(scengenmatrix[,i])>=sd(control)*lower)
       {
            bestatt[,i]<-scengenmatrix[,i]
       }
  }
  bestatt<-bestatt[,!apply( bestatt,2,function(x) all(is.na(x)))]

  return(bestatt)
}
```

A.1.3 Conditional Simulation $ARIMA(1,1,1)(0,1,1)_s$

```
arima.condsimsmooth <- function(object, x, n.ahead, n ,adjer)
{
  L <- length(x); coef <- object$coef;
  arma <- object$arma; model <- object$model;
  p <- length(model$phi); q <- length(model$theta)
  d <- arma[6]; s.period <- arma[5];
  s.diff <- arma[7]
  diff.xi <- 0;
  dx <- diff(x, lag = s.period, differences=s.diff)
  diff.xi[1] <- dx[length(dx) - d + 1];
  dx <- diff(dx, differences = d)
  diff.xi <- c(diff.xi[1], x[(L - s.diff * s.period + 1):L])
  start.period<-20
  end.period<-52
  interval.red<-2
  res[, r] <- abs(xc)
  p.startIndex <- length(dx) - p
  start.innov <- NULL
```

```
  start.innov <- residuals(object)[(L - q + 1):(L)]

  res <- array(0, c(n.ahead, n))
  for(r in 1:n){
    innov = rnorm(n.ahead, sd = sqrt(object$sigma2)/adjer)
    e <- c(start.innov, innov)
    xc <- array(0, dim = p + n.ahead)
      for(i in 1:p) xc[i] <- dx[[p.startIndex + i]]
       k <- 1
        for(i in (p + 1):(p + n.ahead)){
          xc[i] <- e[q + k]
          xc[i] <- xc[i] + sum(model$theta * e[(q + k - 1):k])
          xc[i] <- xc[i] + sum(model$phi * xc[(i - 1):(i - p)])

          k <- k + 1
        }
    xc <- as.vector(unlist(xc[(p + 1):(p + n.ahead)]))
    xc <- diffinv(xc, differences = d, xi = diff.xi[1])[-c(1:d)]
    xc <- diffinv(xc, lag = s.period, differences = s.diff,
         xi = diff.xi[2:(s.diff * s.period + 1)])
    xc <- xc[-(1:(s.diff * s.period))]
  }
  res[start.period:end.period,1:n]<-
     res[start.period:end.period,1:n]/interval.red
  res[1, 1:n]<-x[1]

  return(res)
}
```

A.2 SAMPL Code

A.2.1 Underlying Deterministic Program

```
param NT;
param NP;

set product := 1..NP;
set time := 1..NT;
set time1 within{time} := 5..NT;

param sellprice{time, product};
param varcost{product};
param procurprice{time};
param rawmatRequired{product};
param rawmat{time};
param stockfishred;
```

```
param minstock;
param maxstock;
param minfresh;

var sellprod{t in time, f in product} >= 0;
var makeprod{t in time, f in product} >= 0;

maximize profits: sum{t in time1, f in product
   }rawmatRequired[f]*sellprice[t,f]*sellprod[t,f]
   - sum{t in time1, f in product}
   rawmatRequired[f]*varcost[f]*makeprod[t,f]
   - sum{t in time1}procurprice[t]*rawmat[t];

subject to
   balance{t in time}:
     stockfishred*rawmatRequired[2]*makeprod[t,2]
     + rawmatRequired[1]*makeprod[t,1]
     = rawmat[t];
   salesfresh{t in time1}:
      sellprod[t,1] = makeprod[t,1];
   salesstock{t in 1..NT-4}:
      sellprod[t+4,2] = makeprod[t,2];
   upperstock{t in time, f in product}:
     makeprod[t,2] <= maxstock;
   upperfresh{t in time, f in product}:
     makeprod[t,1] >= minfresh;
```

A.2.2 Stochastic Program

```
param NT;
param NP;
param NS;

set product := 1..NP;
set time := 1..NT;
set time1 within{time} := 5..NT;
set scen := 1..NS;

param sellprice{time, product, scen};
param varcost{product};
param procurprice{time};
param rawmatRequired{product};
param rawmat{time};
param prob{scen} := 1/NS;
param stockfishred;
param minstock;
```

```
param maxstock;
param minfresh;

var sellprod{t in time, f in product, s in scen} >= 0;
var makeprod{t in time, f in product, s in scen} >= 0;

maximize profit: sum{s in scen} prob[s] *
   (sum{t in time1, f in product}
   rawmatRequired[f]*sellprice[t,f,s]*sellprod[t,f,s]
   - sum{t in time1, f in product}
   rawmatRequired[f]*varcost[f]*makeprod[t,f,s]
   - sum{t in time1}procurprice[t]*rawmat[t]);

subject to
   balance{t in time, s in scen}:
      stockfishred*rawmatRequired[2]*makeprod[t,2,s]
      + rawmatRequired[1]*makeprod[t,1,s]
      = rawmat[t];
   salesfresh{t in time1, s in scen}:
      sellprod[t,1, s] = makeprod[t,1,s];
   salesstock{t in 1..NT-4, s in scen}:
      sellprod[t+4,2,s] = makeprod[t,2,s];
   nanat1{t in time1, f in product, s in 5..NS}:
      makeprod[1,f,1] = makeprod[1,f,s];
   nanat2{t in time, f in product, s in 5..NS}:
      sellprod[1,f,1] = sellprod[1,f,s];
   upperstock{t in time, f in product, s in scen}:
      makeprod[t,2,s] <= maxstock;
   upperfresh{t in time, f in product, s in scen}:
      makeprod[t,1,s] >= minfresh;
```

Appendix B

Algorithms

B.1 Algorithms

B.1.1 Scenario Generating Algorithm for Price Scenarios

```
generate a best fit ARIMA(p, d, q);
initialize scenario counter s ∈ {1, ..., N_s} and time periods
 t ∈ {1, ..., N_t};
initialization;
while s ≤ N_s do
    generate random errors ϵ_t ~ N(0, σ^2);
    generate y_ts with the simulated ϵ_t;
    update counters;
    if s = N_s then
        return scenario matrix S ∈ ℝ^{t×s};
    else
        repeat process;
    end
end
```

Algorithm 1: Scenario Generation Algorithm

B.1.2 Scenario Reduction Algorithm

for the control time series, c;
for $N_s = ncol(\mathbf{S})$;
initialize scenario reduction counter $r \in \{1, ..., N_s\}$;
initialization;
while $r \leq N_s$ **do**
  update counters;
  if $\mu_r = \mu_c \,\&\, \sigma_r = \sigma_c \,\&\, skew_r = skew_c \,\&\, kurt_r = kurt_c$ **then**
    store scenario r;
    repeat for scenario $r + 1$;
  else
    discard scenario r;
    repeat for scenario $r + 1$;
  end
  when $r = N_s$, return reduced scenario matrix $\mathbf{R} \in \mathbb{R}^{t \times r}$;
end

Algorithm 2: Scenario Reduction Algorithm

Bibliography

Akaike, Hirotogu. "Information theory and an extension of the maximum likelihood principle". In: *Selected Papers of Hirotugu Akaike*. Springer, 1998, pp. 199–213.

Anton, Howard and Chris Rorres. *Elementary linear algebra: with supplemental applications*. Wiley, 2011.

Bakhrankova, Krystsina, Kjetil T Midthun, and Kristin T Uggen. "Stochastic optimization of operational production planning for fisheries". In: *Fisheries Research* 157 (2014), pp. 147–153.

Becker, Richard A, Allan R Wilks, Ray Brownrigg, Thomas P Minka, and Alex Deckmyn. *maps: Draw Geographical Maps*. R package version 3.1.0. 2016. URL: `https://CRAN.R-project.org/package=maps`.

Begen, Mehmet Atilla and Martin L Puterman. "Development of a catch allocation tool design for production planning at JS Mcmillan fisheries". In: *Infor-Information Systems and Operational Research* 41.3 (2003), pp. 235–244.

Benders, Jacques F. "Partitioning procedures for solving mixed-variables programming problems". In: *Numerische mathematik* 4.1 (1962), pp. 238–252.

Birge, John R. "The value of the stochastic solution in stochastic linear programs with fixed recourse". In: *Mathematical programming* 24.1 (1982), pp. 314–325.

Blanchard, Olivier Jean. "[The New Neoclassical Synthesis and the Role of Monetary Policy]: Comment". In: *NBER Macroeconomics Annual* 12 (1997), pp. 289–293.

Box, George EP, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Carrión, Miguel, Andy B Philpott, Antonio J Conejo, and José M Arroyo. "A stochastic programming approach to electric energy procurement for large consumers". In: *IEEE Trans. Power Syst* 22.2 (2007), pp. 744–754.

Charnes, Abraham and William W Cooper. "Chance-constrained programming". In: *Management science* 6.1 (1959), pp. 73–79.

Conejo, Antonio J, Miguel Carrión, and Juan M Morales. *Decision making under uncertainty in electricity markets*. Vol. 1. Springer, 2010.

Dabbene, F, P Gay, and N Sacco. "Optimisation of fresh-food supply chains in uncertain environments, Part I: Background and methodology". In: *Biosystems Engineering* 99.3 (2008), pp. 348–359.

Dantzig, George Bernard. *Linear programming and extensions*. Princeton university press, 1998.

Dasgupta, Indraneel. “Supply theory sans profit maximization”. In: *The BE Journal of Theoretical Economics* 9.1 (2009).

Dequech, David. “Neoclassical, mainstream, orthodox, and heterodox economics”. In: *Journal of Post Keynesian Economics* 30.2 (2007), pp. 279–302.

Di Domenica, Nico, Cormac Lucas, Gautam Mitra, and Patrick Valente. “Scenario generation for stochastic programming and simulation: a modelling perspective”. In: *IMA Journal of Management Mathematics* 20.1 (2009), pp. 1–38.

Dickey, David A and Wayne A Fuller. “Distribution of the estimators for autoregressive time series with a unit root”. In: *Journal of the American statistical association* 74.366a (1979), pp. 427–431.

Dominguez-Ballesteros, B, G Mitra, and NS Koutsoukis. *Modern Mathematical Programming Modelling Languages: A Comparative Review*. Tech. rep. Technical Report in preparation, Brunel University, 1999.

Gardner, G, AC Harvey, and GDA Phillips. “Algorithm AS 154: An algorithm for exact maximum likelihood estimation of autoregressive-moving average models by means of Kalman filtering”. In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 29.3 (1980), pp. 311–322.

Greenwald, Bruce C and Joseph E Stiglitz. *Asymmetric information and the new theory of the firm: Financial constraints and risk behavior*. Tech. rep. National Bureau of Economic Research, 1990.

Griva, Igor, Stephen G Nash, and Ariela Sofer. *Linear and nonlinear optimization*. Siam, 2009.

Gülpınar, Nalan, Berç Rustem, and Reuben Settergren. “Simulation and optimization approaches to scenario tree generation”. In: *Journal of economic dynamics and control* 28.7 (2004), pp. 1291–1315.

Gunn, EA, HH Millar, and SM Newbold. “Planning harvesting and marketing activities for integrated fishing firms under an enterprise allocation scheme”. In: *European Journal of Operational Research* 55.2 (1991), pp. 243–259.

Heen, Eirik Eriksen, Knut Heen, and PingSun Leung. “Conflicting goals in fisheries management—a study of the Norwegian cod fisheries”. In: *Marine Policy* 49 (2014), pp. 73–80.

Heitsch, Holger and Werner Römisch. “Scenario reduction algorithms in stochastic programming”. In: *Computational optimization and applications* 24.2-3 (2003), pp. 187–206.

Higle, Julia L. “Stochastic programming: optimization when uncertainty matters”. In: *Cole Smith J (ed) Tutorials in operations research* (2005), pp. 30–53.

Hill, R Carter, William E Griffiths, and Guay C Lim. *Principles of econometrics*. Vol. 5. Wiley Hoboken, NJ, 2008.

Høyland, Kjetil, Michal Kaut, and Stein W Wallace. “A heuristic for moment-matching scenario generation”. In: *Computational optimization and applications* 24.2-3 (2003), pp. 169–185.

Huang, GH and DP Loucks. “An inexact two-stage stochastic programming model for water resources management under uncertainty”. In: *Civil Engineering Systems* 17.2 (2000), pp. 95–118.

Hull, John C. *Options, futures, and other derivatives*. Pearson Education India, 2006.

Hyndman, Rob J. *forecast: Forecasting functions for time series and linear models*. R package version 6.2. 2015. URL: `http://github.com/robjhyndman/forecast`.

Hyndman, Rob J and Yeasmin Khandakar. “Automatic time series forecasting: the forecast package for R”. In: *Journal of Statistical Software* 26.3 (2008), pp. 1–22. URL: `http://ideas.repec.org/a/jss/jstsof/27i03.html`.

Kali, Peter and Stein W Wallace. *Stochastic programming*. Springer, 1994.

Kaut, Michal and Stein W Wallace. “Evaluation of scenario-generation methods for stochastic programming”. In: (2003).

Kwiatkowski, Denis, Peter CB Phillips, Peter Schmidt, and Yongcheol Shin. “Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?” In: *Journal of econometrics* 54.1 (1992), pp. 159–178.

Lucas, Robert E. “Econometric policy evaluation: A critique”. In: *Carnegie-Rochester conference series on public policy*. Vol. 1. Elsevier. 1976, pp. 19–46.

Mack, Chris A. “Fifty years of Moore’s law”. In: *Semiconductor Manufacturing, IEEE Transactions on* 24.2 (2011), pp. 202–207.

Matsumoto, Makoto and Takuji Nishimura. “Dynamic creation of pseudorandom number generators”. In: *Monte Carlo and Quasi-Monte Carlo Methods* 2000 (1998), pp. 56–69.

— “Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator”. In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 8.1 (1998), pp. 3–30.

Mikalsen, Bjørnar and Terje Vassdal. “A short term production planning model in fish processing”. In: *Applied operations research in fishing*. Springer, 1981, pp. 223–232.

Millar, HH and EA Gunn. “A two-stage procedure for planning marketing and fishing activities in fish-processing firms”. In: *Fisheries research* 15.3 (1992), pp. 197–215.

Naibaho, Tutiarny and Herman Mawengkang. “Nonlinear Mixed Integer Stochastic Programming Model for Sustainable Seafood Production and Distribution Planning Problem”. In: (2016).

Neumaier, Arnold and Oleg Shcherbina. “Safe bounds in linear and mixed-integer linear programming”. In: *Mathematical Programming* 99.2 (2004), pp. 283–296.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2015. URL: `https://www.R-project.org/`.

Randhawa, Sabah U and Einar T Bjarnason. “A decision aid for coordinating fishing and fish processing”. In: *European Journal of Operational Research* 81.1 (1995), pp. 62–75.

Römisch, Werner. “Scenario reduction techniques in stochastic programming”. In: *Stochastic Algorithms: Foundations and Applications*. Springer, 2009, pp. 1–14.

Saito, Mutsuo and Makoto Matsumoto. “Simd-oriented fast mersenne twister: a 128-bit pseudorandom number generator”. In: *Monte Carlo and Quasi-Monte Carlo Methods 2006*. Springer, 2008, pp. 607–622.

Sak, Halis and Wolfgang Hörmann. “Simulating the Continuation of a Time Series in R”. In: *arXiv preprint arXiv:1212.2393* (2012).

Samuelson, Paul Anthony. “Foundations of economic analysis”. In: (1948).

Schaller, Robert R. “Moore’s law: past, present and future”. In: *Spectrum, IEEE* 34.6 (1997), pp. 52–59.

Schütz, Peter and Asgeir Tomasgard. “The impact of flexibility on operational supply chain planning”. In: *International Journal of Production Economics* 134.2 (2011), pp. 300–311.

Sharma, Kailash Chand, Prerna Jain, and Rohit Bhakar. “Wind power scenario generation and reduction in stochastic programming framework”. In: *Electric Power Components and Systems* 41.3 (2013), pp. 271–285.

Shumway, Robert H and David S Stoffer. *Time series analysis and its applications: with R examples*. Springer Science & Business Media, 2010.

Simbolon, Hotman, P Siantar, and Indonesia Herman Mawengkang. “Stochastic Programming Model For Sustainable Fish Processed Production Planning Under Uncertainty”. In: *International Journal of Computer Science & Applications (TIJCSA)* 2.12 (2014).

Sobol’, Il’ya Meerovich. “On the distribution of points in a cube and the approximate evaluation of integrals”. In: *Zhurnal Vychislitel’noi Matematiki i Matematicheskoi Fiziki* 7.4 (1967), pp. 784–802.

Stiglitz, Joseph and Andrew Weiss. “Alternative approaches to analyzing markets with asymmetric information: reply”. In: *The American Economic Review* 73.1 (1983), pp. 246–249.

Trapletti, Adrian and Kurt Hornik. *tseries: Time Series Analysis and Computational Finance*. R package version 0.10-34. 2015. URL: `http://CRAN.R-project.org/package=tseries`.

Valente, Patrick, Gautam Mitra, Chandra A Poojari, and Triphonas Kyriakis. “Software tools for stochastic programming: a stochastic programming integrated environment (SPInE)”. In: (2001).

Varian, Hal R. “Microeconomic analysis”. In: (1992).

Whitle, Peter. *Hypothesis testing in time series analysis*. Vol. 4. Almqvist & Wiksells, 1951.

Zhou, Qun, Leigh Tesfatsion, and Chen-Ching Liu. “Scenario generation for price forecasting in restructured wholesale power markets”. In: (2009), pp. 1–8.