# Robust Production Planning for District Heating Networks


Jonatan Rantzer


Master's thesis
2015:E43


Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematics


CENTRUM SCIENTIARUM MATHEMATICARUM

**Abstract**

Efficient use of energy becomes increasingly important in the modern society, with climate change as a driving factor. Short term production planning for district heating networks is motivated by a customer demand that varies according to a daily cycle, but which is directly dependent on changes in temperature and customer behaviour. The planning involves challenges in modeling of production unit startups and shutdowns, as well as modeling of heat storage. This thesis suggests the use of model predictive control, with a two stage stochastic programming problem solved at each iteration. Tests are performed for systems with and without accumulation. For both cases, the results indicate that the method has potential to generate savings.

**Acknowledgements**

I would like to thank the collaborators at Modelon AB for creating a welcoming and friendly working environment. Especially I want to thank my supervisors from Modelon AB, Per-Ola Larsson and Stéphane Velut, for providing the idea of the thesis and for being available throughout the working process. Likewise, I want to express appreciation to my supervisor at LTH, Andrey Ghulchack, for his dedication in supporting me. Advice from Henning Larsson on the code design and other programming issues has also been a great help. Finally I am thankful for support from Gurobi Support and from the Pyomo Forum, in particular from David Woodruff.

# Contents

# Nomenclature

| **Symbol/Concept** | **Description** |
|---|---|
| $e_{i,S}$ | accumulator stored heat at time $i$ |
| $I_1$ | end time of Stage 1 |
| $I_2$ | end time of Stage 2 |
| $I_a$ | end time for interval where the demand is considered known |
| $I_b$ | end time for interval where the demand is considered unknown |
| percentile | realized value of a stochastic variable, which exceeds a given percentage of the realizations |
| $q_{i,k}$ | heat produced at hour $i$ by production unit $k$ |
| $q_{i,S}$ | heat transferred from accumulator at time $i$ |
| scenario | a supported realization of the random data in a two stage stochastic programming problem |
| SFB | solid fuel boiler |
| SP | stochastic programming |
| top-up unit | a unit that is meant to contribute to covering the heat load at times when the base load unit is insufficient |
| $u_{i,k}$ | on/off status for production unit $k$ at time $i$ (1/0) |
| WC | worst case (scenario) |
| $y_{i,k}$ | start-up status for production unit $k$ at time $i$ |
| $z_{i,k}$ | shutdown status for production unit $k$ at time $i$ |

The notation has been taken mainly from (Dotzauer, 2001) and (Arroyo and Conejo, 2004).

# 1 Introduction

A district heating network has the aim to efficiently provide customers with requested heat. Efficiency can be defined in different ways, such as low costs or minimal environmental impact, but in general means to make optimal use of the resources put into the system. A district heating network is a complex system and its efficiency depends on many different parameters: the type of production units and fuel, the isolation of the distribution pipes and the heat transfer in the customer substations, to mention a few. Furthermore, the efficiency is dependent on the operation of the network and specifically the planning of the production.

The customer heat demand varies with weather conditions and customer behaviour and thus it is not known in advance. Short term production planning is therefore dependent on load predictions in order to decide how much heat is to be produced. This introduces uncertainty and a need for robustness against unpredicted demand peaks.

The problem is further complicated by the fact that many district heating systems include several different (geographical) production sites, each one made up of several different production units (such as incineration plants, solid fuel boilers, fossil fuel boilers, etc.). As the demand varies over time it is necessary to startup and shut down production units, in order to adjust the amount of produced heat. However, the startup of a production unit requires both time and resources and is non-trivial to include in the optimization model.

Related to this, the short-term production planning can be divided into two separate tasks (Dotzauer, 2001, p. 1): the unit commitment problem and the economic dispatch. The unit commitment problem is the first step and has the purpose to set the binary variables describing when each production unit is to be turned on or off. The economic dispatch problem then sets continuous variables such as produced heat (or alternatively supply temperature).

Previous research projects at Modelon AB have developed models for each of these two problems. The unit commitment problem is formulated in discrete time with linear models and treats heat energy in terms of MW, without specifying the mass flow or the temperature (c.f. Equation (1)) . The results of the unit commitment are then used as input for the economic dispatch. The model used for economic dispatch is more physics-based and simulates the production process in continuous time. (Larsson et al., 2014)

As will be seen, in this thesis the discrete time unit commitment model (or an extension of it, see Section 4.1) is used to solve not only the unit commitment problem, but also the economic dispatch. In fact, the nature of the unit commitment problem solver is such that it is always solves the economic dispatch in parallel with the unit commitment. The purpose of dividing the problem in two is to use only the binary on/off variables achieved from the discrete time model and to get more accurate results for the continuous variables from the physics-based model. However, in the context of this report it is judged sufficient to work with the economic dispatch solution achieved from the discrete time model.

As mentioned already, the optimization requires a prediction on the heat demand, which can be made based on weather forecast, season and time of the day, as well as customer types (industry, residential, etc.). In the previous projects, optimization has been made based on a prediction of the expected demand, not taking into account the prediction uncertainty. However, (Larsson et al., 2014) also suggests simple strategies for how to tackle the uncertainty. These strategies are applied to the unit commitment step, but also affects the economic dispatch, since the economic dispatch builds on top of the unit commitment solution.

A straightforward method to handle uncertainties is what (Larsson et al., 2014, p. 40) calls the 'Wait and see'-method, corresponding to what in this project is called worst case optimization (see Section 4.3). This means basically to prepare for the worst case scenario and then to adjust the production plan when the time of the plan comes closer and thus the demand prediction becomes more trustworthy. The disadvantage with this method is that by always preparing for a high demand with low probability, there is a risk of wasting resources.

A second method presented formulates the optimization as a two stage stochastic programming problem. (Larsson et al., 2014, p. 40) A brief introduction to stochastic programming is given in Section 3.1. In contrast to the worst case optimization, stochastic programming takes into account not only the worst case scenario, but several scenarios for the demand with different probabilities. By doing this, the aim is to minimize the expected costs, while still making the production plan robust to deviations from the expected demand.

This thesis aims to further develop the stochastic programming approach and to evaluate its benefits. This is done by constructing a rigorous implementation of the optimization model and then by constructing realistic tests where choices have to be made for how to handle uncertain demand predictions. In particular the tests consider a case with two top-up units of different properties. The tests also treat networks with an attached accumulator, where heat can be stored.

# 2 Background

This background section starts by introducing district heating networks and customer demand predictions. It then goes on to describe mathematical models for production planning and ends by suggesting robust strategies using stochastic programming and MPC.

## 2.1 District Heating Networks

### 2.1.1 Purpose and Operation

A district heating network has three essential components: production units, distribution pipes and customers, see Figure 1. The purpose is to generate heat in central production units and to distribute it to the customers.[1] The distribution medium is water, which is heated at the production units and pumped out to the customers. Heat is given off at a customer substation[2] and the water is returned to the production units, where it is reheated.

The heat provided to a particular substation can be approximated as

$$Q = \dot{m} \cdot c_p \cdot \Delta T \tag{1}$$

where $\dot{m}$ is the mass flow through the substation, $c_p$ is the specific heat capacity at constant pressure and $\Delta T$ is the temperature difference between local supply and return temperatures. Thus, an increased heat demand can be supplied either by increased mass flow or by increased supply temperature.

### 2.1.2 Sources of Heat Demand

The heat load can be divided into three components: building heating, hot water and distribution losses. In a Swedish context the approximate share that each component has of the total load is 70, 25 and 5 percent respectively. (Kvarnström et al., 2007, p. 5) The building heating demand is dependent on weather conditions such as outdoor temperature, whereas the hot water demand is rather dependent on customer social behaviour.

### 2.1.3 Varying Heat Demand

The customer demand varies with season and with time of day, see e.g. Figure 2 Compare to figure in (Frederiksen and Werner, 2013, p. 93) showing the heat load variation for the entire district heating system in Helsingborg, Sweden.

Naturally, it is desirable to adjust the production over time so that it follows the demand. On the one hand the production shouldn't be too low, since this will not

[1] Note that a possible heat source for the network is industrial waste heat. In this context, the industrial complex can then be considered as the production unit.

[2] In a substation, heat is transferred from the main distribution network to a secondary stream with lower temperature and pressure.(Frederiksen and Werner, 2013)

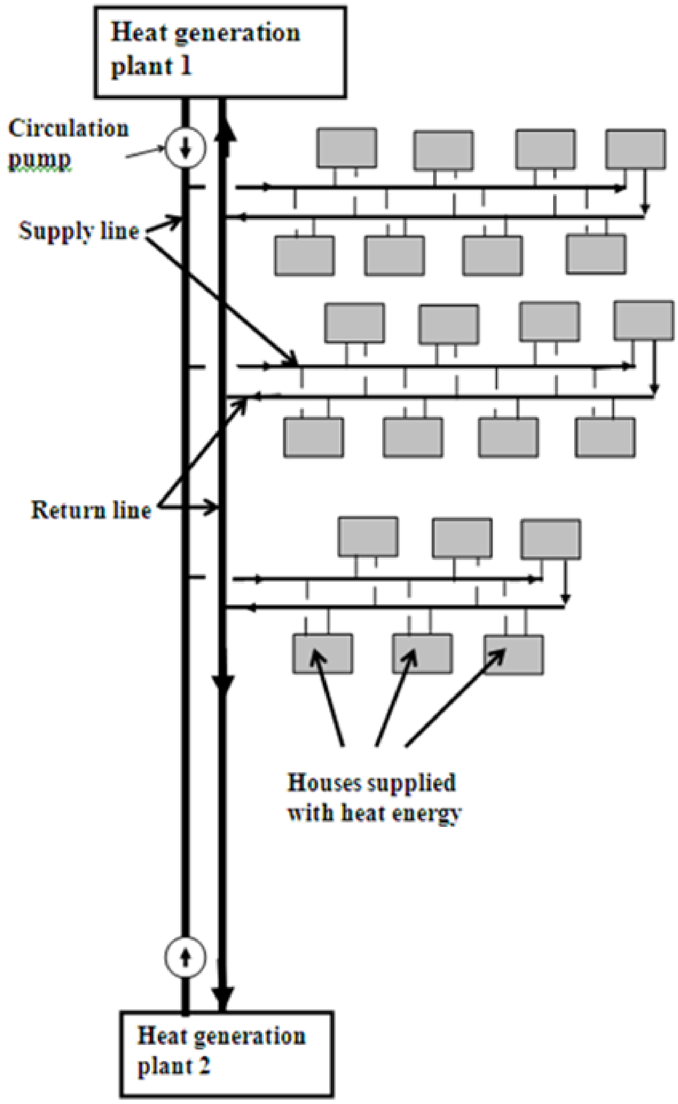


Figure 1: Simple sketch of a district heating network, displaying production sites, distribution pipes and customers.

satisfy the customers. On the other hand the production shouldn't be too high, since this leads to heat losses and excess pumping, which in turn implies increased costs for fuel and driving the pumps.

#### 2.1.4 Load Prediction

Although the typical weekly demand variations could be described as a function of time, the exact demand is naturally different for each day and week. However, methods exist for predicting the load, see e.g. (Kvarnström et al., 2007).

One method is to use regression analysis based on measured historical values in order to make a linear prediction function. A basic pattern for the prediction function is given by a model for the typical daily and weekly variation. Besides this, an

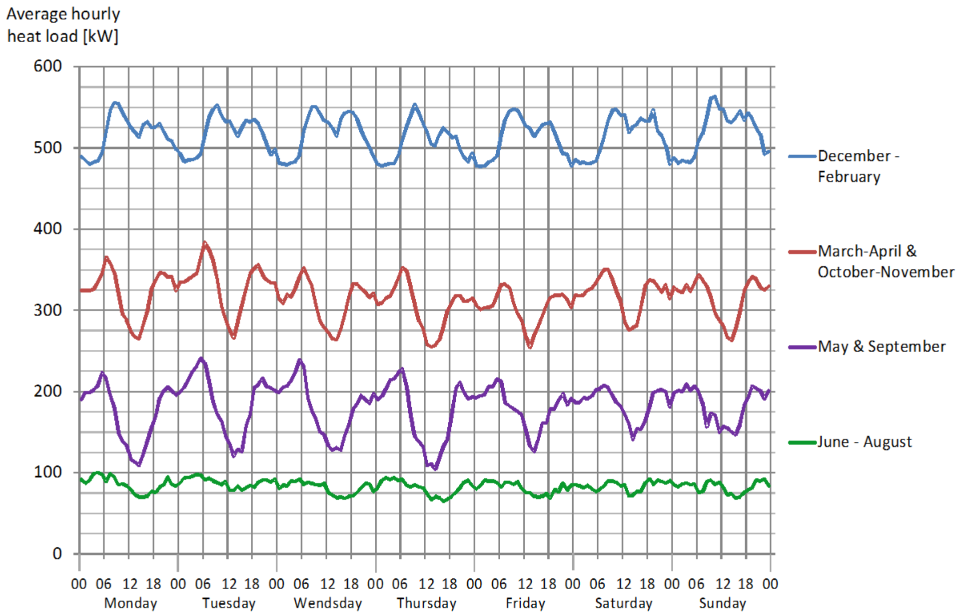


Figure 2: Weekly heat load patterns for a multi-dwelling building, according to (Gadd and Werner, 2013, p. 179). Reprinted with permission.

essential variable in the prediction function is the predicted outdoor temperature. A prediction model described in (Kvarnström et al., 2007) also models a building's ability to store heat, as well as the effect of wind on the natural ventilation.

It is hard to give a general statement on the magnitude of prediction errors. Results from a case study in (Kvarnström et al., 2007) show relative errors around 5 percent in winter and 20-25 percent in summer.[3]

### 2.1.5 Description of Uncertainties in Prediction

The uncertainties involved in predicting heat load distributions for district heating networks are investigated in (Häggståhl et al., 2004). The main source of uncertainty is the weather forecast, but other factors presented are: electricity price, quality of fuel and access to production units. Specifically, the outdoor temperature is the weather factor with greatest influence on the heat demand.

## 2.2 Optimization Models

The thesis (Dotzauer, 2001) expresses the unit commitment problem as a mixed integer programming (MIP) problem, see (Dotzauer, 2001, p. 4). The thermal loads [MW] are continuous decision variables and binary variables are introduced to describe the on/off status of a production unit. The demand constraint is set based on a deterministic demand prediction, while the heat load changes at each

[3]One explanation to the seasonal difference can be that in summer a major part of the heat load is used for hot water consumption, which is dependent on the consumer behaviour and therefore hard to predict.

time step are bounded by constant ramp constraints. An accumulator is included, giving possibility to store a certain amount of heat. However, this model has several limitations, which are discussed below.

Firstly, the model in (Dotzauer, 2001, p. 4) does not model the additional time required to increase the heat production at the start-up of a unit (and likewise at shutdown). The model in (Arroyo and Conejo, 2004) however, adds possibility to model the start-up and shutdown trajectories of the production units. This is accomplished by introducing two additional binary variables, telling for each time step if a unit is started up or shut down respectively.

A second limitation of (Dotzauer, 2001, p. 4) is that it does not include any model of the district heating network. A model without network is suitable when heat losses and delays are negligible or are accounted for exterior to the optimization model. For example, they may be accounted for in the demand prediction, so that the demand prediction specifies the amount of heat that needs to be produced. According to (Kvarnström et al., 2007, p. 21), demand prediction has traditionally been based on data from the production units, specifying previous production. However, since heat losses and delays depend on the production strategy, it would be more natural to work from demand predictions for the customers (rather than the producers). Tests in (Kvarnström et al., 2007) also show improved results for the accuracy of such predictions, compared to the traditional method. This then requires that the dynamics of delays and heat losses are included in the optimization model.

The absence of a net model in (Dotzauer, 2001, p. 4) also limits the production plan to a single production site. In contrast, it is not uncommon to have several different production sites attached to the same network (see for exampel the Uppsala district heating network (Larsson et al., 2014, p. 5)). A net model combined with customer load predictions could enable competition between different geographical production sites in the optimization.

A third limitation in (Dotzauer, 2001, p. 4) is the use of heat energy as decision variables. In reality, there are two degrees of freedom in how to produce a certain amount of heat: the mass flow $\dot{m}$ and the supply temperature $T_{out}$, see (1). In order to further develop the model it could be helpful to replace the decision variables for heat energy by variables for mass flows and supply temperatures. However, this would naturally increase the complexity of the model and therefore it is not considered in this project.

## 2.3 Robust Strategies

The formulation in (Dotzauer, 2001, p. 4) is deterministic and relies on one single prediction for expected load. Despite this, the model is robust to unexpected peaks. For power systems this is achieved through a constraint on reserve power, which gives a lower bound for the available power of the running production units (which with a margin then exceeds the expected demand). For heating systems, robustness is instead achieved by heat storage, either in a separate accumulator or in the network itself.

The possibility to store heat in a district heating network is essential, since this in fact is what enables the transport of heat to the customer. It is true that the heat stored in the network creates a buffer which can handle some amount of unexpected behaviour. Furthermore, adding a well isolated accumulator increases the opportunities for storage. However, on a longer time scale, there must be an equilibrium between heat withdrawn from the network (or the accumulator) by the customers and heat deposited from the production units. And this in turn requires that enough production units are turned on, since they may take long time to start up. Thus, for the unit commitment problem, the possibility to handle demand uncertainties by heat storage is limited.

In the last article of the thesis (Dotzauer, 2001, p. 91), a stochastic model is formulated. The model specifies a scenario tree with different scenarios for the parameters, having all properties in common except the predicted demand and the electricity price. A similar model is proposed in (Shiina and Birge, 2004) for application on power systems. These models can be solved using stochastic programming (see Section 3.1). A third example of a scenario based model is given in (Ruiz et al., 2009). This model for power systems combines stochastic programming with the mentioned idea of a reserve constraint. A parallel for heating systems could be to combine stochastic programming with accumulation modeling (as is done in Section 7.6).

Scenario based formulations such as (Dotzauer, 2001, p. 91) require generation of multiple demand scenarios. A method for predicting the *expected* demand, such as the one discussed in Section 2.1.4, is then insufficient. In contrast, a method suitable for use with stochastic programming is proposed in (Feng et al., 2015).

Another strategy for robustness is presented in (Sandou et al., 2005), based on predictive control (see Section 3.2). The control strategy is robustified by using the network's inherent ability to store heat. In other words, the supply temperature is set with a margin, so that the network can supply demands in spite of model uncertainties and load prediction errors. However, this study does not include constraints for the start-up and shutdown trajectories of production units. Furthermore, it only considers one scenario for the demand prediction. Thus, this strategy will always plan for the expected demand prediction. Since some production units have start-up times of many hours, it is important to in advance consider different scenarios, which may require different sets of production units on-line.

## 2.4 Tools

The optimization model explained in Section 4.1 is implemented using the open source software package Pyomo - Python Optimization Modeling Objects, see online documentation (PyomoDoc) and book (Hart et al., 2012). The package is embedded in the Python programming language, so that the Pyomo objects are Python objects. For the stochastic programming formulation in Section 4.2, the PySP (Pyomo Stochastic Programming) package was used. Pyomo is a so called algebraic modelling language (AML), which is a high lever computer programming language, in which mathematical optimization problems can be formulated similarly to the classical mathematical notation, with sets, indices, constraints, etc. (AMLWiki) For a list of researchers and software projects using Pyomo, see the Pyomo web page (PyomoWeb).

Pyomo, like other AML:s, does not solve the problem itself, but calls an external solver for this purpose. This project uses the commercial solver Gurobi, which has algorithms for solving mixed integer programs, linear as well as quadratic. (GurobiHome) Mixed integer linear programming (MILP) problems are generally solved using the branch-and-bound algorithm. According to the Gurobi web site (GurobiMIP), the algorithm for solving Mixed Integer Quadratically Constrained Problems is similar.

# 3 Theory

This section gives a brief introduction to stochastic programming and model predictive control.

## 3.1 Stochastic Programming

The basic idea of stochastic programming is to solve optimization problems where some parameters are uncertain. This is illustrated graphically in Figure 3 and explained in the subsection 3.1.1. This is followed in subsection 3.1.2 by a practical example called the News Vendor problem. Finally, subsection 3.1.3 then gives a more general formulation of a (linear) two stage stochastic programming problem.

### 3.1.1 Simple Illustration

Consider a minimization problem where the objective function is given as the quadratic function $f_1$ in Figure 3. The solution is marked in the figure by the dashed line. Now assume that the objective function is instead dependent on a stochastic variable. With probability $p_1 = 0.6$, the objective function will be $f_1$, but there is also a probability $p_2 = 0.4$ that the objective function is $f_2$, a second quadratic function plotted in the second subplot of Figure 3. We see that the solution to $\min_x f_1(x)$ is a poor solution to $\min_x f_2(x)$. The idea of stochastic programming applied a problem with uncertain objective function is to find $x$ that minimizes the expected value of the objective function:

$$\min_x \Big(\mathbf{E}[f(x)]\Big) = \min_x \Big(p_1 \cdot f_1(x) + p_2 \cdot f_2(x)\Big). \tag{2}$$

This is illustrated by the red curve and the red dashed line in the bottom subplot of Figure 3.

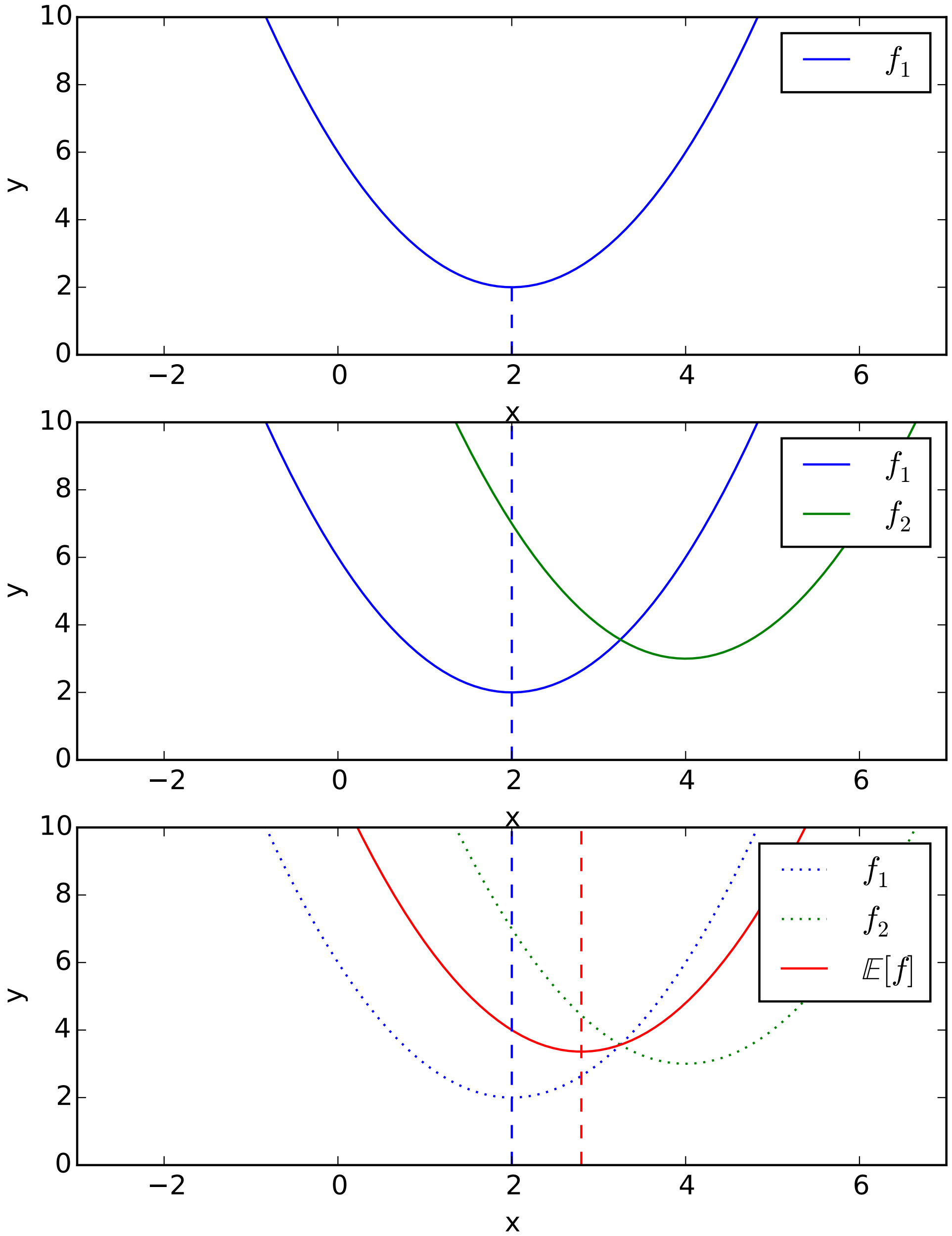


Figure 3: Illustration of stochastic programming. The top subplot simply shows the solution of minimizing a (deterministic) quadratic function. The middle subplot instead illustrates a case with a stochastic objective function $F$, which has the two possible outcomes $f_1$ and $f_2$. It is clear that the optimal $x$ is dependent on the outcome of $F$. The bottom subplot adds a plot in red of the expected values $\mathbf{E}[f(x)]$ of the objective function, for different $x$. The solution to $\min_x \mathbf{E}[f(x)]$ is marked by the red dashed line.

### 3.1.2 The News Vendor Problem

A simple example of a stochastic programming problem is the news vendor problem. (Shapiro et al., 2009, p. 1) Suppose a news vendor wants to maximize his profit from selling news papers. At Day 1 he can buy $n_1$ newspapers for the price $p_1$ SEK per newspaper. However, the demand does not become known until Day 2. He also has the opportunity to buy $n_2$ extra newspapers on Day 2, but for the higher price $p_2$. Newspapers are sold on the market with price $p_m$, while leftover papers can be sold back for the lower price $p_r$ ($r$ as in return). Now, the question is: how many newspapers should be bought at Day 1?

Assuming the newspaper demand is a stochastic variable $D$ with known probability distribution, the optimization problem to be solved is

$$\max_{n_1} \quad \mathbf{E}[P(n_1, D)] - p_1 n_1 \tag{3}$$

where

$$\begin{aligned} P(n_1, D) = \max_{n_2, n_m, n_r} \quad & p_m n_m + p_r n_r - p_2 n_2 \\ \text{s.t.} \quad & n_m \geq D \\ & n_m + n_r \leq n_1 + n_2 \end{aligned} \tag{4}$$

This example in fact illustrates the essence of the production planning problem: how should the production be planned in advance while the demand is still unknown. If the plan turns out to be inaccurate, there may be ways to still cover the demand, but in turn the costs will increase.

### 3.1.3 Two Stage Problem

The News Vendor Problem is an example of a two stage stochastic programming problem. In such a problem, the optimization variables are divided into two stages. The Stage 1 variables ($n_1$ in the News Vendor problem) have to be set before the realization of stochastic parameters, while the Stage 2 variables ($n_2$, $n_m$ and $n_r$ in the News Vendor problem) are set based on knowing the realizations of stochastic parameters, as well as the choices of the Stage 1 variables. A general two stage stochastic linear programming problem can according to (Shapiro et al., 2009, p. 27) be written as

$$\begin{aligned} \min_{x} \quad & c^T x + \mathbf{E}[C(x, \xi)] \\ \text{s.t.} \quad & Ax \leq b \end{aligned} \tag{5}$$

where $C(x, \xi)$ is the minimized cost in to the second stage problem

$$\begin{aligned} C(x, \xi) = \min_{y} \quad & q^T y \\ \text{s.t.} \quad & Tx + Wy \leq h. \end{aligned} \tag{6}$$

and $\xi = (q, T, W, h)$ refers to the second stage data, of which some or all can be considered random. The distributions of the random data are taken into account when calculating the distribution of $C(x, \xi)$, which can also be seen as a stochastic variable. In this thesis the random data has a finitely supported distribution, and a specific realization of the data is called a *scenario*.

Note that this section gives a formulation of a two stage stochastic *linear* programming problem, while we in this project will consider a MIQCP problem formulation (see Section 4.1). However, the stochastic programming ideas are the same in both cases. For a longer introduction to stochastic programming, see (Shapiro et al., 2009) or (Birge and Louveaux, 2011).

## 3.2 Model Predictive Control

Model predictive control is an iterative control method. It uses a model to predict the output signals of the system, over a finite time horizon, given a certain input signal to the system. Based on the predictions, the 'optimal' input signal is chosen, e.g. the signal which is predicted to give lowest cost, or the signal that is predicted to best fulfil the requirements of the system. This input signal is then applied for only one time step, where after the time horizon is moved forward one step and the procedure is repeated. More details can be found in (Maciejowski, 2002).

# 4 Models for Optimization and Scenario Construction

## 4.1 Optimization Model

The unit commitment problem considered in this project can be classified as a Mixed Integer Quadratically Constrained Programming (MIQCP) problem. (GurobiMIP) Integer programming refers to some of the optimization variables being integers and in this case all the integer variables are binary variables, taking only values zero or one. "Quadratically Constrained" means that the problem contains products of variables in both the objective function and in one or more of the constraints (in this case only the demand constraint). The following subsections present the decision variables, the objective functions, the constraints and the initial conditions.

### 4.1.1 Decision Variables

In principle, the problem is to set the amount of heat to be produced, at each time step and for each production unit. On top of this, in case an accumulator is used, at each time step we also have to decide how much should be taken from or added to the accumulator. This information can be summarized in three indexed decision variables:

- heat production ($q_{i,k}$)
- production unit on/off variables ($u_{i,k}$)
- heat transferred from accumulator ($q_{i,S}$)

where $i$ is a time step index and $k$ specifies the production unit. The on/off variables $u_{i,k}$ are binary variables, such that

$$u_{i,k} = \begin{cases} 1, & \text{if at time } i \text{ production unit } k \text{ is turned on} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

However, in order to formulate startup and shutdown conditions, two further binary variables are used:

- startup variables ($y_{i,k}$)
- shutdown variables ($z_{i,k}$).

These are constrained so that

$$y_{i,k} = \begin{cases} 1, & \text{if a startup process starts at time } i \text{ for production unit } k \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

and

$$z_{i,k} = \begin{cases} 1, & \text{if a shutdown process finishes at time} i \text{ for production unit } k \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Thus, variables $y_{i,k}$ and $z_{i,k}$ are directly determined by knowing $u_{i,k}$ and $u_{i-1,k}$. Finally, one variable is also included to denote of the accumulated heat:

- accumulator storage ($e_{i,S}$).

Note that $e_{i,S}$ is directly determined by knowing the transfers $q_{i,S}$ together with the initial accumulator storage. Adding up, there are in total six indexed optimization variables, but the three latter ($y_{i,k}$, $z_{i,k}$ and $e_{i,S}$) can be determined from the three first.[4]

### 4.1.2 Objective Function

The objective function can be written as

$$f(u_{i,k}, q_{i,k}, y_{i,k}, z_{i,k}) = \sum_{k \in K} \Big( \sum_{i=1}^{I} \big( (c_k^f + c_k^v) u_{i,k} q_{i,k} + c_k^{fo} \cdot u_{i,k} + C_k^{up} \cdot y_{i,k} \big) + \sum_{i=1}^{I+DD_k} C_k^{down} \cdot z_{i,k} \Big) \tag{10}$$

where

- $c_k^f$ is the fuel cost
- $c_k^v$ is a cost proportional to produced heat
- $c_k^{fo}$ is a fixed operation cost
- $C_k^{up}$ is a startup cost
- $C_k^{down}$ is a shutdown cost

and where $K$ is the set of production units. Revenues from sold heat are excluded from the objective function, since it is assumed that the amount of sold heat always is equal to the heat demanded, and therefore the revenues will be independent of the solution.

### 4.1.3 Demand Constraint

The demand constraint is perhaps the most essential constraint as it requires the produced energy to cover the customer needs. Most of the tests in this project do not include a net model and assume that the predicted demand $q_{i,D}$ for time step $i$ specifies directly the amount of heat needed to be produced at each hour. In this case, the demand constraint is that

$$\sum_{k \in K} u_{i,k} q_{i,k} + q_{i,S} \geq q_{i,D}, \ \forall i \in [1, I]. \tag{11}$$

However, in Test 4, a simple test is done using a net model. This is especially useful if the demand prediction specifies the demand of individual customers (see discussion in Section 2.2). Based on these predictions and the net model, an alternative demand constraint can be that

$$\sum_{k \in K} u_{i,k} q_{i,k} + q_{i,S} \geq \sum_{c \in C} q_{i+\tau_c, D_c}, \ \forall i \in [1, I]. \tag{12}$$

[4]The Pyomo implementation adds a variable that stores the cost of each stage. This is required for the use of PySP.

Here, $c$ is a customer index, $q_{i,D_c}$ is the predicted demand of customer $c$ at time $i$ and $\tau_c$ is the static delay from the production location to customer $c$. It is assumed that all production units are located at the same production site, in order for the customer delays to be independent on production unit. Furthermore, the delays are assumed to be static. In Sections 4.2 and 4.3, the demand constraint (11) is the one included, but for Test 4 this is exchanged by contraint (12).

#### 4.1.4 Accumulator Constraints

The accumulator is modelled by three contraints:

$$e_{i,S} = (1-\alpha)(e_{i-1,S} - q_{i,S}) \tag{13}$$

$$\underline{e}_S \leq e_{i,S} \leq \bar{e}_S \tag{14}$$

$$\underline{Q}_S \leq q_{i,S} \leq \bar{Q}_S \tag{15}$$

Here

- $\alpha$ is a static loss coefficient
- $\underline{e}_S$ is the minimal accumulator storage
- $\bar{e}_S$ is the maximal accumulator storage
- $\underline{Q}_S$ and $\bar{Q}_S$ limits the amount of heat transferred to and from the accumulator during one time step.

Additionally, the accumulator storage at the end of the optimization horizon is required to be greater or equal than the initial storage:

$$e_{I,S} - q_{I,S} \geq e_{0,S}. \tag{16}$$

#### 4.1.5 Startup Constraints

The start-up constraints are taken from (Arroyo and Conejo, 2004) (constraints numbered (1) - (12) in the article). In order to interpret the start-up constraints it is helpful to note that

$$\sum_{m=1}^{\mathrm{UD}_k} y_{i-m+1,k} = \begin{cases} 1 & \text{during the hours of a start-up period} \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

where $\mathrm{UD}_k$ is the startup duration of production unit $k$. Likewise,

$$\sum_{m=1}^{\mathrm{DD}_k} z_{i+m,k} = \begin{cases} 1 & \text{during the hours of a shutdown period} \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

where $\mathrm{DD}_k$ is the shutdown duration of production unit $k$.

The start-up constraints are now formulated as (see interpretations on next page):

$$q_{i,k} \geq \underline{Q}_k \left[ u_{i,k} - \sum_{m=1}^{\mathrm{DD}_k} z_{i+m,k} - \sum_{m=1}^{\mathrm{UD}_k} y_{i-m+1,k} \right] + \sum_{m=1}^{\mathrm{UD}_k} Q_k^U(m) y_{i-m+1,k} \tag{19}$$

$$q_{i,k} \geq \underline{Q}_k \left[ u_{i,k} - \sum_{m=1}^{\mathrm{DD}_k} z_{i+m,k} - \sum_{m=1}^{\mathrm{UD}_k} y_{i-m+1,k} \right] + \sum_{m=1}^{\mathrm{DD}_k} Q_k^D(m) z_{(i+\mathrm{DD}_k-m+1),k} \tag{20}$$

$$q_{i,k} \leq \sum_{m=1}^{\mathrm{UD}_k} Q_k^U(m) y_{i-1+1,k} + \bar{Q}_k \left[ u_{i,k} - \sum_{m=1}^{\mathrm{UD}_k} y_{i-m+1,k} \right] \tag{21}$$

$$q_{i,k} \leq \sum_{m=1}^{\mathrm{DD}_k} Q_k^D(m) z_{(i+\mathrm{DD}_k-m+1),k} + \bar{Q}_k \left[ u_{i,k} - \sum_{m=1}^{\mathrm{DD}_k} z_{i+m,k} \right] \tag{22}$$

$$q_{i,k} - q_{i-1,k} \leq \bar{Q}_k \sum_{m=1}^{\mathrm{UD}_k} y_{i-m+1,k} + \mathrm{RU}_k \left[ u_{i,k} - \sum_{m=1}^{\mathrm{UD}_k} y_{i-m+1,k} \right] \tag{23}$$

$$q_{i,k} - q_{i-1,k} \leq \bar{Q}_k \sum_{m=1}^{\mathrm{DD}_k} z_{i+m-1,k} + \mathrm{RD}_k \left[ u_{i,k} - \sum_{m=1}^{\mathrm{DD}_k} z_{i+m-1,k} \right] \tag{24}$$

$$y_{i,k} - z_{i,k} = u_{i,k} - u_{i-1,k} \tag{25}$$

$$u_{i,k} \geq \sum_{m=1}^{\mathrm{UD}_k} y_{i-m+1,k} \tag{26}$$

$$u_{i,k} \geq \sum_{m=1}^{\mathrm{DD}_k} z_{i+m,k} \tag{27}$$

$$y_{i,k} + \sum_{m=1}^{\mathrm{UD}_k+\mathrm{DD}_k-1} z_{i+m-1,k} \leq 1 \tag{28}$$

$$q_{i,k} \geq Q_k^U(\mathrm{UD}_k) \left[ \sum_{m=1}^{\mathrm{DD}_k} z_{i+m,k} + \sum_{m=1}^{\mathrm{UD}_k} y_{i-m+1,k} - 1 \right] \tag{29}$$

$$q_{i,k} \geq Q_k^D(1) \left[ \sum_{m=1}^{\mathrm{DD}_k} z_{i+m,k} + \sum_{m=1}^{\mathrm{UD}_k} y_{i-m+1,k} - 1 \right]. \tag{30}$$

Here

- $\bar{Q}_k$ is the maximal production for unit $k$

- $\underline{Q}_k$ is the minimal production for unit $k$

- $Q_k^U(m)$ is the production of unit $k$ at the $m$th step up a start-up

- $Q_k^D(m)$ is the production of unit $k$ at the $m$th step up a shutdown

- $\mathrm{RD}_k$ is the ramp-down limit for production unit $k$ and

- $\mathrm{RU}_k$ is the ramp-up limit for production unit $k$.

These constraints hold for all $i \in [1, I]$ and $k \in K$. Note that in order for the constraints to be well defined for all $i \in [1, I]$, $y_{i,k}$ and $z_{i,k}$ need to be defined for all $i$ in the range $[1 - UD_k, I + DD_k]$. In the implementation this is accomplished by letting all decision variables[5] be defined on the interval

$$[1 - UD_{max}, I + DD_{max}]. \tag{31}$$

where

$$UD_{max} = \max_k UD_k \tag{32}$$

and

$$DD_{max} = \max_k DD_k. \tag{33}$$

Here follow brief explanations of the startup constraints, for details, see (Arroyo and Conejo, 2004):

- constraints 19 and 20 give lower limits during startup and shutdown of production unit $k$, respectively
- constraints 21 and 22 give upper limits during startup and shutdown of production unit $k$, respectively
- constraint 23 gives a ramp up limit when the production unit is not being started up
- constraint 24 gives a ramp down limit when the production is not being shut down
- constraints 25, 26 and 27 set the relationship between variables $u$, $y$ and $z$
- constraints 28, 29 and 30 sets limits for the peculiar case when a startup process and a shutdown process are overlapping

In the tests, the start-up and shutdown trajectories $Q_k^U$ and $Q_k^U$ are assumed to be linear, but they could be adjusted to fit the characteristics of a particular production unit.

[5] Thus the implementation will generally include decision variables that are meaningless, such as e.g. $q_{-2,k}$ denoting the heat produced by production unit $k$ at time $-2$. However, these contribute neither to fulfilling the demand constraint 11 nor to the objective function 10 and thus they do not affect the solution.

#### 4.1.6 Initial Conditions

Finally, the initial conditions specify the operation of the production units at time $t = 0$, as well as the initial state of the accumulator. Marking the initial conditions with hats (e.g. $\hat{q}_k$), we have that

$$\begin{aligned} q_{0,k} &= \hat{q}_k & & \\ u_{0,k} &= \hat{u}_k & & \\ y_{i,k} &= \hat{y}_{i,k}, & i &\in [1-UD, 0] \\ z_{i,k} &= \hat{z}_{i,k}, & i &\in [I+1, I+DD] \\ e_{0,S} &= \hat{e}_S & & \\ q_{0,S} &= \hat{q}_S. & & \end{aligned} \tag{34}$$

### 4.2 Stochastic Programming

Now we will formulate the unit commitment problem as a two stage stochastic programming problem. To enable a compact notation, let $x_1$ and $x_2$ be vectors containing all the decision variables in Stage 1 and Stage 2 respectively. If we let $I_1$ denote the end time of Stage 1, this then means that $x_1$ contains all decision variables with time index $i \leq I_1$ and $x_2$ contains all decision variables with time index $i > I_1$. See the Appendix for an explicit definition.

In addition, let

$$MC = \{x|(13) - (15), (19) - (30)\} \tag{35}$$

be the set of decision variables that fulfil the model constraints (except the end time accumulation constraint (16)) and startup constraints. Also, let

$$IC = \{x|(34)\} \tag{36}$$

be the set of decision variables that fulfil the initial conditions.

#### 4.2.1 Two Stage Formulation

A two stage stochastic programming problem can then be written as

$$\min_{x_1} \quad \sum_{i=1}^{I_1} \sum_{k \in K} f(x_1) + \mathbf{E}[C(x_1, \xi)] \tag{37}$$

$$\text{s.t.} \quad \sum_{k \in K} u_{i,k} q_{i,k} + q_{i,S} \geq q_{i,D}, \qquad i \in [1, I_1] \tag{38}$$

$$x_1 \in MC \tag{39}$$

$$x_1 \in IC \tag{40}$$

where $C(x_1, \xi)$ is the solution to the second stage problem

$$
\begin{aligned}
C(x_1, \xi) = \min_{x_2} \quad & \sum_{i=I_1+1}^{I_2} \sum_{k \in K} f(x_2) && && (41)\\
\text{s.t.} \quad & \sum_{k \in K} u_{i,k} q_{i,k} + q_{i,S} \geq q_{i,D}(\xi), && i \in [I_1+1, I_2] && (42)\\
& e_{I_2,S} - q_{I_2,S} \geq \hat{e}_S && && (43)\\
& x_2 \in MC. && && (44)\\
& && && (45)
\end{aligned}
$$

Here $\xi$ is the outcome of the second stage demand and the notation $q_{i,D}(\xi)$ marks the stochastic variables. Let $\Xi$ denote the probability space for the outcome $\xi$, consisting of a finite number of scenarios for the demand.

### 4.2.2 Dependency Between Stages

It is important to note that the constraints in Stage 2 depend implicitly on the solutions for Stage 1, $x_1$, since the following initial values of Stage 2 are decided in Stage 1:

$$
\begin{aligned}
& q_{I_1,k} \\
& u_{I_1,k} \\
& y_{i,k}, \quad i \in [I_1 + 1 - UD, I] \\
& z_{i,k}, \quad i \in [I_1 + 1, I_1 + DD] \\
& e_{I_1,S} \\
& q_{I_1,S}.
\end{aligned}
\tag{46}
$$

An interesting decision is the choice of the parameter $I_1$, which describes at what time index the decision variables are separated between stages 1 and 2. In the case net model is used, so that constraints (38) and (42) are modified in accordance to (12), there is a time delay in the demand constraint. In that case the index $I_1$ must be chosen such that the indexed demand $q_{i,D_c}$ can be assumed to be certain for all $i$ up to $i = I_1 + \max_c \tau_c$.

## 4.3 Worst Case Optimization

In order to have a reference for evaluation of the stochastic programming results, this project will also consider a worst case approach.[6] The idea for this is simple: optimize stage one based on the scenario with maximum demand in stage two and

[6]Notice that this definition differs slightly from what's called worst case approach in (Shapiro et al., 2009, p. 4). Instead of defining the worst case scenario by maximum cost, it is defined by maximum demand. However, as $\xi$ is only present in the demand constraint (42), we see directly from (48) that the optimization variable space for the worst case scenario $\xi_{max}$ is a subspace of the variable space for any other scenario. This in turn means that the worst case scenario will have the maximal cost, and thus for the chosen scenarios this definition coincides with (Shapiro et al., 2009).

then reoptimize when the outcome of demand in stage two becomes available (c.f. the 'Wait and See'-method in (Larsson et al., 2014, p. 40)). More precisely, the worst case optmization solves the problem

$$\begin{aligned} \min_{x_1} \quad & \sum_{i=1}^{I_1} \sum_{k \in K} f(x_1) + C(x_1, \xi_{\max}) \\ \text{s.t.} \quad & \sum_{k \in K} u_{i,k} q_{i,k} + q_{i,S} \geq q_{i,D}, \qquad i \in [1, I_1] \\ & x_1 \in MC \\ & x_1 \in IC \end{aligned} \tag{47}$$

where $q_{i,D}(\xi_{\max})$ is constructed so that

$$q_{i,D}(\xi_{\max}) \geq q_{i,D}(\xi), \ \ \forall \xi, \ \ \forall i \in [I_1 + 1, I_2] \tag{48}$$

and the second stage problem is the same as for stochastic programming (see previous section). To clarify, the only difference between the stochastic programming formulation and the worst case approach is in the objective function, where $\mathbf{E}[C(x_1, \xi)]$ is substituted by $C(x_1, \xi_{\max})$.

## 4.4 Scenario Construction

The formulations in the two previous sections both rely on a discrete set of scenarios for the demand prediction. As mentioned in Section 2.3, methods have been proposed for the generation of such scenarios. However, this project focuses not on the generation of realistic scenarios but rather on the evaluation of different robust strategies. Therefore, using methods such as those proposed in (Feng et al., 2015) is considered outside the scope of the project. Instead scenarios are constructed based on an intuitive demand probability distribution, described in the next subsection.

Once a probability distribution has been set, discrete scenarios need to be generated. One goal for these scenarios is that they summarize possible outcomes of the heat demand. In particular, the scenarios should cover outcomes with high demands, since these are the outcomes crucial to robustness.

In the tests, concrete scenarios are chosen so that at each point they go through a value corresponding to a particular percentile of the normal distribution. For example, one scenario follows the 25-percentile, meaning that according to the model it exceeds the actual demand with probability 25 percent.

The described method for scenario construction is simple and straightforward. However, it could be argued that it gives a poor representation of the demand probability space, as all scenarios have the same shape. An alternative strategy is to use Monte-Carlo methods, as described in (Boyd, 2014).

## 4.5 Stochastic Demand Model

The demand is split in two periods, where the first period's demand is assumed to be known, while the demand in period two is unknown. Call the time index of the last known demand hour $i = I_a$, so that period one is

$$[1, I_a] \tag{49}$$

and period two is

$$[I_a + 1, I_b]. \tag{50}$$

As explained in Section 4.2, when the demand constraint with net model (12) is used, it is necessary that

$$I_a \geq I_1 + \max_c \tau_c. \tag{51}$$

On the contrary, when the demand constraint without net model (11) is used, it is natural to let the known and unknown period correspond to Stage 1 and Stage 2 respectively, so that

$$I_a = I_1 \tag{52}$$

and

$$I_b = I_2. \tag{53}$$

This section models a probability distribution for the demand in the the second period.

Model the total customer heat demand as a stochastic process $q_{i,D}(\xi)$, where $i \in I_b$ is the time index. As mentioned in Section 2.1.2, the demand can be divided into different parts with different properties. Neglecting the losses, the total demand can be written as a sum of stochastic variables $q_{i,b}(\xi)$ and $q_{i,C}(\xi)$

$$q_{i,D}(\xi) = q_{i,b}(\xi) + q_{i,C}(\xi). \tag{54}$$

where $q_{i,b}(\xi))$ is the amount of heat used for building heating and $q_{i,C}(\xi)$ gives the heat demand for water heating. Now, assume that the realization of $q_{i,b}(\xi))$ is mainly dependent on the weather. Since the time intervals considered are short compared to the rate of weather changes, the distributions of $q_{i,b}(\xi)$ should not be independent for different $i$. Rather, assume that the deviation from the expected curve $q^*_{i,b} = \mathbf{E}[q_{i,b}(\xi)]$ is a linear function of time, so that

$$q_{i,b}(\xi) = q_{I_a,b} + r(\xi) \cdot (i - I_a), \qquad\qquad i > I_a \tag{55}$$

where

$$r(\xi) \sim \mathcal{N}(0, c_1 \cdot q_{I_a,b}). \tag{56}$$

Furthermore assume that $q_{i,C}$ are independent random variables with normal distribution and standard deviation proportional to its expected value,

$$q_{i,C}(\xi) \sim \mathcal{N}(q^*_{i,C}, c_2 \cdot q^*_{i,C}) \tag{57}$$

where $q^*_{i,C} = \mathbf{E}[q_{i,C}(\xi)]$ and $c_2$ is a constant.

# 5 Implementation Using Pyomo

This section gives a brief explanation of how the implementation has been performed. As explained in Section 2.4, the model is written in the Pyomo modelling language. The test scripts are then written in Python.[7]

## 5.1 Model Formulation With Pyomo

The optimization model of section 4.1 has been implemented as a Pyomo AbstractModel object. This means that the parameter data is specified in a data file outside the model object, which is then sent as input when an instance of the model is created. As mentioned in Section 2.4, the implementation is mostly straight forward and follows the model in Section 4.1. Examples of Pyomo code are given below for definition of a set, a parameter, a double indexed variable and an initial condition.

```
# Define the set of production units
model.setOfPU = Set()

# Define the time of the first planning hour
model.tStart = Param(within= Integers)

# Define the on/off variable u
model.u = Var(model.setOfHExtended, model.setOfPU, domain=Binary)

# Initialize the amount of heat stored in the accumulator
def ac_init_constraint_rule(model):
    return model.storAc[model.tStart - 1] == model.acInit

model.acInitConstraint = Constraint(rule = ac_init_constraint_rule)
```

## 5.2 Worst case Optimization

The worst case optimization is in Tests 1, 2 and 4 is performed by a Python script that first solves the Stage 1 problem, which in this case is an ordinary MIQCP problem. Thereafter, the Stage 2 problem is solved for each scenario, with initial conditions decided by the Stage 1 solution (see explanation in the following section). For the MPC tests, Test 3 and Test 5, there is no need to solve a second stage problem, since the length of Stage 1 is set to be equal to the iteration period.

### 5.2.1 Setting Stage 2 Initial Conditions

At the start of stage two, the states of the production units and the accumulator need to be specified. As is explained in Section (4.2), the initial values of Stage 2 need to be set according to the solution in stage 1. In order to explain this a bit further, the conditions are repeated and numbered here:

[7] The programming code is stored by Modelon AB.

$$q_{I_1,k} = q^*_{I_1,k} \tag{58}$$
$$u_{I_1,k} = u^*_{I_1,k} \tag{59}$$
$$y_{i,k} = y^*_{i,k}, \qquad i \in [I_1 + 1 - UD, I_1] \tag{60}$$
$$z_{i,k} = z^*_{i,k}, \qquad i \in [I_1 + 1, I_1 + DD] \tag{61}$$
$$e_{I_1,S} = e^*_{I_1,S} \tag{62}$$

where $i = I_1$ is the last time index of stage 1.

Thus, in addition to knowing which production units are turned on (59) and the heat they produce (58) it is also necessary to know which of them are in start-up and shutdown mode. This is set by (60) and (61) respectively. Finally, it is required to specify the initial heat storage (62).

The method for setting the initial conditions of Stage 2 is simple: read the variables in (34) from the previous iteration and enter them into the constraints.

## 5.3 Stochastic Programming with PySP

The stochastic programming formulation with PySP requires two types of data files:

- one data file for each scenario (or alternatively each node in the scenario tree)
- one file specifying the scenario structure

Besides this the Pyomo model needs to be modified so that it contains variables for the costs in each stage, as mentioned in Section 4.1. The data files are specified in the same way as for deterministic problems.

### 5.3.1 Specifying the Scenario Structure

Once the scenarios have been fixed, they can be organized in a tree structure. The organization is such that each stage contains a number of nodes. The number of nodes in the first stage is one, since all scenarios are assumed to have the same properties in stage one. The number of nodes in the last stage on the other hand (Stage 2 in this case), is equal to the number of scenarios.

### 5.3.2 Solving the Problem

The Pyomo extension PySP offers two methods to solve stochastic programming problems: extended form and progressive hedging. (PyomoDoc) This project uses the extended form method, which means to extend the deterministic problem formulation, into one that contains separate variables and constraints for each of the scenarios. If the original problem is a MILP-problem, this results in a larger scale MILP-problem. (Hart et al., 2012, p. 138)

## 5.4 Reinitializing for MPC

Section 5.2.1 explains a method to set the initial conditions for Stage 2, when doing worst case optimization for a two stage problem. Likewise, each iteration of the MPC needs to specify initial conditions, regardless of whether SP or WC is applied.

For iterated worst case optimization, the straight forward method of Section 5.2.1 can be reused: simply read the solution from the previous iteration and use it to set the initial conditions for the current iteration. When using stochastic programming however, this method was not implemented, since the default summary of the PySP results[8] didn't print out all the neccesary values of startup variables $y_{i,k}$. Therefore, three different cases are considered to set the initial conditions in (60)[9], that is $y_{i,k}$ for $i \in [I_1 + 1 - UD, I_1]$:

1. if
$$\begin{cases} u_{I_1,k} = 1 \\ u_{I_1-1,k} = 0 \end{cases} \tag{63}$$
then
$$y_{i,k} = \begin{cases} 1, & \text{for } i = I_1 \\ 0, & \text{otherwise} \end{cases} \tag{64}$$

2. if
$$\begin{cases} u_{I_1,k} = 1 \\ u_{I_1-1,k} = 1 \\ q_{I_1,k} \leq \underline{Q}_k \\ q_{I_1-1,k} < \underline{Q}_k \end{cases} \tag{65}$$
then
$$y_{i,k} = \begin{cases} 1, & \text{for } i = I_1 - (\frac{q_{I_1,k}}{\underline{Q}_k} \cdot UD) \\ 0, & \text{otherwise} \end{cases} \tag{66}$$

3. otherwise
$$y_{i,k} = 0, \forall\, i \tag{67}$$

Note that this implementation utilizes the linear growing production during startup, which is mentioned at the end of Section 4.1.5.

[8]To print the PySP solution results, the csvsolutionwriter was used, mentioned in (PyomoDoc).

[9]Note that this implementation does not ensure that all $y_{i,k}$ are the same as in the previous iteration, but what it does accomplish is to decide whether or not each production unit is in startup mode at time $i = I_1 + 1$.

# 6 Test Set Up

The tests are designed to represent circumstances where stochastic programming can have advantages to the worst case optimization described in Section 4.3. As the stochastic programming is formulated, the customer demand will be met under any circumstances. The same holds for the worst case scenario optimization. What may differ between the strategies is how the heat is produced and what are the resulting costs.

## 6.1 Production Unit Parameters

Tests have been performed based on a network with one production site consisting of three different production units: a base load unit ('Base'), a fossil fueled top up unit ('Fossil') and a solid fuel boiler top up unit ('SFB'). The characteristics of the production units are shown in Table 2. The table indicates that two different combinations of maximal production limits are considered for the Base unit and the Fossil unit. Call these PU Case 1 and PU Case 2. Letting the index of the Base Unit be $k = 1$ and the index of the Fossil Unit $k = 3$, PU Case 1 means that

$$\begin{cases} \bar{Q}_1 = 240 \\ \bar{Q}_3 = 50 \end{cases} \tag{68}$$

while PU Case 2 means that

$$\begin{cases} \bar{Q}_1 = 210 \\ \bar{Q}_3 = 80. \end{cases} \tag{69}$$

The main idea behind these two cases is to do tests without accumulation using PU Case 1 and tests with accumulation using PU Case 2. Otherwise, the choice of parameters has two main goals:

1. to have reasonable parameters

2. to get interesting test results

The following two paragraphs motivate how each of these goals have been considered.

Many of the specified parameters have been taken directly from (Larsson et al., 2014), where similar tests of using stochastic programming are performed. In order to work with prices in SEK, rather than normalized prices, an electricity price of 300 SEK is assumed. The relationships between startup/shutdown costs and fuel costs have been chosen considering data for the Uppsala district heating network studied in (Larsson et al., 2014), as well as considering the ratio between parameters F, B and C in (Arroyo and Conejo, 2004). The fuel cost for the solid fuel boiler has been set according to the price for densified wood fuels in (Energimyndigheten, 2015).[10] The efficiency has been set as $\eta = 0.9$ for all three units.

[10]A tax of 20 percent has been added.

In order to have interesting test results, it is necessary to have a set of production units that allows for different strategies of covering the demand. The Fossil unit is fast and cheap to start-up but has high fuel costs. The SFB on the other hand is slow and expensive to start-up but has low running costs. Thus the combination of the Fossil unit and the SFB, gives two different strategies to cover the demand peaks. The use of the two different cases, PU Case 1 and PU Case 2, is motivated by how the different cases give different results.

| **Parameter** | **Base unit** | **Fossil** | **SFB Unit** | |
|---|---|---|---|---|
| Maximal production | 240 (210) | 50 (80) | 120 | MW |
| Minimal production | 130 | 15 | 15 | MW |
| Ramp up limit | 50 | 65 | 50 | MW/h |
| Ramp down limit | 50 | 65 | 50 | MW/h |
| Duration of startup process | 10 | 3 | 7 | h |
| Duration of shutdown process | 10 | 2 | 2 | h |
| Fuel cost | 156 | 393 | 300 | SEK/MWh |
| Startup cost | 45000 | 9000 | 30000 | SEK/startup |
| Shutdown cost | 21000 | 3000 | 15000 | SEK/shutdown |
| Fixed maintenance cost | 3180 | 606 | 606 | SEK/h |
| Variable maintenance cost | 6.3 | 4.5 | 5.1 | SEK/MWh |

Table 2: Characteristics of the three production units. In Tests 1-4, the Base Unit has maximal production 240 MW and the Fossil unit has maximal production 50 MW, while in Test 5 the maximal productions are set as 210 and 80 MW. respectively.

## 6.2 Accumulator

In certain tests, an accumulator is added to the district heating network. Its parameters are specified in Table 3.

| **Parameter** | **Value** | **Unit** |
|---|---|---|
| Capacity | 1000 | MWh |
| Max transfer to accumulator | 100 | MW |
| Max transfer from accumulator | 100 | MW |
| Initial storage | 200 | MWh |
| Minimum end storage | 200 | MWh |
| Loss coefficient | 0.005 | |

Table 3: Accumulator parameters for test 2.

## 6.3 Demand

Before describing the properties of the demand used for the test cases, it is helpful to introduce some additional notations. When doing MPC as explained in Section

6.6, the time indices $i$ are shifted with each iteration and thus it is practical to also have a set of absolute time indices. Let therefore $Q_{t,b}$ and $Q_{t,C}$ denote the building heating demand and the hot water demand respectively, at absolute time $t$.

The customer demand varies with the time of the day, as explained in Section 2.1.3. In the stochastic model proposed in Section 4.5, the building heat demand varies linearly with expected slope $r = 0$. Thus the periodic daily variations are assumed to be concerning the hot water demand (remember, losses are being neglected). In the tests, the expected hot water demand $Q^*_{t,C}$ varies according to the function

$$Q^*_{t,C} = 30 + \sum_{n=0}^{3}(35 \cdot e^{-\frac{(t-n\cdot 24-7)^2}{5}} + 70 \cdot e^{-\frac{(t-n\cdot 24-17)^2}{c}}),\ t \in [1, 96]. \tag{70}$$

This gives a total expected demand that each day has a small peak at 7 a.m. and a second larger peak at 17 p.m., see Figure 4.[11] As can be seen, the starting value of building heat demand is set as

$$Q_{0,b} = 170 \text{ MW}. \tag{71}$$

While the expected demand follows a certain pattern, the actual demand is modelled stochastically, as described in Section 4.5. The standard deviation of the building heat demand and the hot water demand have been chosen to give reasonable results and are specified in Table 4.

| **Parameter** | **Value** | **Unit** |
|---|---|---|
| $\sigma(r)$ | $0.01/6 \cdot q_{I_a,b}$ | MW |
| $\sigma(q_{i,C})$ | $0.2 \cdot q^*_{i,C}$ | MW |

Table 4: Demand parameters, based on the formulation in Section 4.5.

In order to better understand the standard deviations of Table 4, let's calculate their values at some different time point. Starting with the building heating demand, we see by Equation (55) that the deviation from expected value is increasing linearly with the time elapsed from the start of the 'unknown' period.[12] For example at time $i = I_a + 6$ we have the standard deviation

$$\sigma(Q_{I_a+6,b}(\xi)) = 0.01 \cdot 170\,\text{MW} = 1.7\,\text{MW} \tag{72}$$

[11]The plotting is done with the function step in matplotlib.pyplot. The specification where='post' tells the function to keep the value of the previous sample until the next sample comes.

[12]It would perhaps be more intuitive to assume that the variance of the building heating demand to increase linearly from the current time, rather than having a constant variance of zero for the first period and then suddenly having a linear increase in variance. However, this assumption would result in a 'discontinuous' increase in variance from time $i = I_a$ to time $i = I_a + 1$, which also seems unrealistic.

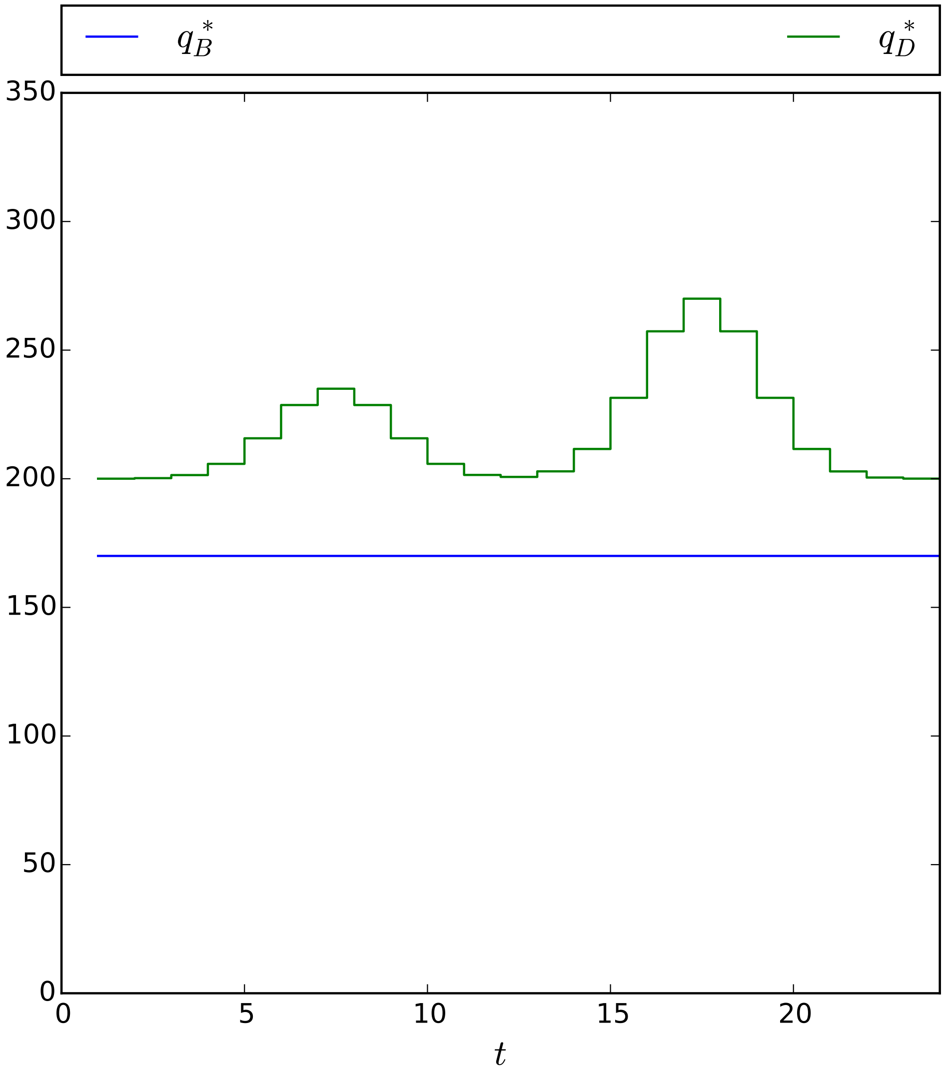


Figure 4: Expected demand curve for hours 1 to 24. The blue line seperates the parts assumed to be used for building heating (below) and hot tap water (above).

while at time $i = I_a + 12$ we have the standard deviation

$$\sigma(Q_{I_a+12,b}(\xi)) = 0.02 \cdot 170\,\mathrm{MW} = 3.4\,\mathrm{MW}. \tag{73}$$

In the MPC, we use an optimization horizon of 48 hours. The unknown part of this interval is $48 - 12 = 36$ hours long. The standard deviation in the last point will therefore be

$$\sigma(Q_{I_a+36,b}(\xi)) = 0.06 \cdot 170\,\mathrm{MW} = 10.2\,\mathrm{MW}. \tag{74}$$

The prediction error of the hot water demand on the other hand is not modelled as proportional to time, but as proportional to the hot water demand. Calculate therefore the standard deviation at the largest peak in Figure 4, $t = 17$. We have

$$\sigma(Q_{17,C}) = 0.2 \cdot (70 + 30)\,\mathrm{MW} = 20\,\mathrm{MW}. \tag{75}$$

On the other hand, the hot water demand in the "valleys", e.g. at time $t = 1$ or $t = 12$, is approximately 30 MW, which gives a standard deviation of

$$\sigma(Q_{t_1,C}) = 0.2 \cdot (30)\,\mathrm{MW} = 6\,\mathrm{MW}. \tag{76}$$

We see that with the current implementation, the hot water demand stands for most of the load prediction uncertainty for the early hours and at peaks. As the time moves further ahead however, the impact of the building heating variance increases. The sensitivity analysis in Section 7.5 makes tests for modified demand probability distributions.

## 6.4 Scenarios

Based on the stochastic demand model, four demand scenarios are constructed, such that at each point of time the outcome will be below the scenario with probability $p \in P$, where

$$P = \{0.25, 0.5, 0.75, 0.9\}. \tag{77}$$

An equivalent way of saying this is that the scenarios go through the 25-, 50-, 75- and 90-percentiles of the distributions at each point. The resulting demand trajectories are displayed in Figure 5. Of course, based on a continuous distribution, the probability that the demand would exactly follow one of these scenarios is infinitely small. However, in order to do stochastic programming we make the approximation that these scenarios together represent the whole class of scenarios. The probability for each scenario is estimated as 0.3, 0.4, 0.2 and 0.1 respectively.[13] The scenario properties are summarized in table 5.

| **Scenario** | **Percentile** | **Probability** |
|---|---|---|
| Scenario 1 | 25 | 0.3 |
| Scenario 2 | 50 | 0.4 |
| Scenario 3 | 75 | 0.2 |
| Scenario 4 | 90 | 0.1 |

Table 5: Description of scenarios. The percentile column indicates at each time point the percentage of demand values below the scenario.

[13]This estimation is based on the percentiles each scenario follows and on the shape of the normal distribution. Perhaps these probabilities could be approximated more rigorously, but this is not considered necessary in this context.

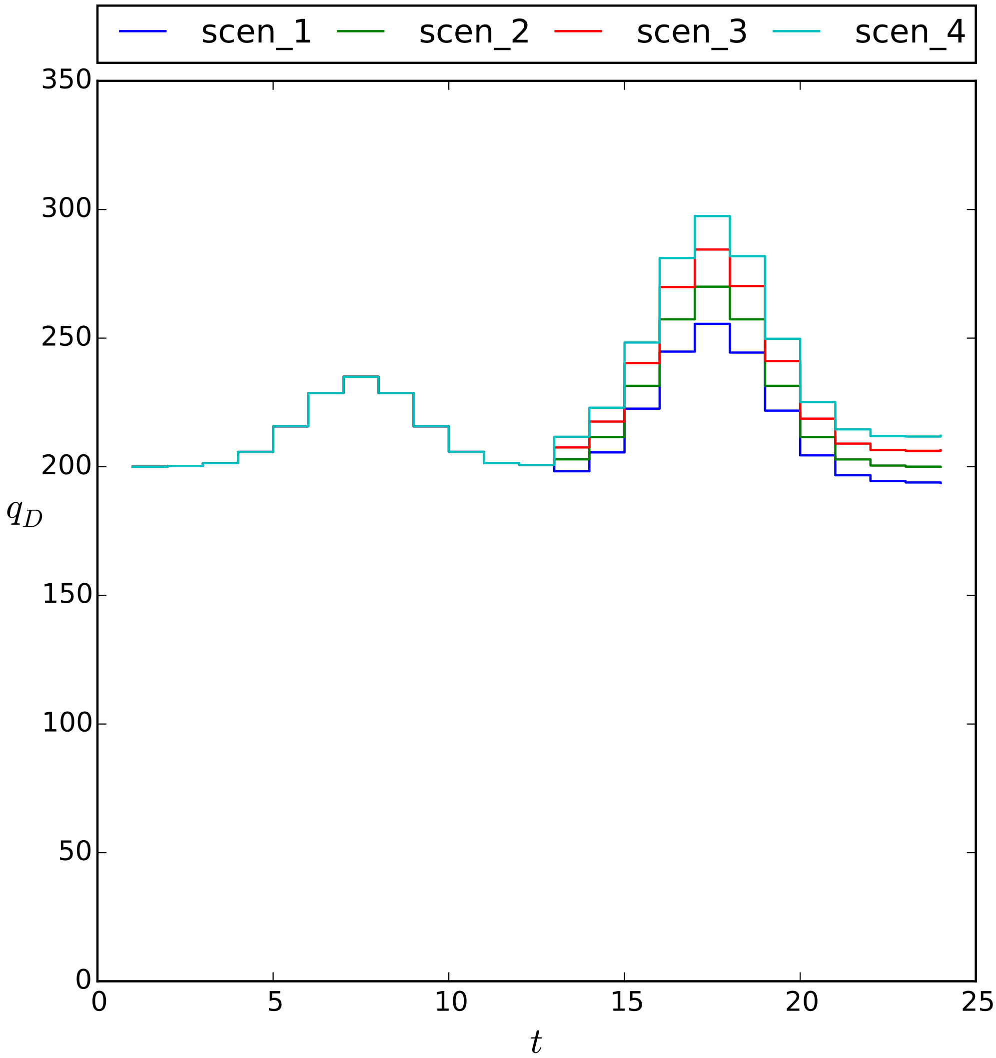


Figure 5: The demand trajectories for the four scenarios considered when performing optimization for hours 1-24.

### 6.5 Randomly Generated Demand Outcomes

The scenarios described in the previous subsection, Section 6.4, are used as realizations for the stochastic demand variable ($q_{i+\tau_c,D}(\xi)$) in the demand constraint (42), when solving for the first stage. However, the aim is that the resulting production plan should be able to supply for the demand not only in case of the scenarios of table 5, but for any demand outcome with a "reasonable" maximal load. Therefore, in Test 3, randomly generated demand outcomes are used to test robustness and performance.

In order to achieve results for outcomes with different properties, the following method is used to select a set of test realizations with different maximal loads:

1. Generate 1000 (or another large number) realizations

2. Sort the realizations by $\max_i q_{i+\tau_c,D}$ and number them so that the realization with lowest maximal demand has number 1 and the realization with highest maximal demand has number 1000

3. Pick out the realizations with numbers $n = 1000 \cdot p$, where $p \in P$ i.e. corresponding to the percentiles in table 5

The selected realizations are called Outcome 1-4 and are plotted in Figure 6. These will now have maximal loads which are exceeded with approximately probability $p$ (by the law of large numbers).

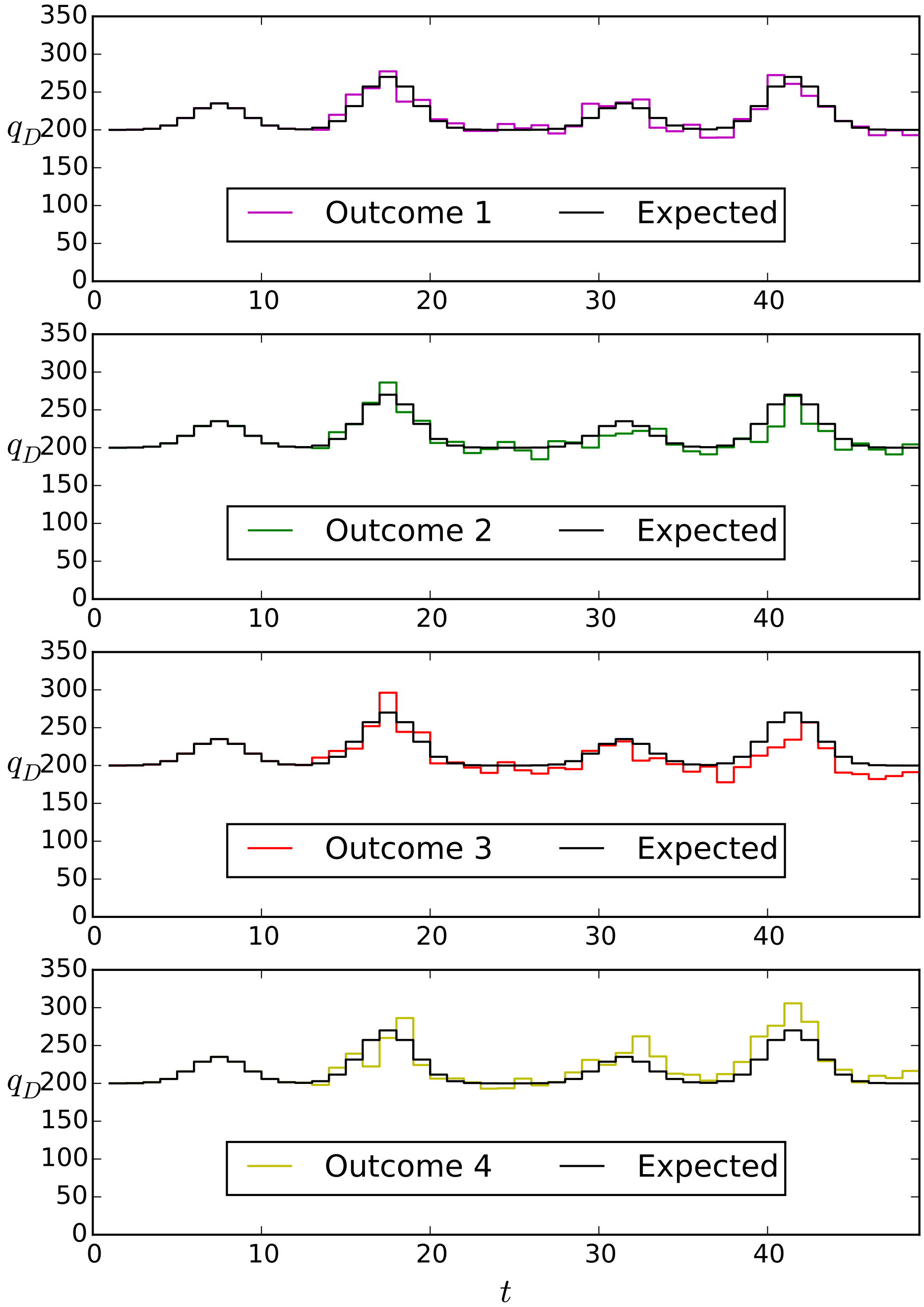


Figure 6: Four randomly simulated demand outcomes chosen from a large number of realizations, based on maximum loads. The outcomes have been chosen so that their maximal loads exceed the actual maximal load with approximately the probabilities in $P$, see (77). In other words, the top outcome has a maximal load which with 25 % exceeds the real outcome, the second one with 50 %, the third with 75 % probability and the bottom one with 90 % probability. The expected demand curve is included for comparison.

## 6.6 MPC

The tests described in Subsections 7.3 and 7.6 iterate the two stage problem solving, resulting in an MPC-strategy. This section explains how the test is set up.

The optimization horizon is set as 48 hours and the prediction horizon is set as 24 hours. At each iteration optimization is performed, either for the worst case scenario or using stochastic programming. It is assumed that new weather forecasts are given with six hour intervals and that no other information is updated regarding the behaviour of the system. The iterations are therefore not performed for every time step, but once in every six hours. Furthermore, as with the non-iterative two stage problems, it is assumed that perfect information is given for twelve hours forward in time. In stochastic programming, the length of Stage 1 can therefore be set to anywhere between 6 and 12 hours. For 12 hours the demand curves are identical for the different scenarios, but the decision variables will only be used for 6 hours, which gives a lower bound for $I_1$. However, as noted in Section 4.2, when a net model with time delays is used, there needs to be a margin so that

$$I_a \geq I_1 + \max_{c \in C} \tau_c. \tag{78}$$

Therefore, we set $I_1 = 6$. A summary of the MPC parameters is given in Table 6, where $t_{final}$ denotes the end time of the optimization horizon and $T$ denotes the period time.

| Parameter | Value |
|---|---|
| $t_{final}$ | 48 |
| $I_2$ | 24 |
| $I_1$ | 6 |
| $T$ | 6 |
| $I_a$ | 12 |

Table 6: Parameters for Test 3.

As an example, let's follow the MPC for Outcome 1. Figure 7 illustrates the prediction at the three first stages. At each step, the demand is assumed to be known for the first twelve hours but unknown for the following twelve hours. Therefore, four different scenarios are calculated for this period according to the description in Section 6.4. In iteration one, both the known demand in stage 1 and the scenarios of stage 2 are the same as in Figure 5. However, as we move to the second iteration, the demand for hours 13 - 18 is read from the trajectory of Outcome 1. Likewise in iteration three, the demand for hours 19 - 24 is read and entered into the prediction, and so on.

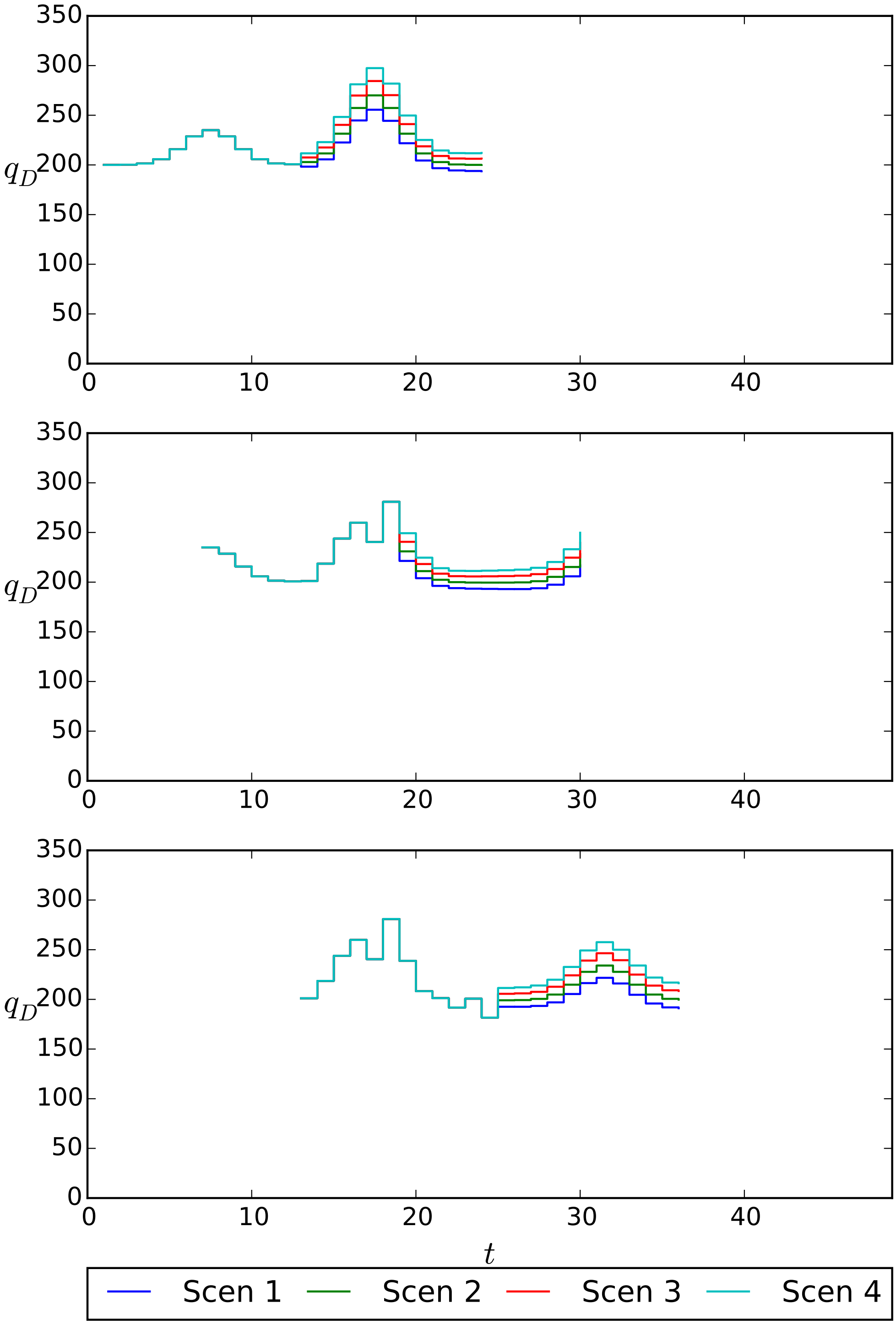


Figure 7: The figure describes the MPC process, showing generated scenarios for iterations 1 (top), 2 (middle) and 3 (bottom). At each iteration a prediction is made for 24 hours, assuming perfect information for the first twelve hours. The predictions are based on the case where the realized demand follows Outcome 1 (see Figure 6).

# 7 Results

Results are presented for tests comparing the stochastic programming approach to the results of worst case optimization. In Subsections (7.1) to (7.5), results are presented for PU Case 1, with maximal production limits according to (68). This includes the sensitivity analysis in Subsection (7.5). Section (7.6) uses production units according to PU Case 2, in order to test the benefits of stochastic programming when accumulation is enabled.

## 7.1 Test 1: Startup and Shutdown of Top-Up Units

The first tests compare results of stochastic programming and worst case optimization where Stage 1 and Stage 2 are both 12 hours long. Test 1.a handles a growing demand, requiring the startup of a top-up unit. A top-up unit is simply a unit that is meant to contribute to covering the heat load at times when the base load unit is insufficient. In contrast to Test 1.a, Test 1.b starts with a top-up unit being turned on, in order to see it a shutdown is motivated. Initial conditions for both tests are set according to Table 7. The costs for the two strategies are compared in table 8, showing that in Test 1.a the SP approach has expected savings of 2.4 percent compared to the WC approach. In Test 1.b the corresponding savings are 0.9 percent.

| **Parameter** | **Test 1.a** | **Test 1.b** |
|---|---|---|
| $t_{start}$ | 1 | 13 |
| $t_{end}$ | 24 | 36 |
| $\hat{u}_1$ | 1 | 1 |
| $\hat{q}_1$ | $q_{0,D}$ | $q_{0,D} - 15$ |
| $\hat{u}_2$ | 0 | 1 |
| $\hat{q}_2$ | 0 | 15 |
| $\hat{u}_3$ | 0 | 0 |
| $\hat{q}_3$ | 0 | 0 |
| baseLoad | 170 | 180 |

Table 7: Initial conditions for Test 1.a and Test 1.b. The time interval in test 1.b. is adjusted 12 hours forward and the SFB unit is turned on at the start of the interval.

The results of Test 1.a are shown in Figure 8 and Figure 9. We see that the worst case strategy is to start the solid fuel boiler, while the stochastic programming plan relies primarily on the fossil unit. In Scenario 4, the SP approach in addition has to start the SFB, since the fossil unit is not sufficient. Notice how this is a poor (although necessary) use of the SFB, since before the startup procedure is even finished, the demand has gone down again. This is reflected in Table 8 which shows that for Scenario 4, the SP approach is more expensive (2.3 %) than the WC approach. Indeed, for any two stage problem, the WC approach will by definition give the best result for Scenario 4, since its objective is to minimize the costs for Scenario

4, while the SP approach minimizes the expected costs.

The results of test 1.b are shown in Figure 10 and Figure 11. Here the SFB is initially turned on, together with the Base Unit. The WC approach keeps the SFB turned on, while the SP strategy turns it off and uses the Fossil unit for top up in Stage 2. Note that in this test, the base load was set to 180, compared to 170 in Test 1.a. If instead the base load is kept at 170 for Test 1.b, both WC and SP would turn off the SFB in Stage 1, resulting in identical strategies.

| | **SP Savings in 1.a [%]** | **SP Savings in 1.b [%]** |
|---|---|---|
| Scenario 1 | 3.6 | 1.6 |
| Scenario 2 | 2.7 | 1.0 |
| Scenario 3 | 2.3 | 0.2 |
| Scenario 4 | -2.3 | -0.4 |
| Expected value | 2.4 | 0.9 |

Table 8: Savings from using stochastic programming (SP) compared to worst case optimization (WC), for Tests 1.a and 1.b.

## 7.2 Test 2: Compensation with Accumulator

In Test 2, an accumulator is added to the system of Test 1.a, with parameters specified in Table 3. The purpose of the test is to see how uncertainty can be handled with the accumulator, and if stochastic programming still can add benefits. Naturally, the results are dependent on the capacity of the accumulator.

Figure 12 display the results of worst case optimization. With the added accumulator, the full heat load can be managed using the Base production unit and no top-up unit is needed. As a result, there are no benefits of using stochastic programming, since SP gives the same strategy as the WC approach. A calculation of the savings compared to the WC production plan without accumulator is shown in Table 9.

| | **Cost in 1.a [k SEK]** | **Cost in 2.a [k SEK]** | **Savings with Ac** |
|---|---|---|---|
| Scenario 1 | 1062.4 | 1006.0 | 5.3 % |
| Scenario 2 | 1086.6 | 1024.3 | 5.7 % |
| Scenario 3 | 1111.0 | 1042.8 | 6.1 % |
| Scenario 4 | 1133.5 | 1059.5 | 6.5 % |
| Expected value | 1088.9 | 1026.0 | 5.8 % |

Table 9: Savings from using the WC approach with accumulator compared to WC optimization without accumulator. Costs are given in thousands of SEK.

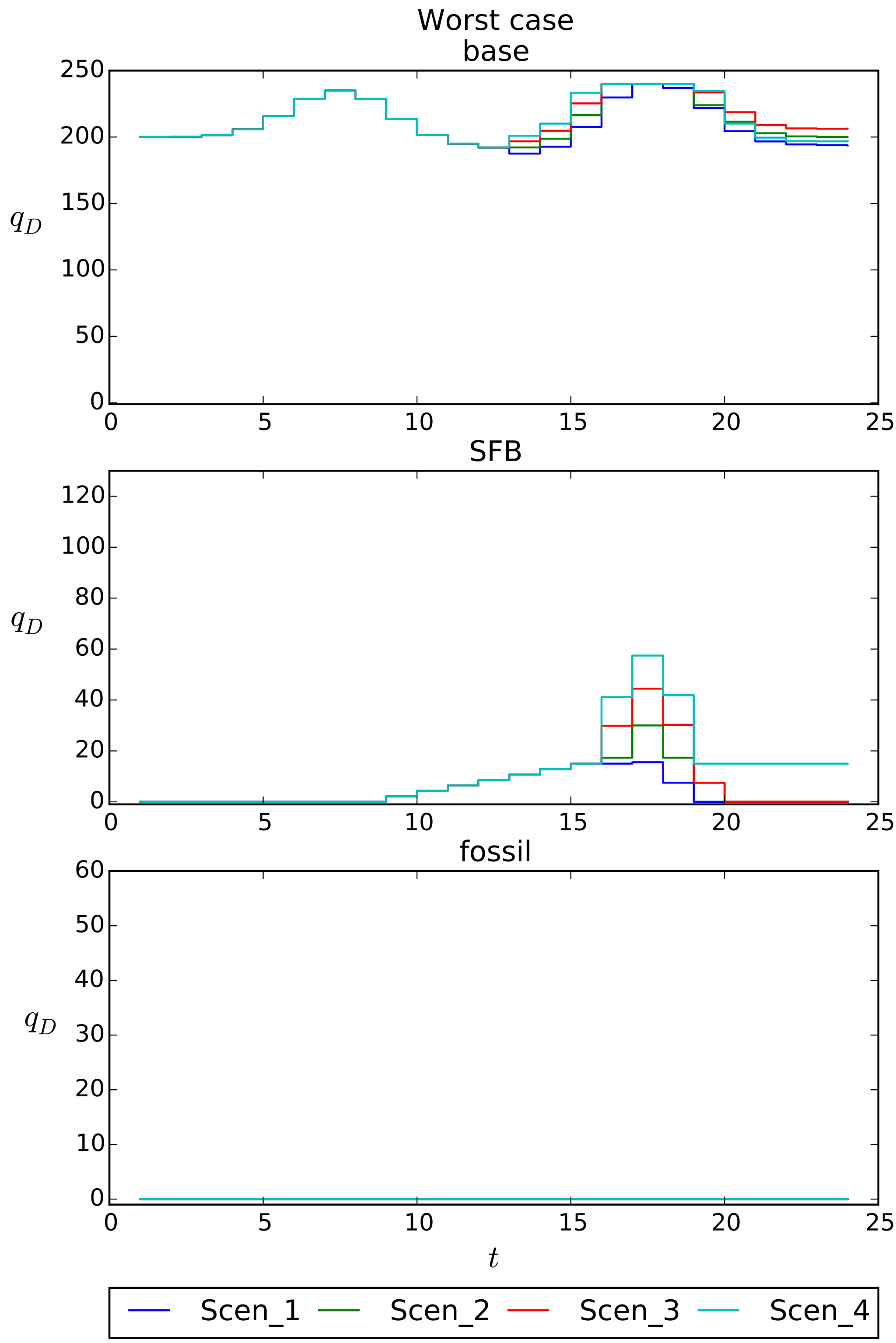


Figure 8: The production plan optimized for the worst case scenario in test 1.a. Initially only the base load unit is running. Towards the end of Stage 1, the solid fuel boiler is turned on to meet the increasing demand doming in the Stage 2.

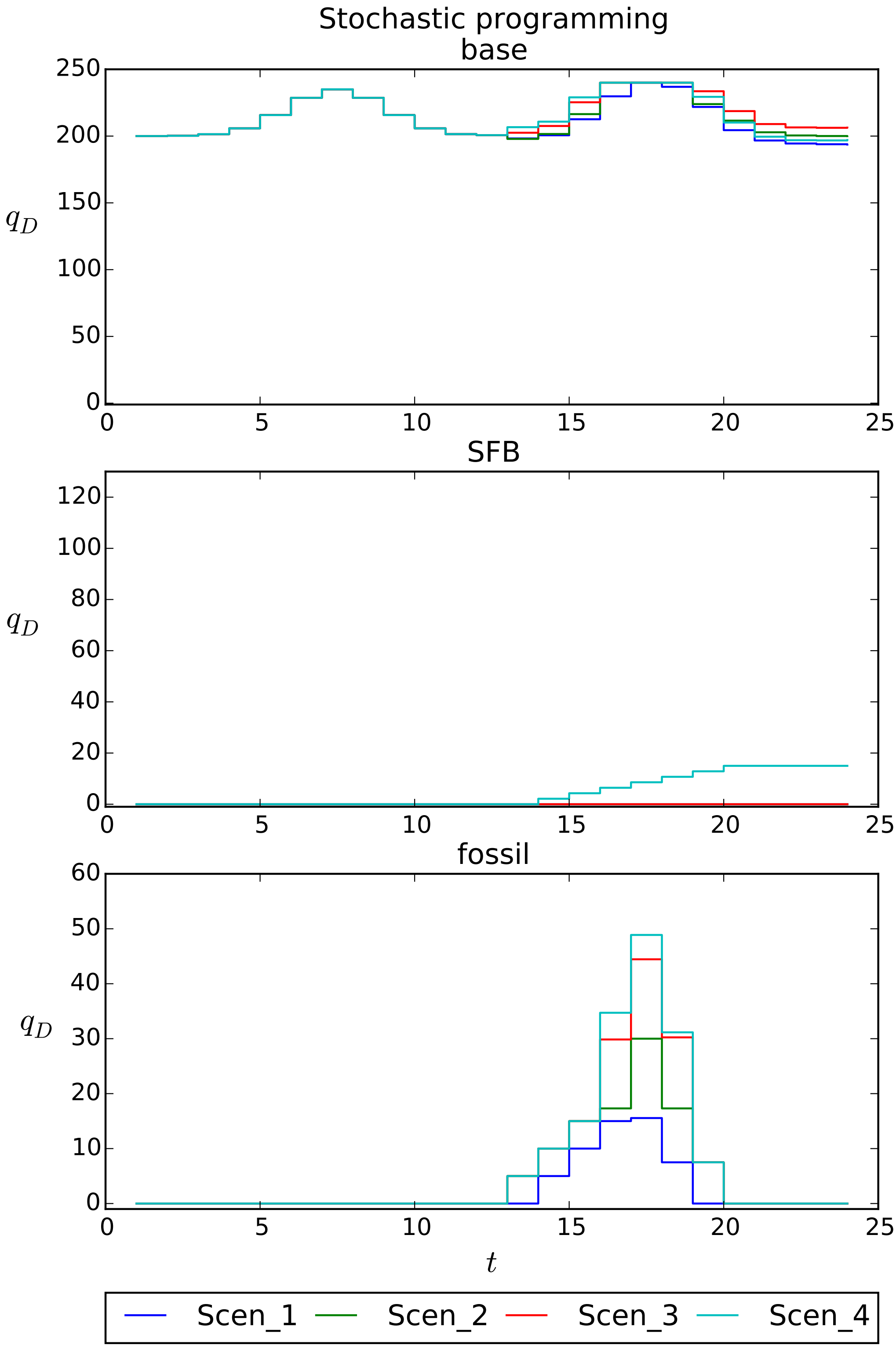


Figure 9: The production plan attained using stochastic programming in test 1.a. Again only the base load unit is running at the start, but this time the solid fuel boiler is not turned on in Stage 1. Instead it is primarily the fossil fuel boiler that works as a top-up unit, and since it has a short startup time, it is turned on first in Stage 2. However, since it has a maximal production of 50 MW, the worst case scenario requires also the solid fuel boiler to be turned on.

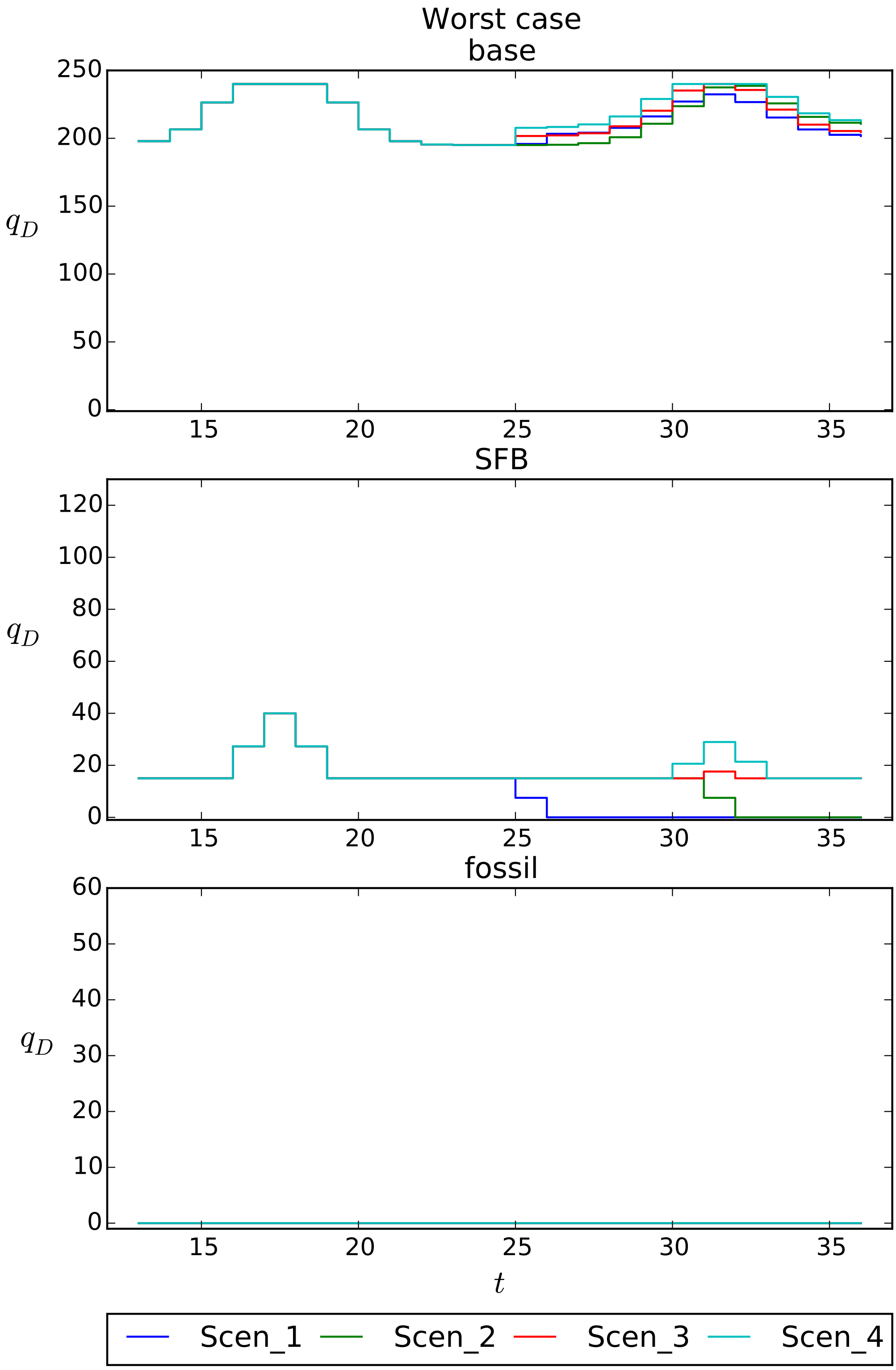


Figure 10: The production plan optimized for the worst case scenario in Test 1.b. At the start both the base load unit and the solid fuel boiler are turned on. Since the production plan is optimized for the worst case scenario, the solid fuel boiler is kept turned on until the start of Stage 2. In Stage 2 it is turned off for Scenarios 1 and 2, while Scenarios 3 and 4 keeps it running.

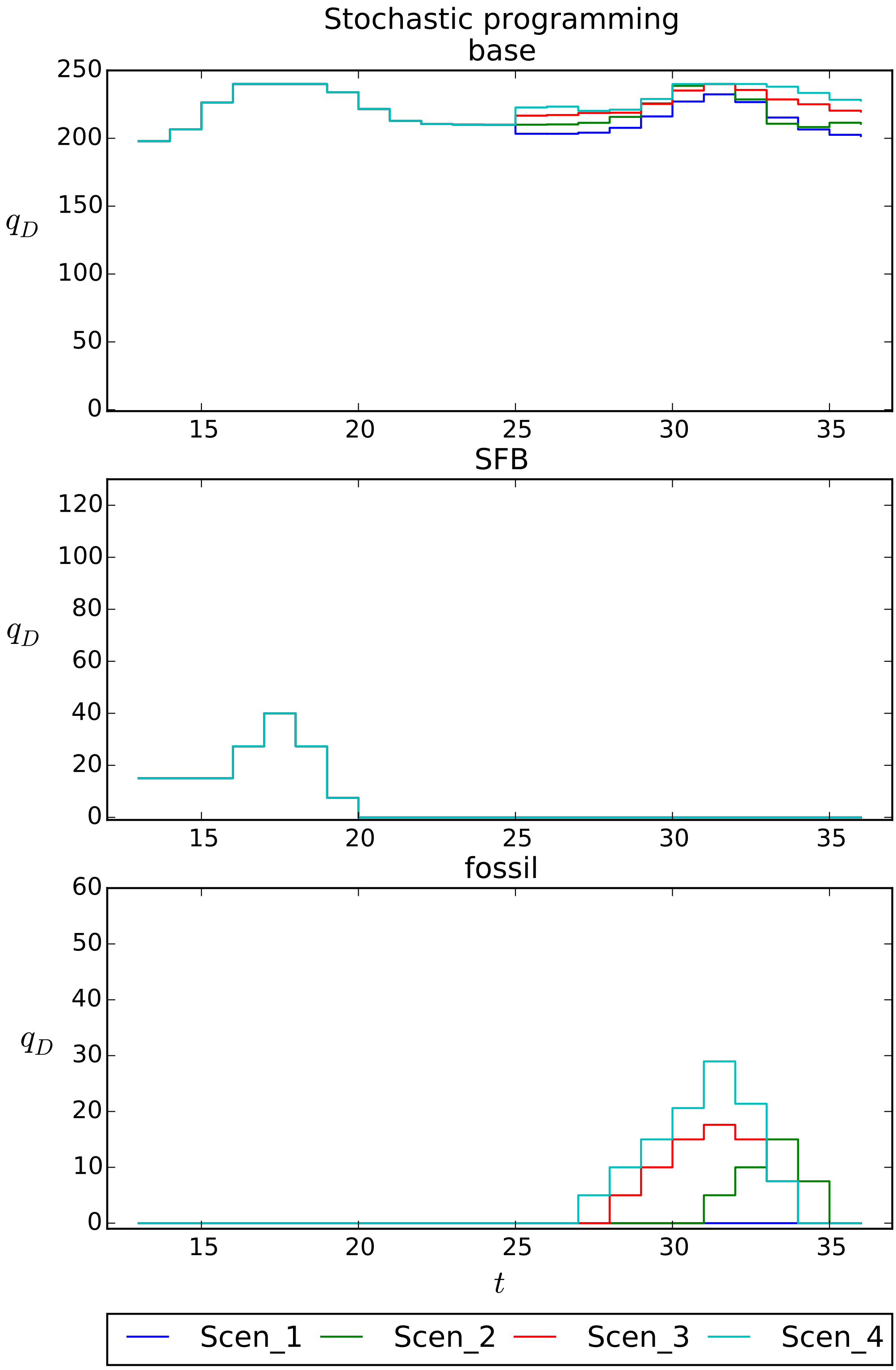


Figure 11: The production plan attained using stochastic programming in Test 1.b. Initially the base load unit and the solid fuel boiler are both turned on. Since the production is optimized for the expected value, it is not worth to keep the solid fuel boiler running into Stage 2. Instead it is turned off and the fossil fuel boiler is used as a top-up unit in Stage 2.

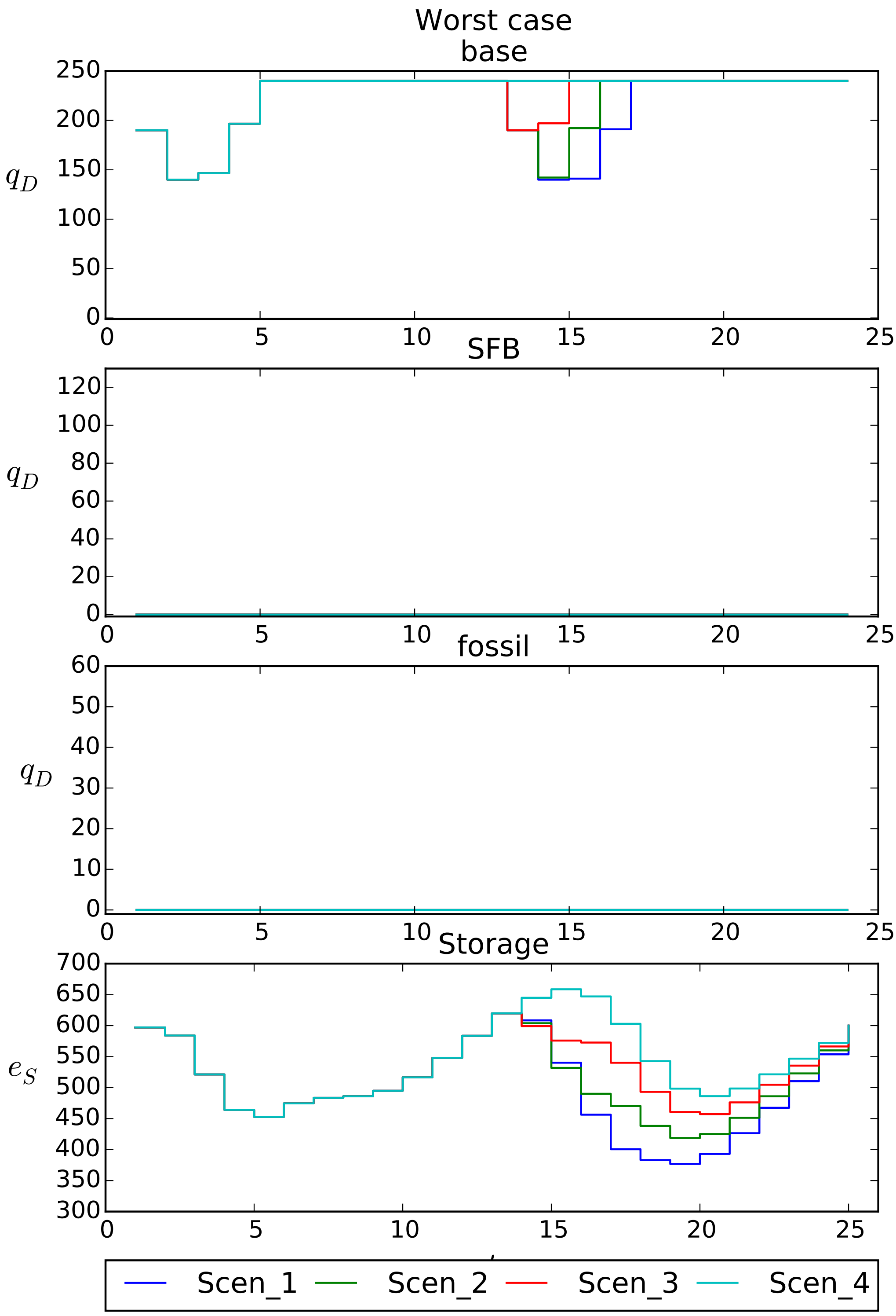


Figure 12: The results of worst case optimization with initial conditions of Test 1.a (see Table 8) an the accumulator specified in Table 3. No top-up unit is needed, since the heat from the base load unit can be stored from times of low demand to times of peaks.

## 7.3 Test 3: Iteration and MPC

Test 3 performs MPC, as described in Section 6.6. Four outcomes are simulated as described in Section 6.5. They are plotted in Figure 6 and a copy of the figure is included here in Figure 13 to simplify comparisons. The results are displayed in Figure 14, Figure 15, Figure 16 and Figure 17.

A summary of the savings of stochastic programming is provided in Table 10. The numbers show that the WC approach is slightly cheaper for Outcome 4, while for the other outcomes the SP strategy is most beneficial. Looking at Figure 17, we see that the WC approach starts the SFB at time $i = 24$, which turns out helpful to meet the two demand peaks that follow, see Figure 13. The same SFB start occurs for Outcome 1 and Outcome 2, but here the peaks are lower and would better be handled by the fossil unit. The strategies for Outcome 3, see Figure 16 show an example where SP closes down the SFB, while the WC approach keeps it running in vain.

| | **WC Costs [k SEK]** | **SP Costs [k SEK]** | **Savings [%]** |
|---|---|---|---|
| Scenario 1 | 2168 | 2129 | 1.78 |
| Scenario 2 | 2140 | 2073 | 3.12 |
| Scenario 3 | 2116 | 2080 | 1.69 |
| Scenario 4 | 2230 | 2235 | -0.25 |
| Expected cost | 2152 | 2108 | 2.10 |

Table 10: Comparison of costs in test 3 for stochastic programming (SP) and worst case optimization (WC).

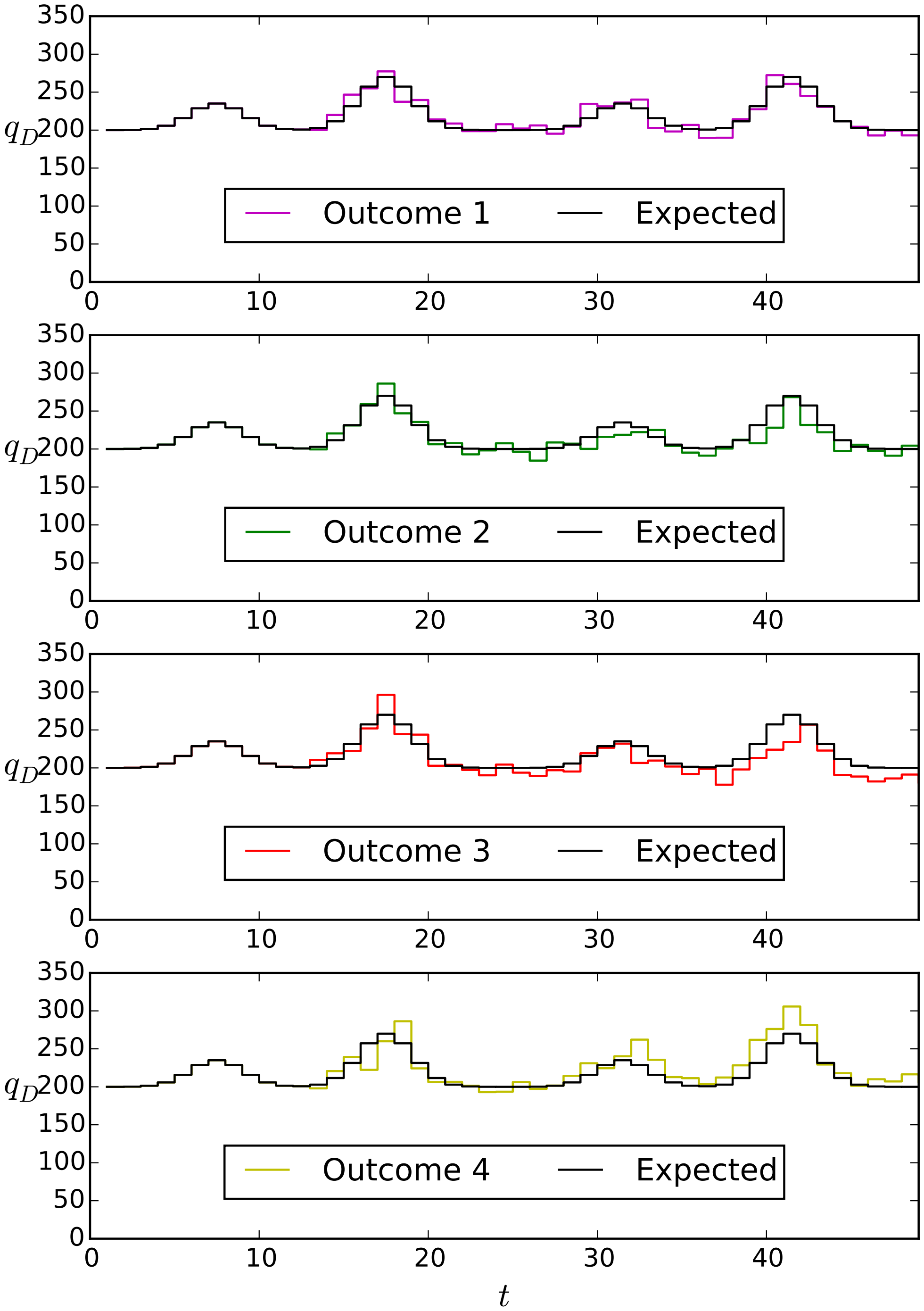


Figure 13: Randomly generated outcomes used to test stochastic programming and worst case optimization in combination with MPC. This figure is a copy of Figure 6, by which you find further information in its caption.

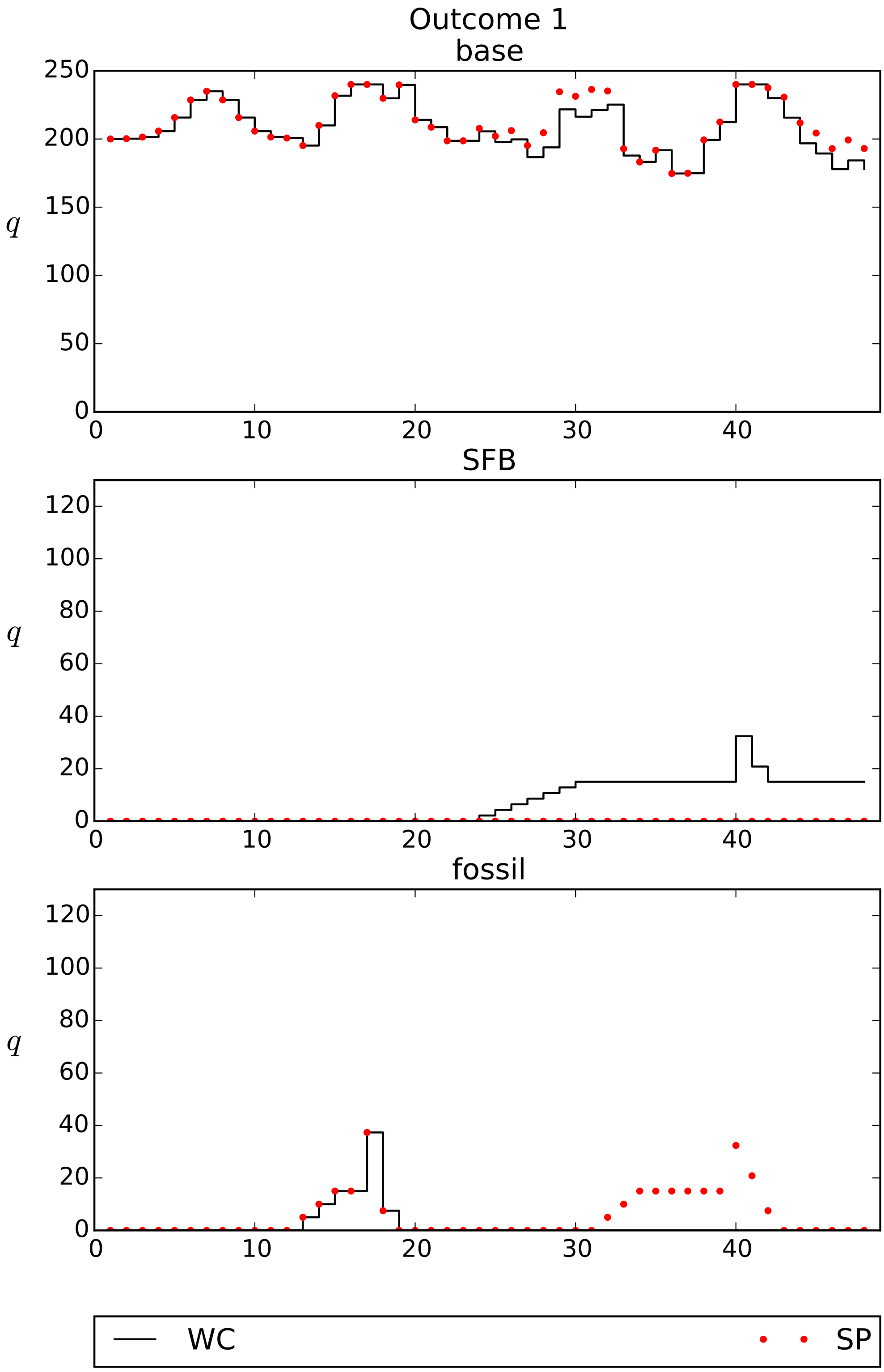


Figure 14: The results for Test 3, doing MPC using stochastic programming and worst case optimization respectively, when the outcome follows Outcome 1. Worst case optimization starts the fossil fuel unit once and the solid fuel boiler once, while stochastic programming uses the fossil fuel boiler twice and leaves the solid fuel boiler unused.

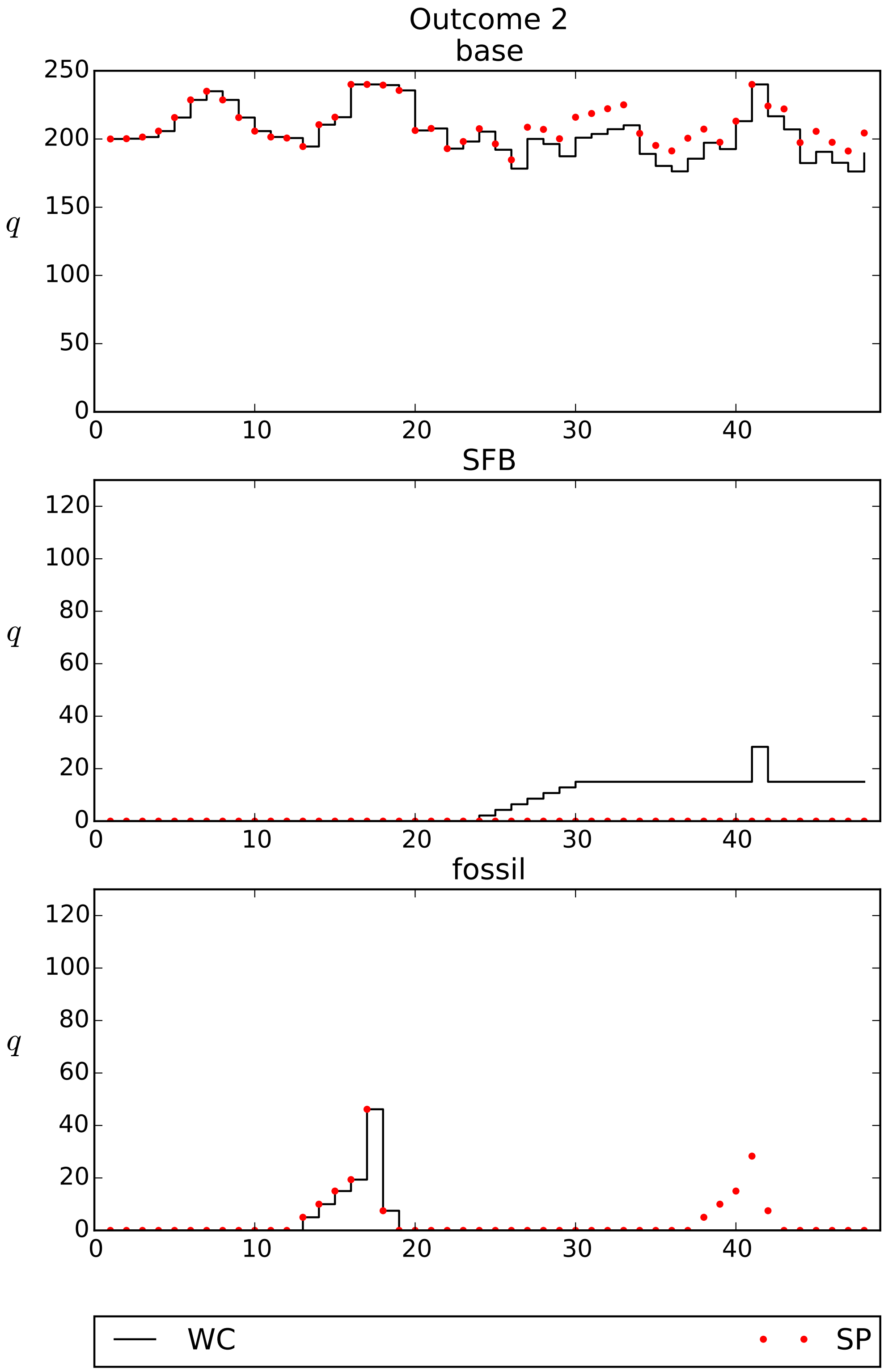


Figure 15: The results for Test 3 when the outcome follows Outcome 2. As in the previous figure, worst case optimization starts the fossil fuel unit once and the solid fuel boiler once, while stochastic programming uses the fossil fuel boiler twice and leaves the solid fuel boiler unused.

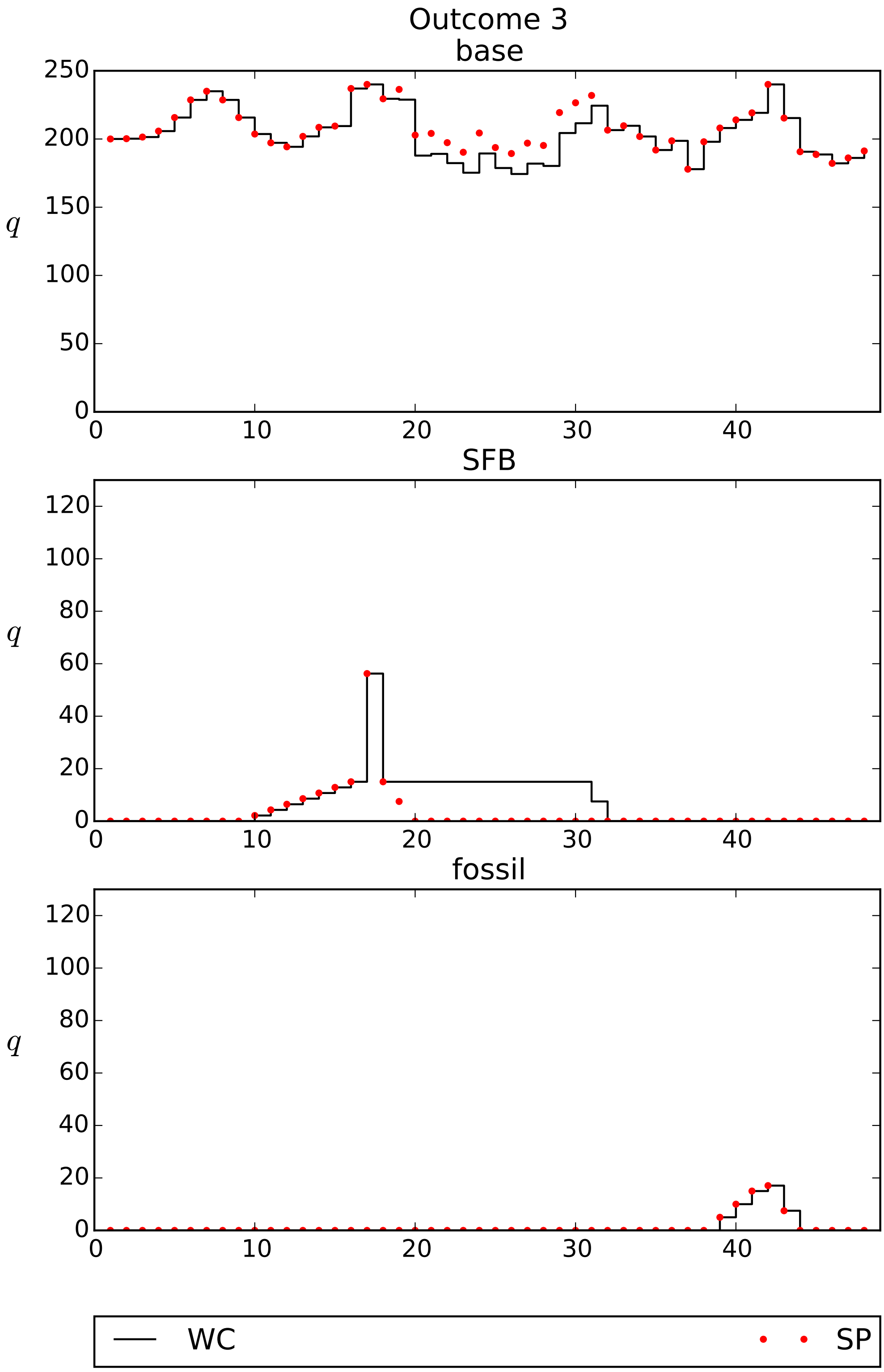


Figure 16: The results for Test 3 when the outcome follows Outcome 3. Both strategies start up the solid fuel boiler at time $t = 10$ and the fossil fuel boiler at time $t = 39$. The only difference comes from the fact that the worst case optimization keeps the solid fuel boiler turned on longer than in the case of stochastic programming.

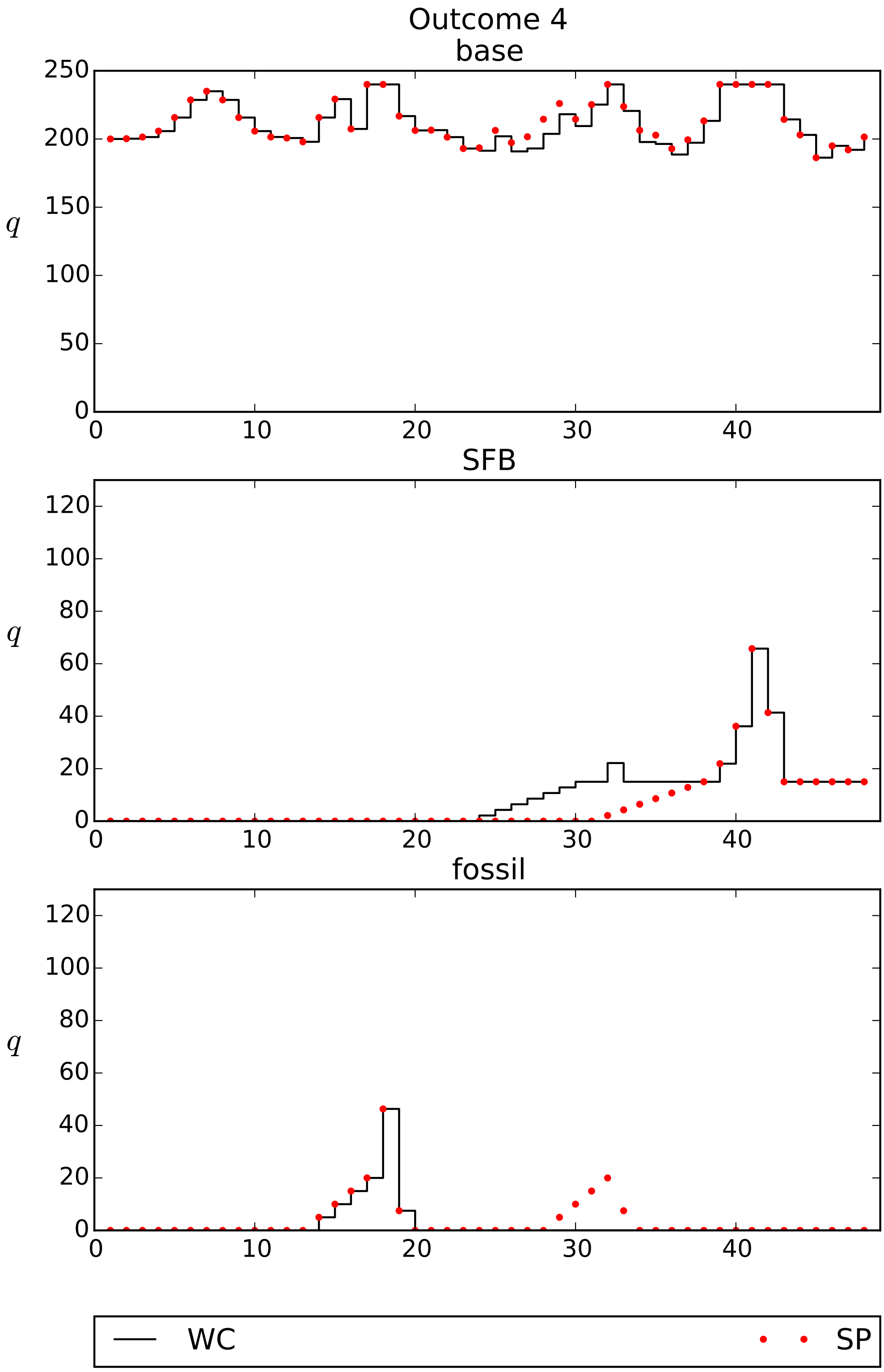


Figure 17: The results for Test 3 when the outcome follows Outcome 4. Both strategies start up the solid fuel boiler once, but with stochastic programming later than with worst case optimization. In return, the stochastic programming strategy requires a second start of the fossil fuel unit, while the worst case approach only starts it once.

### 7.3.1 Test 3.b - Extended Prediction Horizon

In Test 3.b, the prediction horizon is increased, from 24 hours to 48 hours. This gives the results displayed in figures 18 to 21. The costs are summarized in Table 11. Comparing to Test 3, we see that for Outcome 1, 2 and 4, the increased prediction horizon causes the WC strategy to turn on the SFB already for the evening peak of the first day ($t = 19$). For Outcome 3 on the other hand, the SFB is not turned off, but kept on for the remainder of the optimization interval. With these changes, the WC approach actually gives higher expected costs with the increased prediction horizon, compare Table 10 and Table 11. The SP costs however remain constant.

| | **WC costs [k SEK]** | **SP costs [k SEK]** | **Savings [%]** |
|---|---|---|---|
| Scenario 1 | 2176 | 2129 | 2.17 |
| Scenario 2 | 2147 | 2073 | 3.45 |
| Scenario 3 | 2123 | 2080 | 2.01 |
| Scenario 4 | 2234 | 2235 | -0.06 |
| Expected cost | 2160 | 2108 | 2.43 |

Table 11: Comparison of costs in test 3.b for stochastic programming (SP) and worst case optimization (WC).

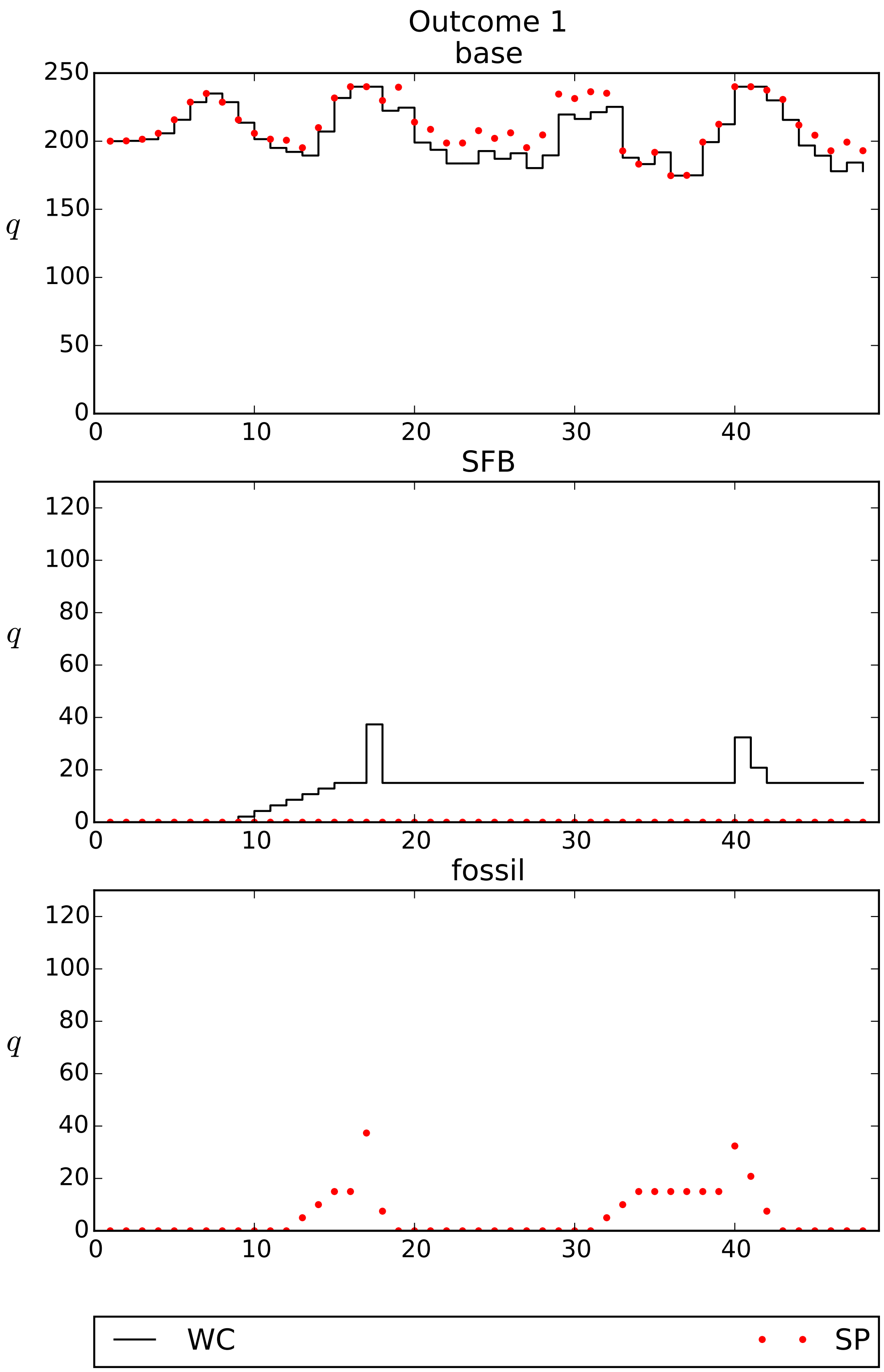


Figure 18: The results for Test 3b when the demand follows Outcome 1. The worst case optimization uses the solid fuel boiler as top-up unit, turning it on at time $t = 9$ and keeping it running throughout the simulation interval. The stochastic programming strategy on the other hand is use the fossil fuel boiler as top-up unit, turning it on for the peaks, but keeping it off in between.

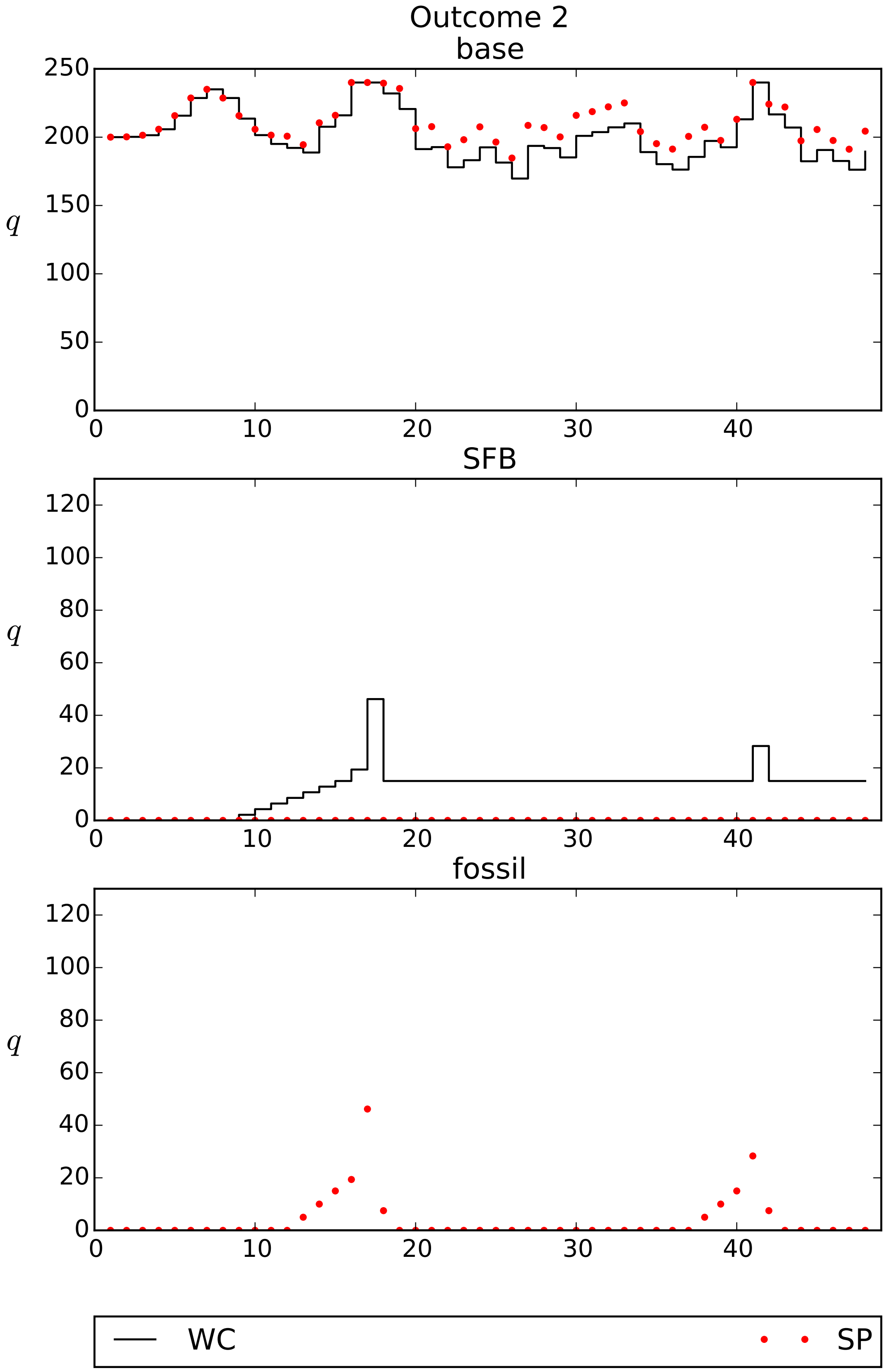


Figure 19: The results for for Test 3b, when the demand follows Outcome 2. As in the previous figure, the worst case optimization uses the solid fuel boiler as top-up unit, keeping it running throughout most of the simulation interval. The stochastic programming strategy on the other hand is instead of the solid fuel boiler, use the fossil fuel boiler. It is turned on for the peaks, but kept off in between.

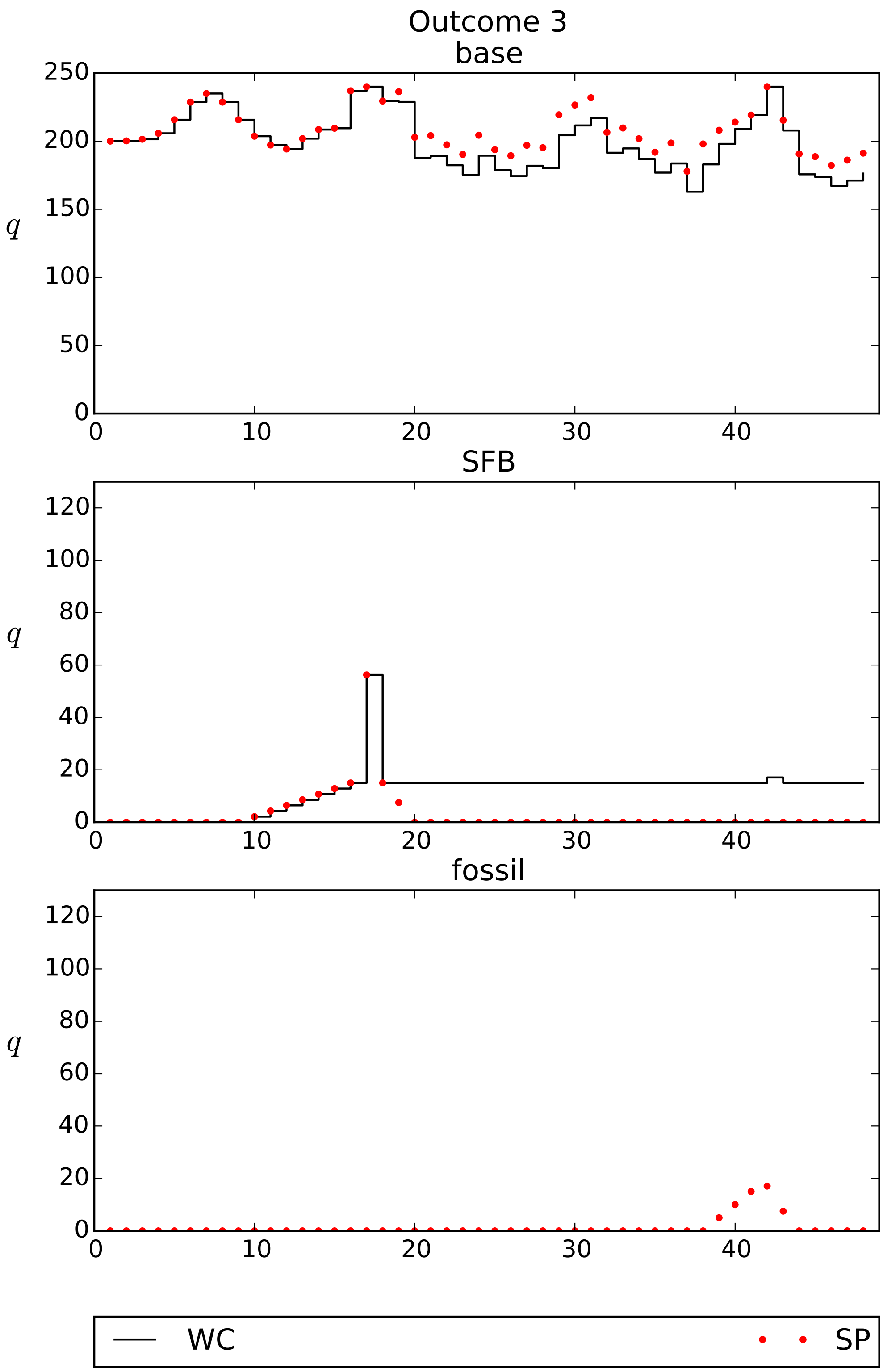


Figure 20: The results for Test 3b, the demand following Outcome 3. The worst case approach is similar to in the two previous figures, using the solid fuel boiler as top-up unit and keeping it turned on once it's started up. The stochastic programming approach on the other hand differs from in the previous two figures. This time it turns on the solid fuel boiler at the same time as with the worst case approach. However, once the demand peak has passed, the solid fuel boiler is again turned off and the second major peak is covered by the fossil fuel boiler.

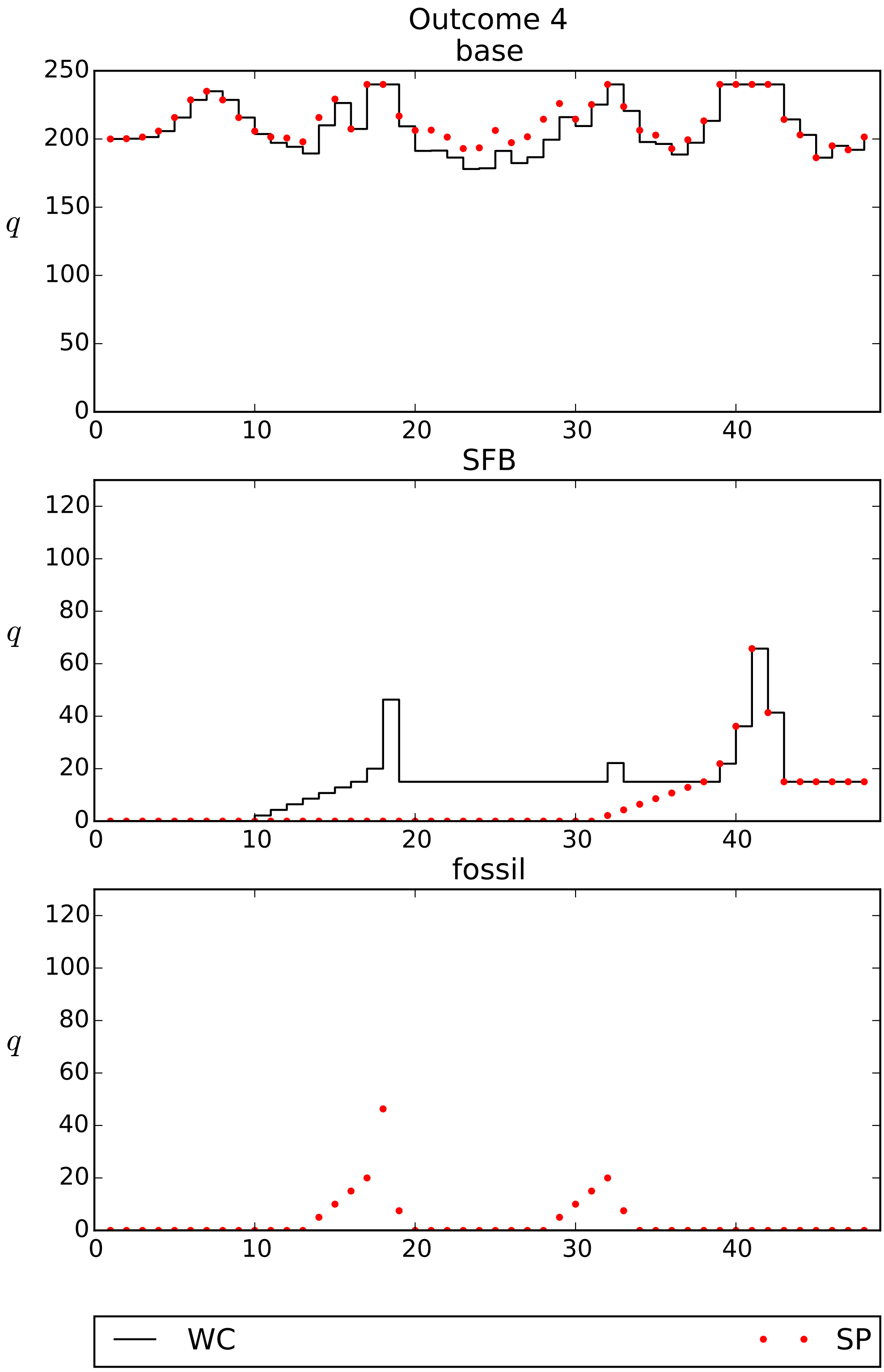


Figure 21: The results for Test 3b, the demand following Outcome 4. The worst case approach is the same as in the previous figures, relying on the solid fuel boiler as top-up unit. The stochastic programming approach here satisfies two peaks with the fossil fuel boiler, before it turns on the solid fuel boiler towards the end.

## 7.4 Test 4: With Net Model

Test 4 shows a simple test where the net model is in use and thus the demand constraint (11) is substituted with (12). Two customers $c_1$ and $c_2$ are considered, with delays $\tau_{c_1} = 1$ and $\tau_{c_2} = 2$. The demand for each customer is at each time point half of the demand in Test 1. A simple two stage optimization is done and the remaining parameters are the same as in Test 1. The resulting production plan for WC optimization is shown in Figure 22 and the resulting production plan for SP is identical.

Comparing Figure 22 to Figure 8 and Figure 9, we see that the production top is moved one step backwards in time, which is caused by the delay. More importantly, we see that the plans have significantly changed: The WC strategy has abandoned the SFB for the fossil unit and the SP strategy no longer needs to start the SFB for the worst case scenario. This is caused by the fact that when the demand is split over different customers with different delays, but the same demand trajectory, the demand tops will occur at different times and thus the total demand - given as the superposition of the customer demands - will have a peak that is lower and wider.

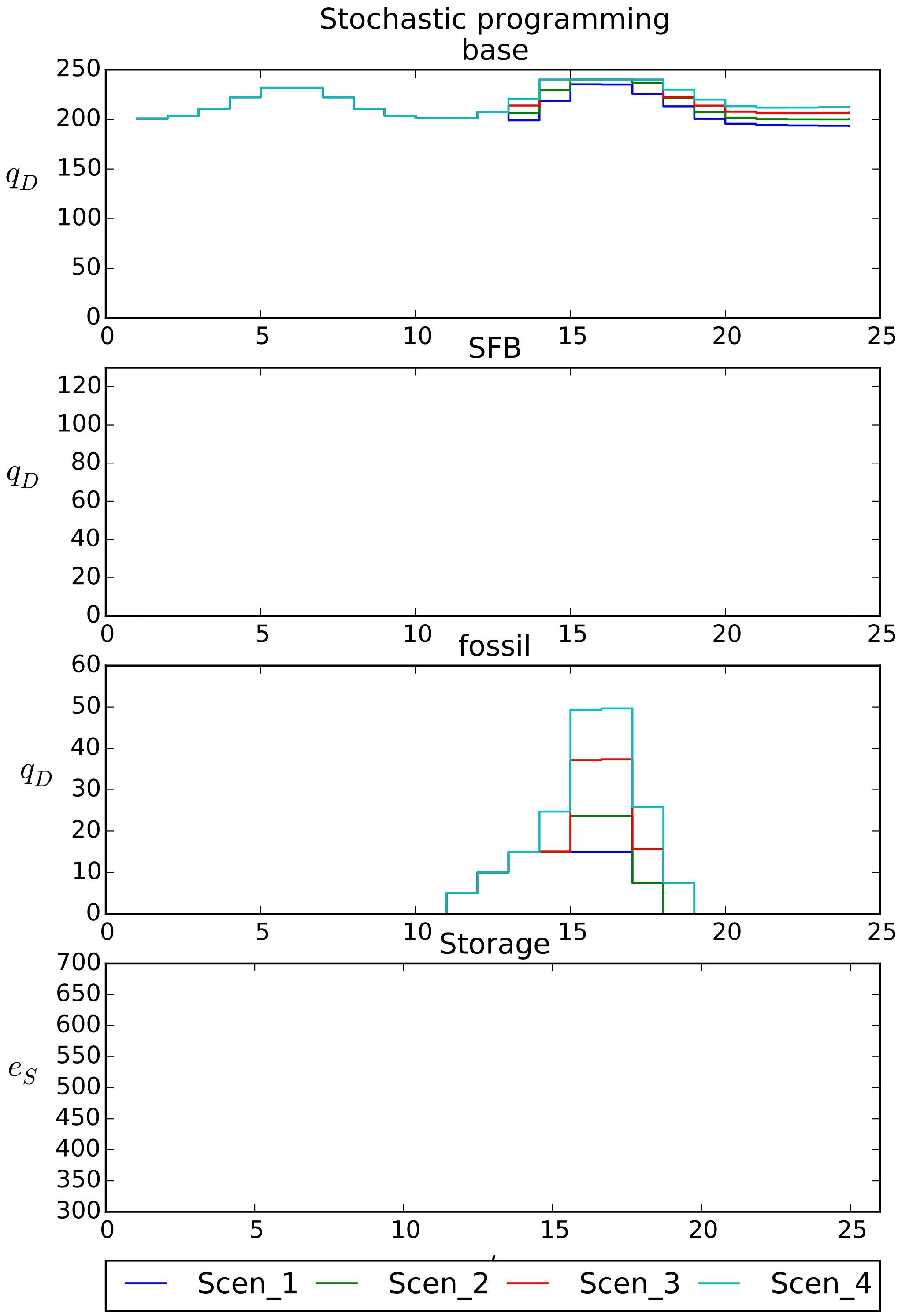


Figure 22: The worst case optimization results for Test 4 where the heat load is split equally between two customers with delays $\tau_{c_1} = 1$ and $\tau_{c_2} = 2$. Compare to Test 1.a, where the total demand is the same, but where no net model is considered. The production plan resulting from stochastic programming is identical.

## 7.5 Sensitivity Analysis

The demand parameters in Table 4 are chosen intuitively, to give reasonable scenarios. It is therefore interesting to investigate the sensitivity in results to changes in assumed probability distribution. The choice of probability distribution affects both simulated outcomes and scenarios. A sensitivity analysis is done with MPC, using the same production units and initial conditions as in Test 3. However, the simulated outcomes are here not randomly generated but instead follow the pattern for the scenarios described in Section 6.4. In other words, they are constructed so that at each point they exceed the actual outcome with probabilities in (77). The outcomes based on the default probability distribution are plotted in Figure 23.

The reason for avoiding randomly generated outcomes is that we want to construct similar outcomes for each test case, but adjusted according to the change in probability distribution. If the outcomes would be independently randomly generated, it would be hard to draw conclusions on what is caused by a changed standard distribution. On the other hand, constructing the outcomes as in Figure 23 means that the outcomes follow the trajectories of the scenarios. This is a disadvantage, since the outcomes then exactly follow the different plans, which would never happen in practice.

The resulting costs for these default outcomes are displayed in Table 12. Comparing it to the case when random outcomes are used (c.f. Table 10), we see that the savings are similar in both cases. For outcomes 1-3, the benefits of stochastic programming are slightly decreased, which may be because the outcomes all follow the same shape as the scenarios and therefore the WC optimization plan will work well for any outcome. For outcome 4 the benefits are instead increased, since at each iteration a 12-hour segment of this demand curve has been one of the possible scenarios. However, despite this differences from working with random outcomes, the sensitivity analysis should still give an indication of the relation between probability distribution and the benefits of stochastic programming.

| | **WC costs [k SEK]** | **SP costs [k SEK]** | **Savings [%]** |
|---|---|---|---|
| Outcome 1 | 2046 | 2015 | 1.51 |
| Outcome 2 | 2171 | 2114 | 2.63 |
| Outcome 3 | 2256 | 2229 | 1.20 |
| Outcome 4 | 2330 | 2330 | 0 |
| **Expected** | 2166 | 2129 | 1.74 |

Table 12: Costs for worst case optimization and stochastic programming, with outcomes based on the default probability distribution (see Figure 23).

Now, four simulations are performed, each one adjusting the value of one parameter in the stochastic demand model. The outcomes for the four simulations are shown in Figures 24 to 27. Resulting costs are summarized in Table 13.

To exemplify what can be the effects of a different probability distribution, consider the results for Outcome 1, shown for the default probability distribution in Figure 28. Looking at Table 13, we see that the SP strategy is 1.51 percent cheaper than the WC strategy. On the contrary, looking at Figure 28, the two strategies seems identical. However, a sharp eye may notice that for time $t = 48$ in the SFB plot, there is a tiny black dot sticking up behind the red dot. This marks a startup of the SFB unit for the WC strategy, which explains why it's more expensive than the SP plan.

Now let's analyse what happens when $\sigma_{q_{i,C}}$ is increased to 0.4. Increasing $\sigma_{q_{i,C}}$ means to increase the variation of hot water demand, which in turn means higher peaks for the worst case scenario. In the MPC performed for Outcome 1 and with the WC strategy, this forces a startup of the SFB at time $t = 24$, see Figure 29. However, as Outcome 1 has relatively low peaks, this startup turns out unnecessary and with stochastic programming it is avoided. This explains the increased savings of stochastic programming seen in Table 13, for Outcome 1, $\sigma_{q_{i,C}} = 0.4$.[14]

Now consider instead decreasing $\sigma_{q_{i,C}}$ to 0.01, resulting in lower peaks for the worst case demand. According to Table 13, this reduces the SP savings for Outcome 1 to zero. Looking closely at Figure 30, one can see that this is because the startup of the SFB for the WC strategy at time $t = 48$ is no longer neccesary.

| **Savings [%]** | Default | $\sigma_r = \frac{0.02}{6}$ | $\sigma_r = \frac{0.001}{6}$ | $\sigma_{q_{i,C}} = 0.4$ | $\sigma_{q_{i,C}} = 0.01$ |
|---|---|---|---|---|---|
| Outcome 1 | 1.51 | 4.43 | 0 | 5.36 | 0 |
| Outcome 2 | 2.63 | 2.63 | 0 | 2.63 | 0 |
| Outcome 3 | 1.19 | 0.09 | 0.47 | 0 | 1.42 |
| Outcome 4 | 0 | 0 | 0 | 0 | 0.49 |
| **Expected** | 1.74 | 2.40 | 0.09 | 2.66 | 0.33 |

Table 13: Sensitivity analysis, showing expected savings of stochastic programming compared to worst case optimization. The row decides which demand outcome is considered and the column which values of the standard deviations $\sigma_{q_C(t)}$ and $\sigma_r$ are used. Only one parameter is changed at a time.

[14]The barely visible startup of the SFB for the WC strategy at time $t = 48$ occurs just as for the default $\sigma_{q_{i,C}} = 0.2$.

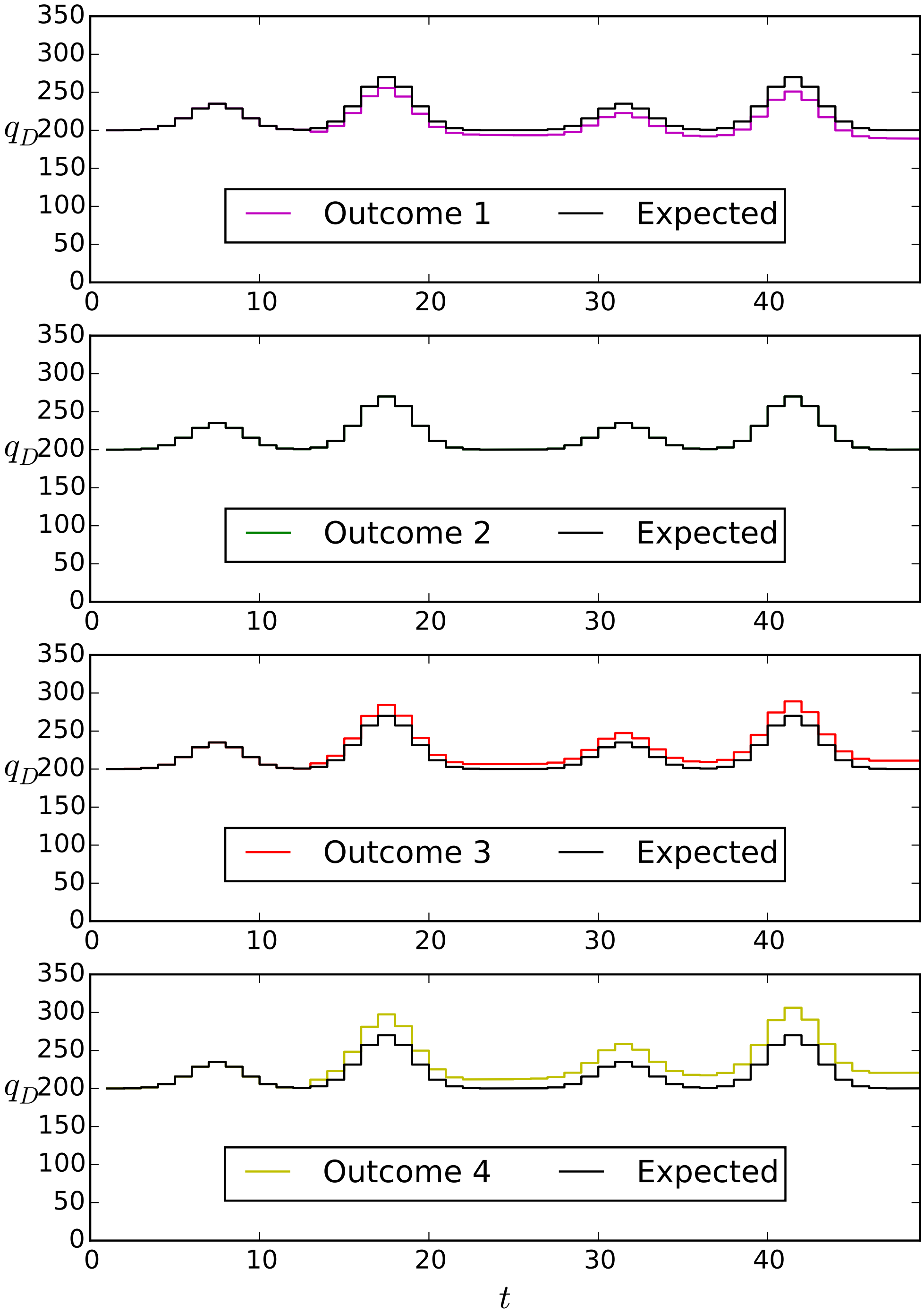


Figure 23: Outcomes used for the sensitivity analysis, for the case with default probability distribution. Outcome 1 follows the 25-percentile, Outcome 2 is based on the expected values, Outcome 3 follows the 75-percentile and Outcome 4 follows the 90-percentile.

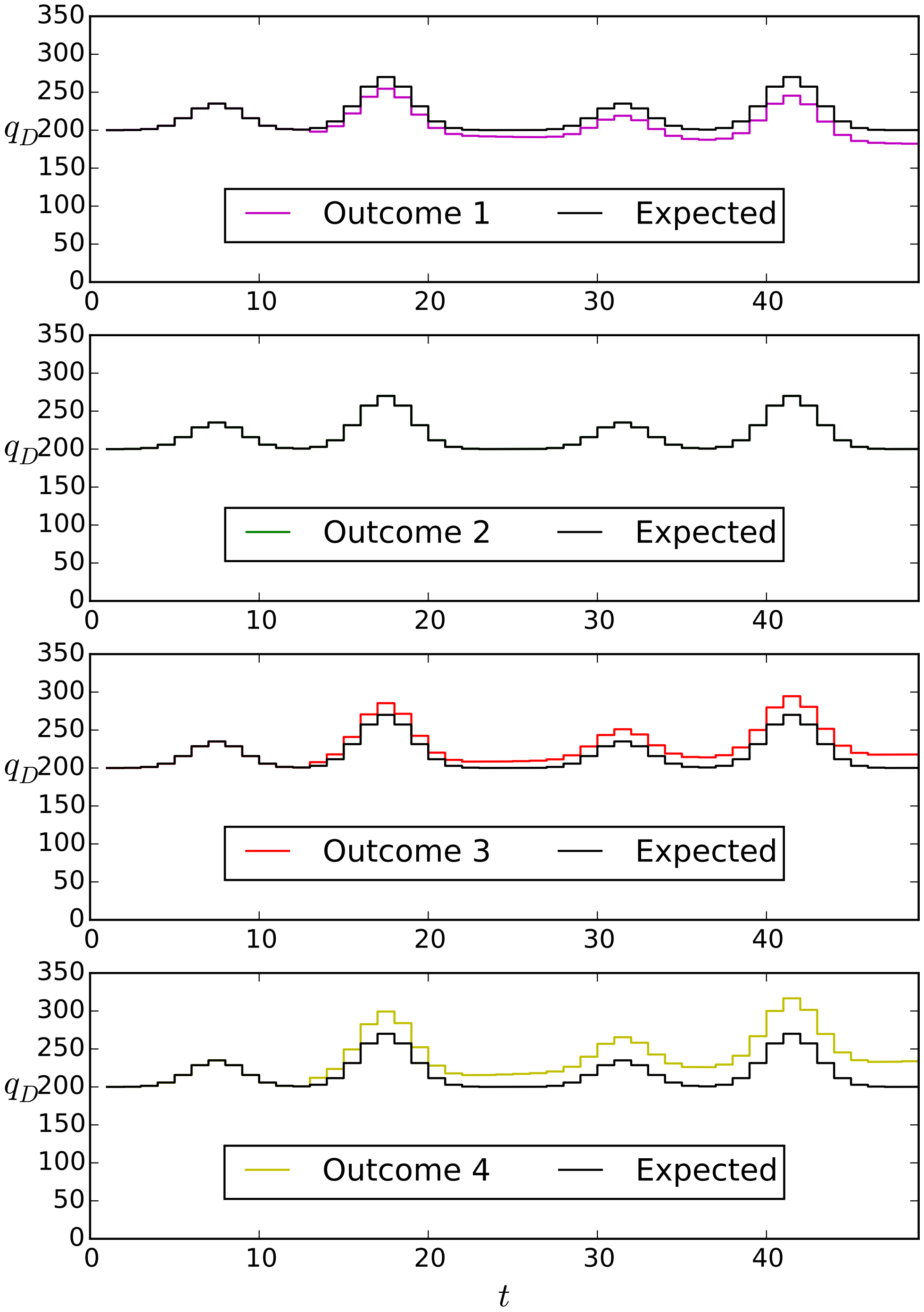


Figure 24: Outcomes for case with $\sigma_r = \frac{0.02}{6}$, which means large variation in building heating.

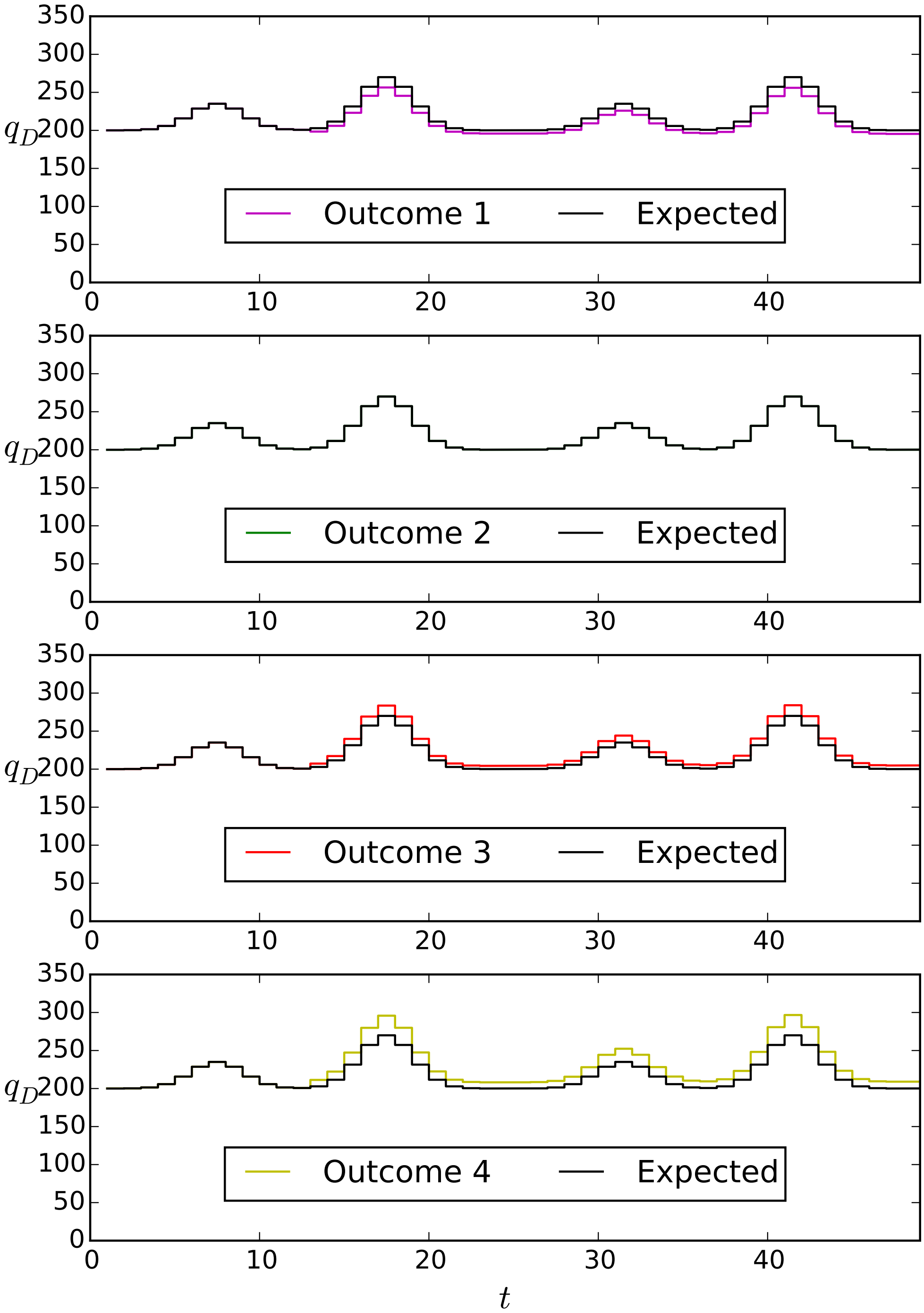


Figure 25: Outcomes for case with $\sigma_r = \frac{0.001}{6}$, which means minimal variation in building heating.

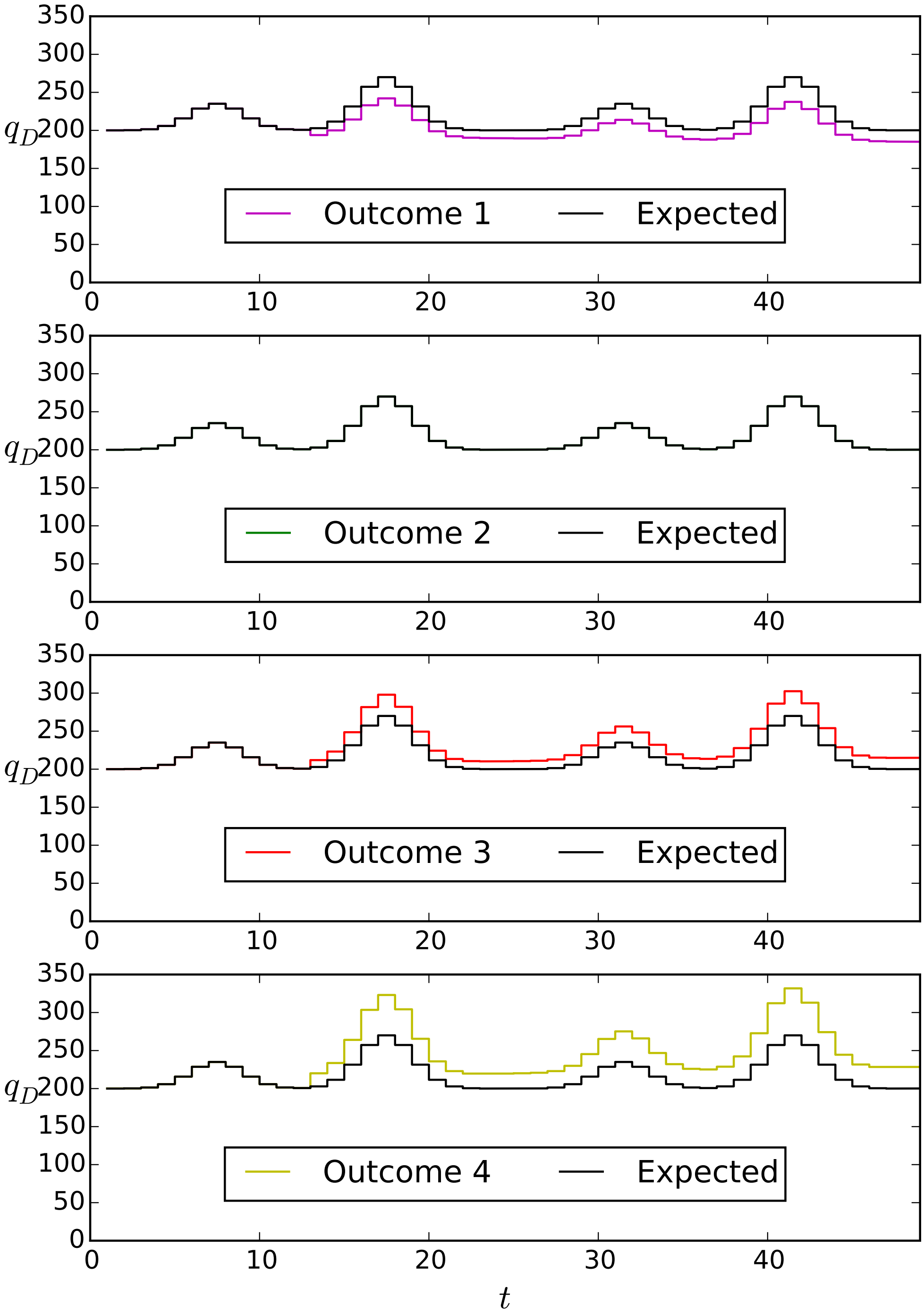


Figure 26: Outcomes for test with $\sigma_{q_{i,C}} = 0.4$, which means large variation in hot water demand.

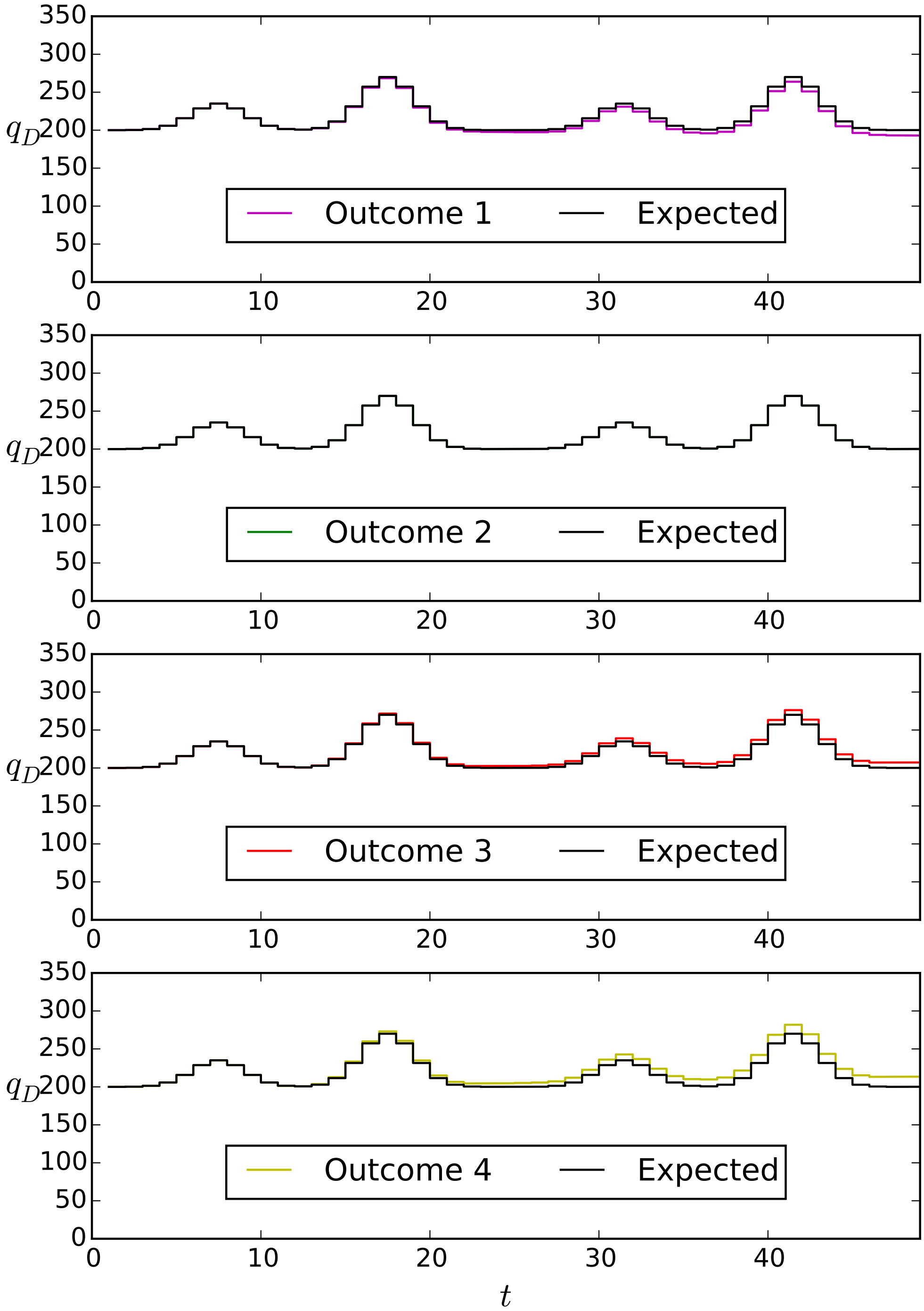


Figure 27: Outcomes for test with $\sigma_{q_{i,C}} = 0.01$, which means minimal variation in hot water demand.

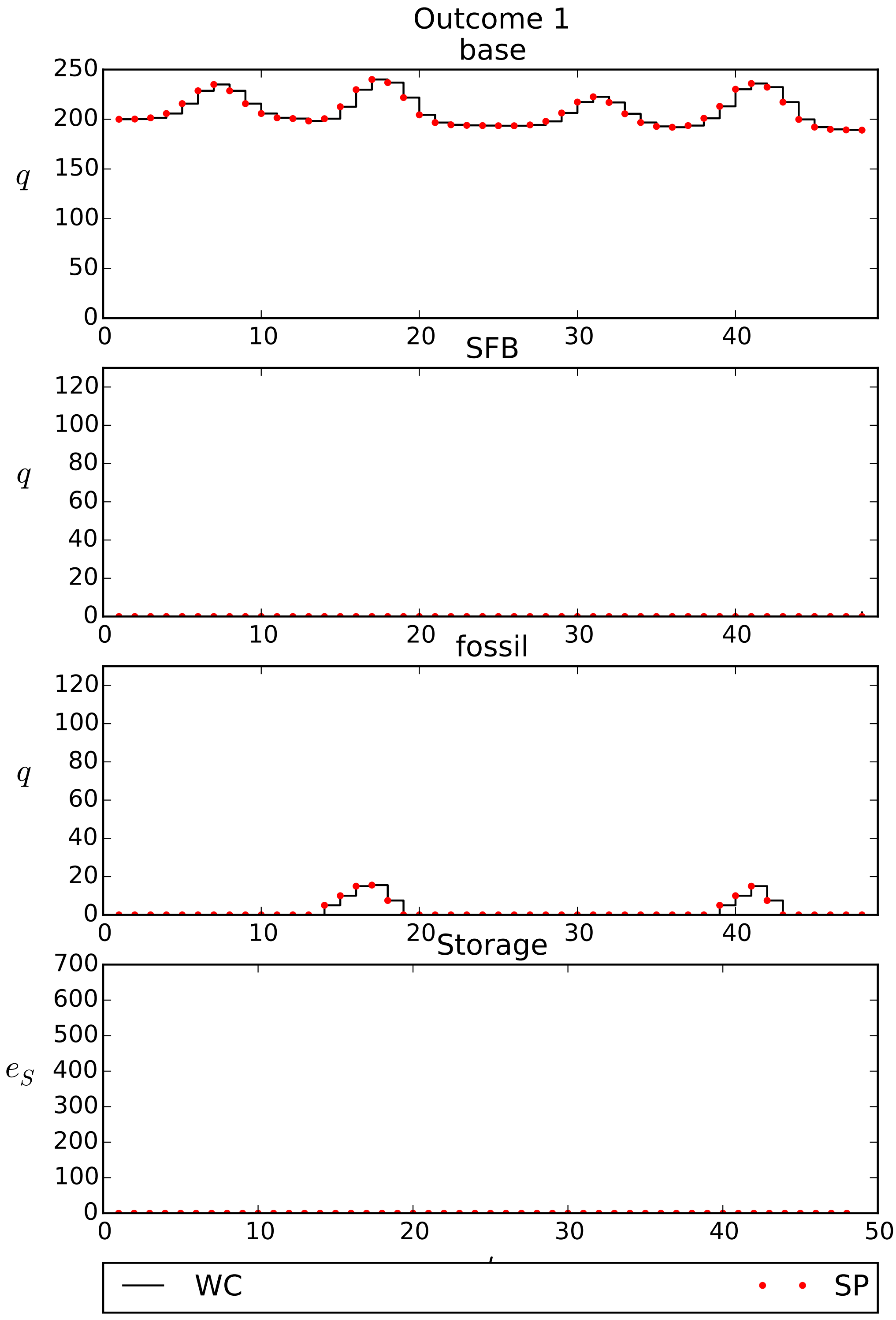


Figure 28: Comparing production plans with worst case optimization and stochastic programming approaches for Outcome 1, with default probability distribution. The two plans seems identical, but as Table 13 shows they are in fact not. For the worst case strategy there is a start up of the SFB at time $t = 48$, which is hardly visible in this figure and which does not occur when using stochastic programming.

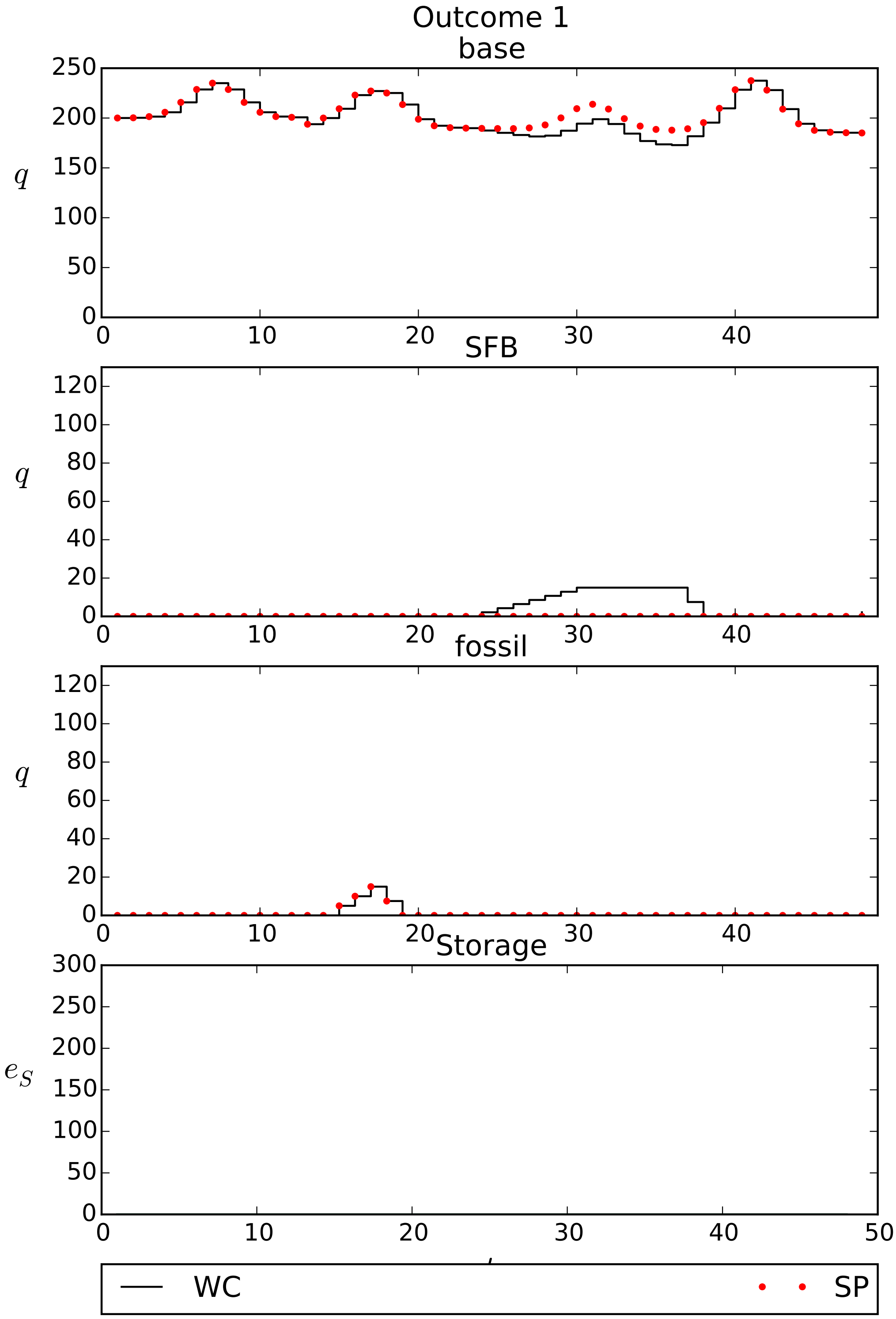


Figure 29: Comparing production plans resulting from stochastic programming and worst case optimization, for Outcome 1 with $\sigma_{q_{i,C}} = 0.4$. Worst case optimization here requires two startups of the solid fuel boiler, one at time $t = 24$ and one at time $t = 48$ (as in Figure 28, again hardly visible).

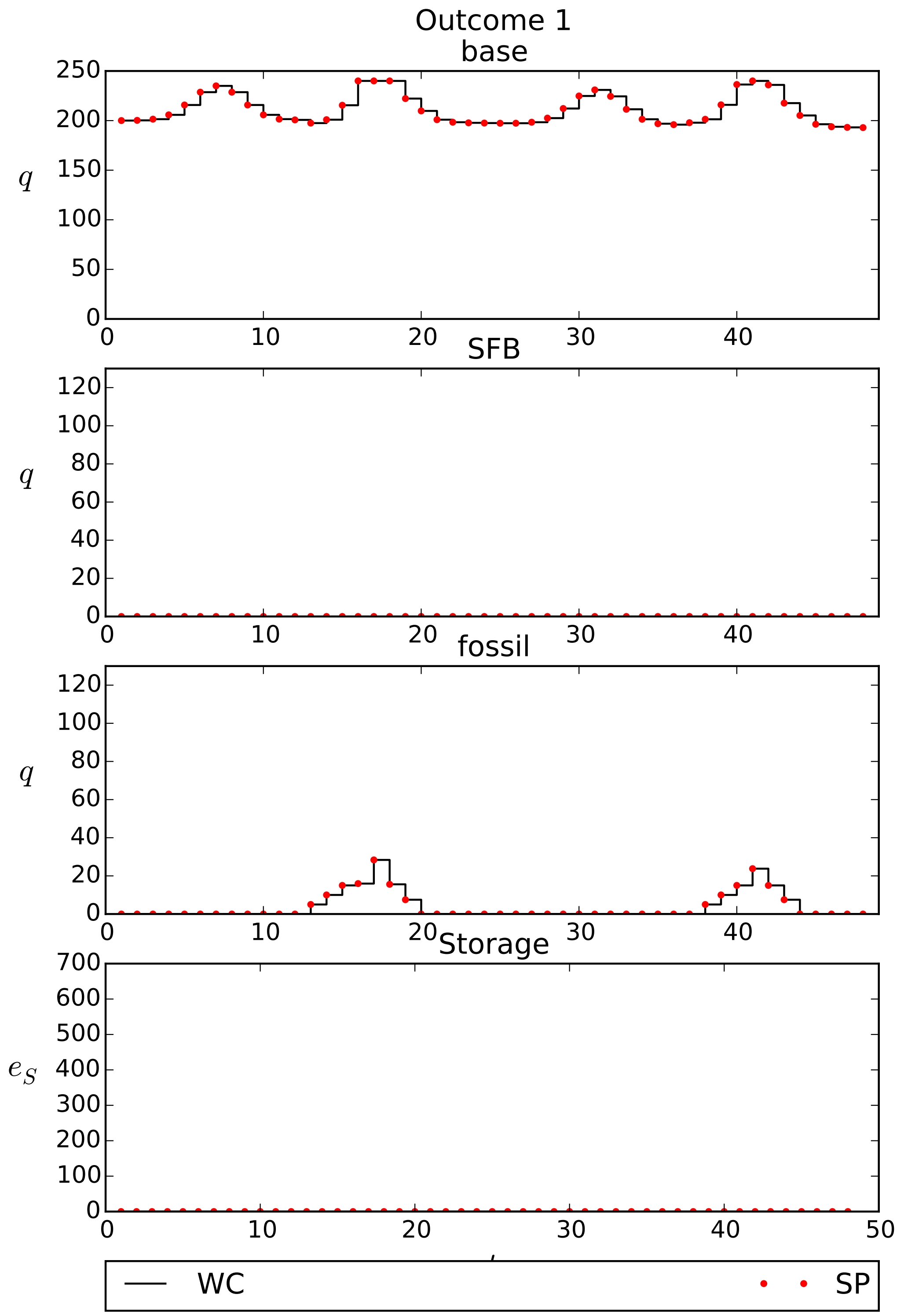


Figure 30: Comparing production plans resulting from stochastic programming and worst case optimization, for Outcome 1 with $\sigma_{q_{i,C}} = 0.01$. Here no startup is required of the solid fuel boiler (compare to Figure 28) and the production plans are identical.

## 7.6 Test 5: MPC with Accumulation

This test illustrates how stochastic programming also can have benefits for systems with accumulators. For this purpose, the set of production units is specified according to PU Case 2. The test is performed as Test 3, until the last iteration. The constraint on the accumulator storage is the one specified in (43). This puts a constraint on the storage at the end of each prediciton horizon, but since MPC is used with a period shorter than the prediction horizon, a drift effect may occur, and no constraint is put for the storage at time $t_{final}$. This means that it is hard to compare the strategies at time $t_{final}$ since this would require a way to take into account both the costs and the end time heat storage.

However, one straight forward way to evaluate the strategies is to assume that the demand for the time interval $[t_{final}+1, t_{final}-T-I_2]$ (in this example $[49, 66]$) follows the worst case scenario for the last iteration. Then a comparison of the benefits of two strategies can be accomplished by including their (Stage 2) production plans for the worst case scenario of the last iteration. This is what is done for this test.

The results of the tests are displayed in figures 31 - 34. A comparison of the costs for WC optimization and SP is shown in Table 14.

| | **WC Costs [k SEK]** | **SP Costs [k SEK]** | **Savings [%]** |
|---|---|---|---|
| Outcome 1 | 3085 | 3090 | -0.16 |
| Outcome 2 | 3067 | 3028 | 1.26 |
| Outcome 3 | 2949 | 2899 | 1.68 |
| Outcome 4 | 3275 | 3284 | -0.28 |
| Expected cost | 3069 | 3047 | 0.76 |

Table 14: Comparison of costs in test 5 for stochastic programming (SP) and worst case optimization (WC).

We see that for Outcome 1 and Outcome 4, the costs for SP and WC optimization are approximately the same, while for Outcome 2 and Outcome 3, savings can be made with stochastic programming. With the accumulator in use, it's slightly harder to analyse the reasons for different behaviour in SP and WC optimization, since the accumulated heat at a certain time point may differ between the strategies. For Outcome 2, it's interesting to notice the peak in the SFB soon after $t = 30$. From the bottom plot of Figure 32, we see that this is not driven to cover a high demand. Rather it is economical (with the current cost function) to store some heat before the unit is turned off.

It is not obvious what makes the WC strategy more expensive than the SP strategy. However, here is an attempt to explain the difference for Outcome 2. Considering the cost difference of 39000 SEK (see Table 14) and the startup cost of the SFB of 30000 SEK (see Table 2), we see that most of the cost difference is explained by the fact that the WC strategy starts up the SFB units twice, compared to one time for stochastic programming. The remaining 9000 SEK difference is assumed to come from the greater heat losses in the WC approach: for hours 30 to 60, the accumulator holds on average approximately 300 MWh extra heat. According to the loss model (13), this results in additional heat loads of approximately

$$300\,\mathrm{MWh} \cdot (1 - 0.005) \cdot (60 - 30)\,\mathrm{h} = 45\,\mathrm{MWh}. \tag{79}$$

Assuming that this heat has to be produced by the SFB unit, the heat loss costs are finally approximated to $45\,\mathrm{MWh} \cdot 300\,\mathrm{SEK/MWh} = 13500$ SEK, which is of the same magnitude as the 9000 SEK difference.

The observant reader may ask what is the cause of the single time point ($t = 37$) in Figure 33, where the Base Unit power is suddenly not run on full power. This issue illustrates the priorities of the optimization when selecting heat sources. The first choice is heat from the accumulator, under the constraint (43) for the minimum storage at the last time point of second stage. The second choice is production by the Base Unit, because of its low production cost. The third priority is to use one of the top up units. Based on this, the conclusion can be made that at the iteration starting at time $t = 37$, the average demand for the prediction horizon is so low that because of the heat stored in the accumulator, the Base Unit is not needed to run at full power in order to cover the worst case scenario demand. This conclusion is supported by Figure 6 where the demand for Outcome 3, time $t = 37$ and forward, is significantly below the expected demand curve for this time period.

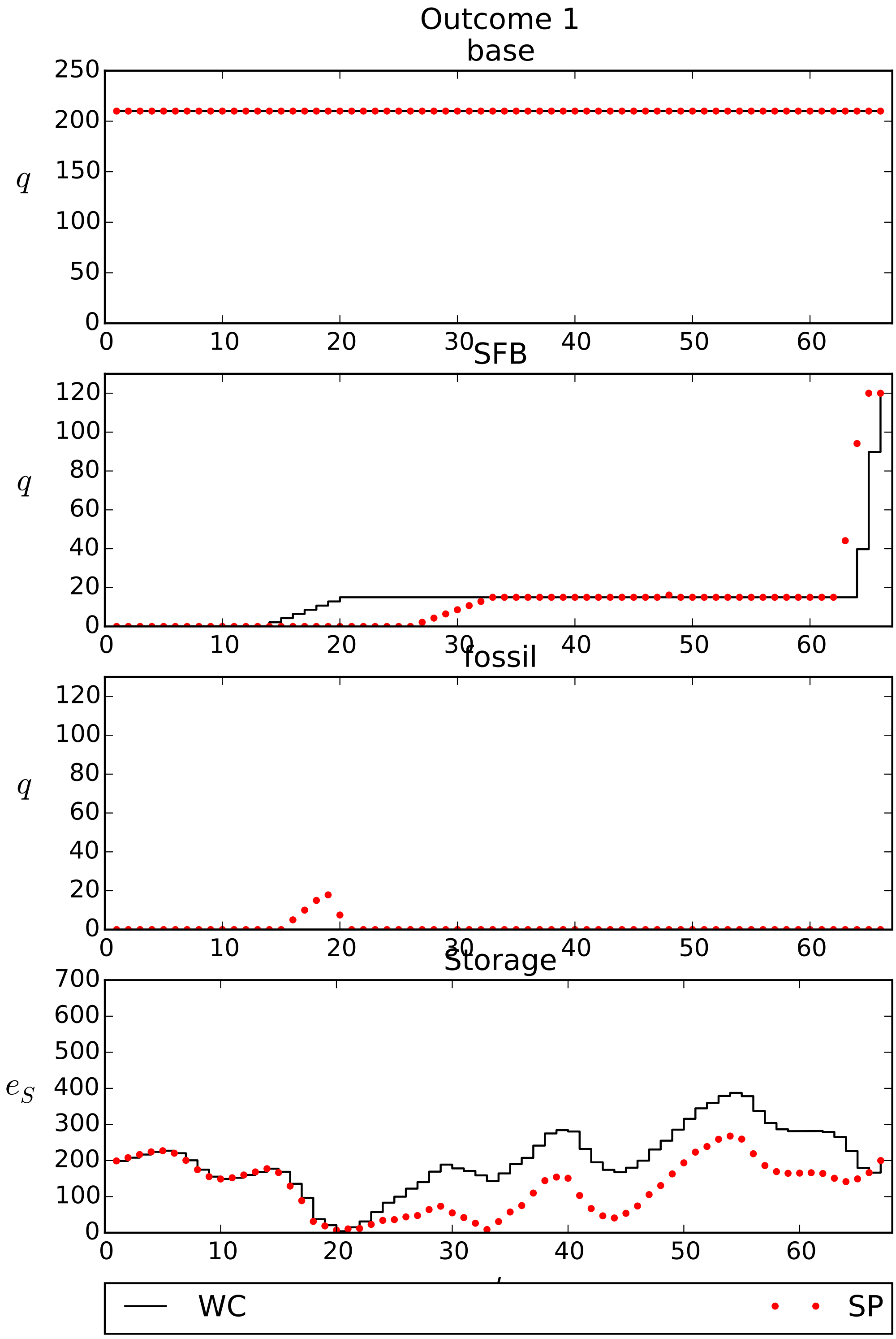


Figure 31: Comparing results for worst case optimization and stochastic programming in Test 5 and Outcome 1.

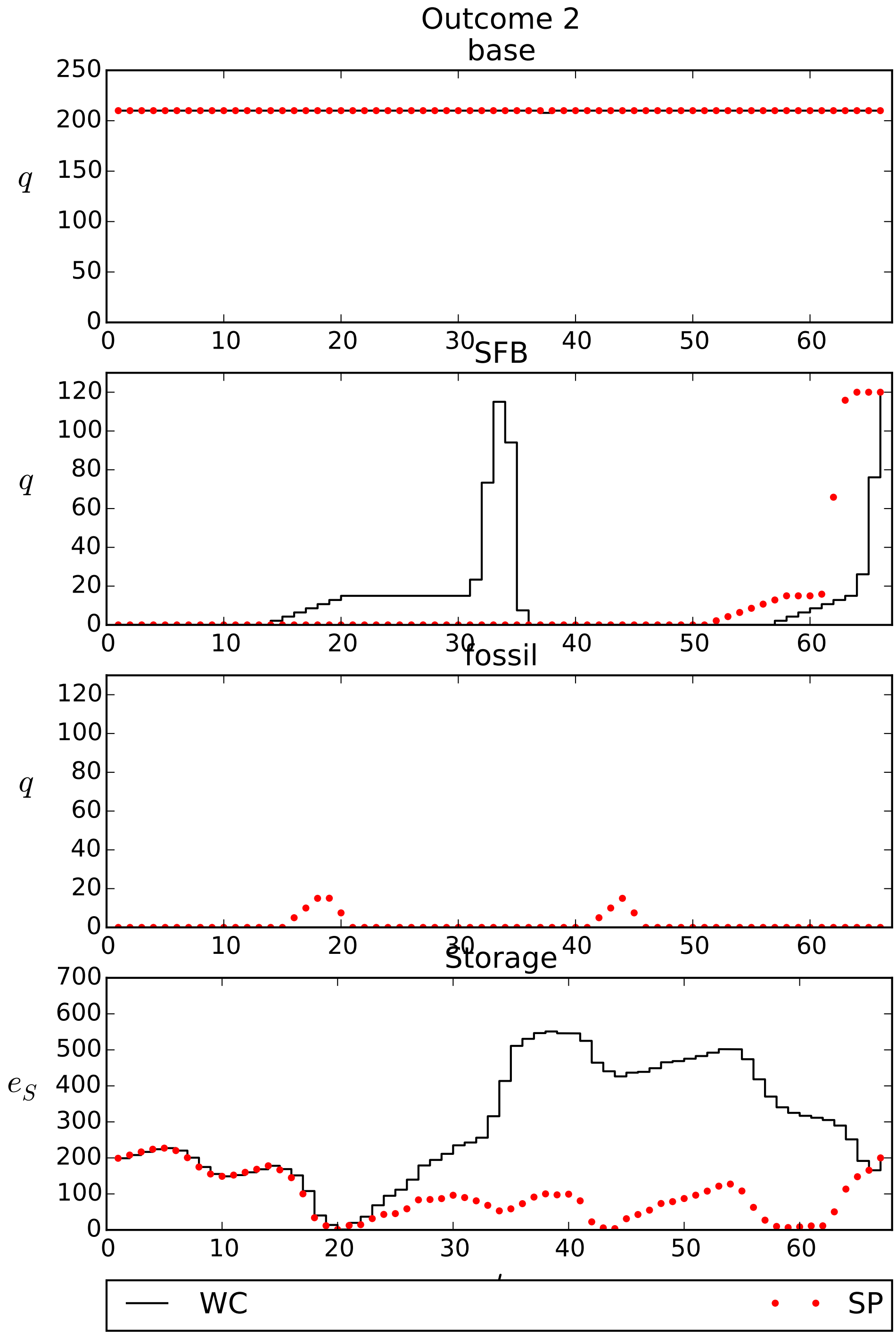


Figure 32: Comparing results for worst case optimization and stochastic programming in Test 5 and Outcome 2. Notice the peak for the solid fuel boiler using worst case optimization, at time $t = 33$. This has the purpose to fill up the accumulator before the solid fuel boiler is turned off. Also notice that the worst case optimization has higher costs than with stochastic programming, mainly because the solid fuel boiler is turned on twice, compared to once with stochastic programming.

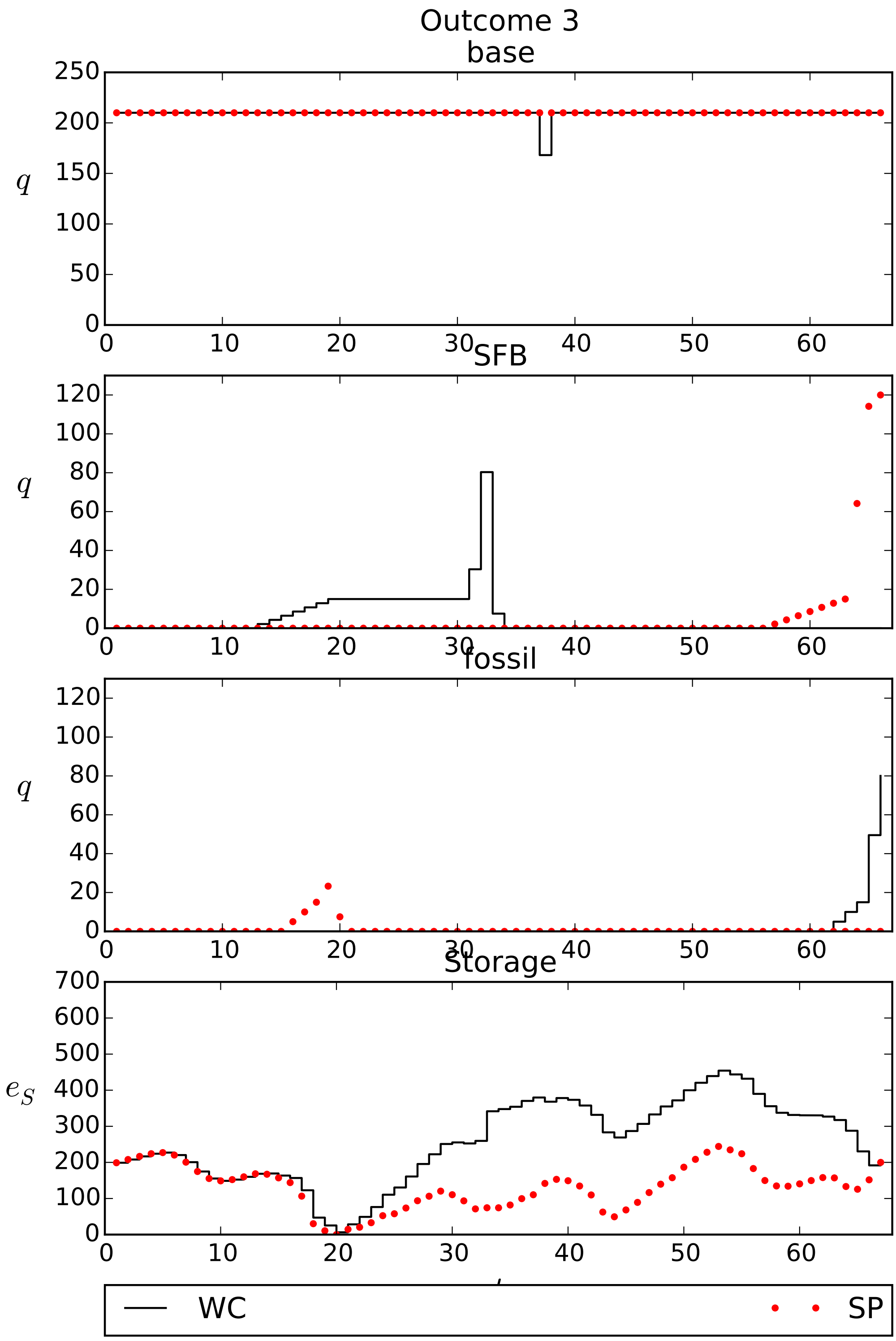


Figure 33: Comparing results for worst case optimization and stochastic programming in Test 5 and Outcome 3. Notice how the base unit power is lowered at time $t = 37$, in order to avoid excess heat in the accumulator at the end time.

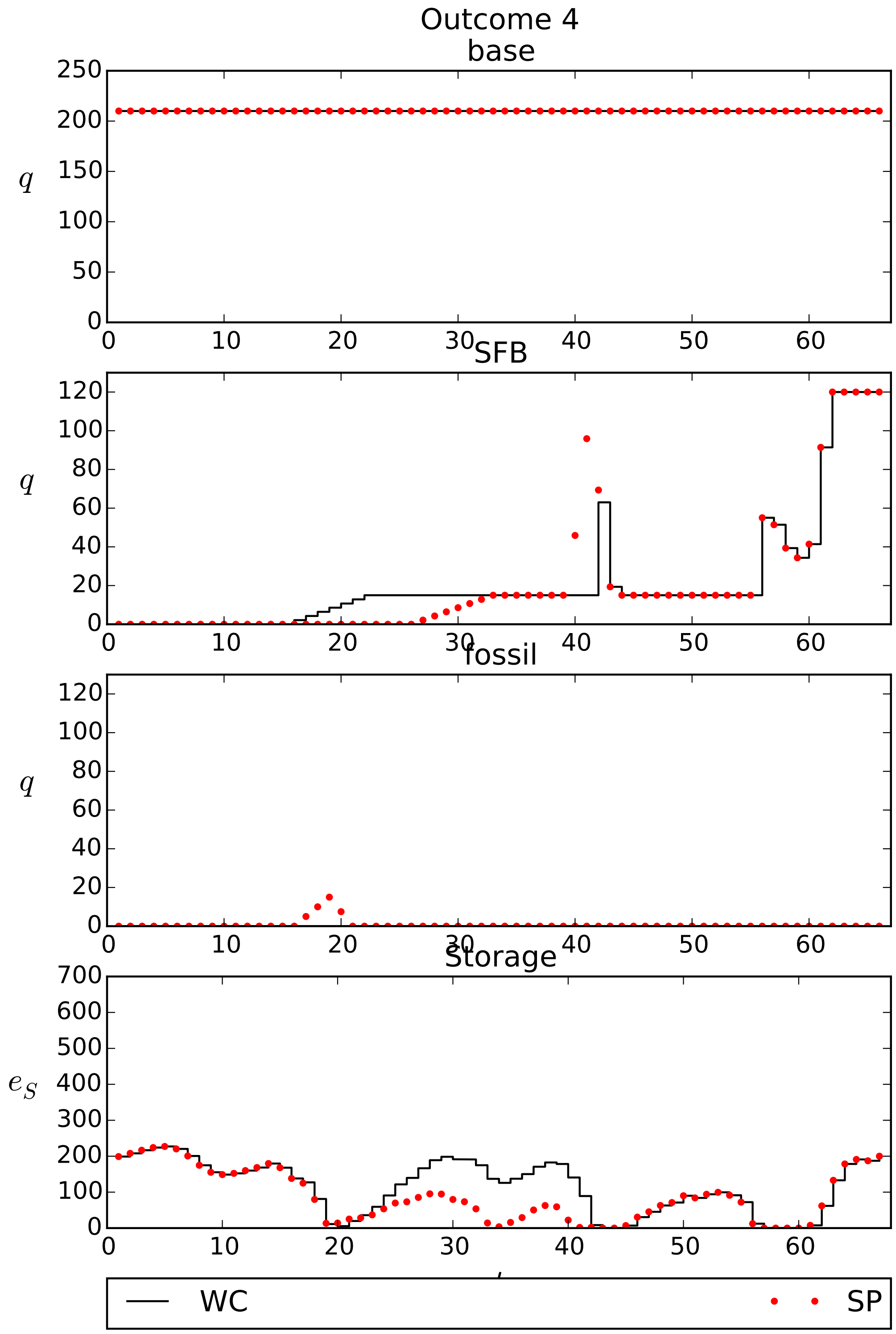


Figure 34: Comparing results for worst case optimization and stochastic programming in Test 5 and Outcome 4.

# 8 Discussion

This section discusses the interpretation of the results, an evaluation of the benefits of stochastic programming, the assumptions made and finally which steps to take in further work.

## 8.1 Interpretation of Results

Test 1 shows two simple examples where stochastic programming can give a production plan with lower costs than the worst case approach. In Test 1.a there is a need of starting up a top up unit, and the strategies give different priorities to the two different top up units. In Test 1.b, the SP plan shuts down the SFB, while the WC approach keeps it running. Test 2 illustrates the benefits of adding an accumulator to the system.

Test 3 then exemplifies the use of model predictive control, at each iteration performing a two stage stochastic programming problem. The outcomes are randomly generated, which adds to the realism of the test. In particular it means that the outcome demand $q_D$ is not in the set of scenarios $\{q_D(\xi) : \xi \in \Xi\}$. However, as in test 1, still twelve hours of perfect information are considered at each iteration. The test again shows savings of stochastic programming, having the same order of magnitude as in Test 1, see Table 10. Increasing the prediction horizon in Test 3.b increases the benefits of stochastic programming, since it in a sense makes the WC approach more extreme, by forcing it to plan for a demand increase lasting over two days.

The sensitivity analysis shows how changes in demand probability distribution affect the benefits of stochastic programming. We see in Table 13 that lowering the variation of either hot water demand or building heating demand to a minimum significantly lowers the savings from stochastic programming. However, even for $\sigma_{q_{i,C}} = 0.01$, the SP savings for Outcome 3 are above one percent. If the demand variation is instead increased, the savings increase as well.

Test 4 shows the impact of delays in lowering the required production peaks of a district heating network. For systems where customer delays or relative customer loads are assumed to vary significantly over time, this result indicates the need of working with a net model in order to for the optimization model to be accurate. As mentioned in 4.1.3, a net model should preferably be used together with demand predictions for the individual customers, rather than for the production unit.

Finally, Test 5 shows that stochastic programming can also have benefits for a net with accumulator, although the expected savings in the example are lower (see Table 14).

## 8.2 Evaluation of Stochastic Programming Benefits and Suggestions for Further Work

When evaluating the use of stochastic programming in a wider sense, there are many aspects to consider. Firstly, it should be noted that the tests where constructed in order to give different results for the two strategies. The savings in these cases are based on having a competition between different production units to provide the customer demand. This presumes having different production units with differing properties, such that there is no obvious hierarchy for which unit should be started up first. It also presumes that the demand varies in a manner where it can be profitable to turn production units on and off. Since the demands are strongly dependent on season (see Figure 2), there may be certain seasons where this is the case and other seasons where e.g. the variations can be covered within the production span of a single base unit.

Although the aforementioned aspects point out that the results in this project where achieved for particular situations where stochastic programming has benefits, there are also different factors which could increase the benefits of stochastic programming. One such feature that has not been considered in this project is the use of chance constraints, see (Shapiro et al., 2009, p. 5). A chance constraint is not a hard constraint that must never be violated, but it is a constraint that's violated with a limited probability. For this unit commitment problem, it would make sense to formulate the demand constraint as such a chance constraint, since a slight violation may still be acceptable. Especially, since there may be ways to cover the customer demands which do not require the optimization constraints to hold (e.g. increasing the mass flow in the network can give fast changes in delivered heat).

Another feature that applies for networks with combined heat and power (CHP) production units is the possibility to also regard the electricity price as a random variable, cf. (Larsson et al., 2014, p. 76).

An essential need to be met in order to start applying the stochastic programming methods on practical problems, is the generation of scenarios. This has been done here in an ad hoc way, for the purpose of testing, but for real problems it would require methods that start from actual data, such as the ones described in (Feng et al., 2015).

Besides changes in optimization method, the model could also be improved. One weakness of the current implementation is the net model and its effect on the demand. This is an area where much could be done. A first step could be to work with demand predictions for individual customers, rather than directly for the production sites. This would improve the detail of the model and could enable the situation mentioned in Section 2.2, where different production sites can compete for the customers. However, it is not obvious how to write the demand constraint with several production sites, since the delays then depend on which producer is coupled to which consumer. A second step could be to somehow exchange the heat energy decision variable with variables for return temperature and mass flow, in order to model time varying delays.

A more minor issue is that the current formulation of the MPC simulation is quite limited, since it assumes perfect information for twelve hours forward. A further development could be to work with a two stage formulation, where the demand of stage one is set as expected demand, but where the actual demand differs from the expected. The problem then is how to model the effect on the system when the actual demand doesn't match the delivered heat.

# 9 Summary

The goal of this project was to develop robust strategies for production planning of district heating networks, in particular using stochastic programming. The first step towards this was to put together an optimization model which handles the different properties of a district heating network: demand constraints, startup and shutdown constraints, accumulator constraints and initial conditions.

A second step was to formulate the stochastic programming problem, as well as the worst case optimization. An important issue connected to this was the generation of scenarios, which in this project was done based on constructing a stochastic demand model, specifying the demand probability distribution.

The first tests evaluated the use of a two stage stochastic programming formulation, by comparing the results to those of worst case optimization. Test cases were constructed where the strategies differed, both for choosing which production unit should be started up and also for when to shut down a production unit.

In order to make use of the available updates in weather predictions, an MPC-method was proposed, solving a two stage stochastic programming problem at each stage. A method was also constructed for randomly generating scenarios, in order to test the method for different shapes of the demand curve.

Test cases where constructed, with two different sets of production units, in order to test the MPC method for systems both with and without accumulation. The test was also performed with different prediction horizons and a sensitivity analysis investigated the effect of different demand probability distributions.

Comparing the stochastic programming approach to worst case optimization, the tests show that stochastic programming can give lower expected costs, with robustness maintained.

# 10 Bibliography


Algebraic modeling language — wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Algebraic_modeling_language`. Accessed: 2015-07-03.

Gurobi - mixed integer programming basics. `http://www.gurobi.com/resources/getting-started/mip-basics`. Accessed: 2015-07-01.

Gurobi home page. `http://www.gurobi.com/index`. Accessed: 2015-07-03.

Pyomo documentation. `https://software.sandia.gov/downloads/pub/pyomo/PyomoOnlineDocs.html`. Accessed: 2015-07-03.

Pyomo web page. `http://www.pyomo.org/impact`. Accessed: 2015-09-02.

Arroyo, J. and Conejo, A. (2004). Modeling of start-up and shut-down power trajectories of thermal units. *Power Systems, IEEE Transactions on*, 19(3):1562–1568.

Birge, J. R. and Louveaux, F. (2011). *Introduction to stochastic programming.* Springer-Verlag New York.

Boyd, S. (2014). Stochastic programming. Lecture Slides for course in Convex Optimization.

Dotzauer, E. (2001). *Energy system operation by Lagrangian relaxation.* Univ., Linköping. Diss. Linköping : Univ., 2001.

Energimyndigheten (2015). Trädbränsle- och torvpriser. SVERIGES OFFICIELLA STATISTIK, Prisblad.

Feng, Y., Rios, I., Ryan, S., Spürkel, K., Watson, J.-P., Wets, R.-B., and Woodruff, D. (2015). Toward scalable stochastic unit commitment. part 1: load scenario generation. *Energy Systems*, pages 1–21.

Frederiksen, S. and Werner, S. (2013). *District Heating and Cooling.* Studentlitteratur AB, Lund.

Gadd, H. and Werner, S. (2013). Heat load patterns in district heating substations. *Applied Energy*, 108(0):176 – 183.

Häggståhl, D., Dotzauer, E., and för värmeteknisk forskning, S. (2004). *Produktionsplanering under osäkerhet - simulatorbaserad produktionsplanering av medelstora kraftvärmeanläggningar.* Värmeforsk.

Hart, W. E., Laird, C., Watson, J.-P., and Woodruff, D. L. (2012). *Pyomo - Optimization Modeling in Python.* Springer-Verlag New York.

Kvarnström, J., Dotzauer, E., Gollvik, L., and Andersson, C. (2007). *Lastprognoser för fjärrvärme.* Värmeforsk.

Larsson, P.-O., Runvik, H., Velut, S., Razavid, S. M., Bohlin, M., and Funkquist, J. (2014). Decision support for short-term production planning of district heating using non-linear programming. Technical Report 1262, Värmeforsk.

Maciejowski, J. (2002). *Predictive Control: With Constraints*. Pearson Education. Prentice Hall.

Ruiz, P., Philbrick, C., Zak, E., Cheung, K., and Sauer, P. (2009). Uncertainty management in the unit commitment problem. *Power Systems, IEEE Transactions on*, 24(2):642–651.

Sandou, G., Font, S., Tebbani, S., Hiret, A., and Mondon, C. (2005). Predictive control of a complex district heating network. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 7372–7377.

Shapiro, A., Dentcheva, D., and Ruszczyński, A. (2009). *Lectures on Stochastic Programming*. Society for Industrial and Applied Mathematics.

Shiina, T. and Birge, J. R. (2004). Stochastic unit commitment problem. *International Transactions in Operational Research*, 11(1):19–32.

# 11 Appendix

For the two stage formulations in Sections 4.2 and 4.3, the following notation is used to gather the decision variables of Stage 1 and Stage 2 into two separate vectors:

$$x_1 = \begin{bmatrix} q_{1-UD_{max}} \\ \vdots \\ q_{I_1} \\ u_{1-UD_{max}} \\ \vdots \\ u_{I_1} \\ q_{1-UD_{max},S} \\ \vdots \\ q_{I_1,S} \\ y_{1-UD_{max}} \\ \vdots \\ y_{I_1} \\ z_{1-UD_{max}} \\ \vdots \\ z_{I_1} \\ e_{1-UD_{max},S} \\ \vdots \\ e_{I_1,S} \end{bmatrix} \qquad x_2 = \begin{bmatrix} q_{I_1+1} \\ \vdots \\ q_{I_2+DD_{max}} \\ u_{I_1+1} \\ \vdots \\ u_{I_2+DD_{max}} \\ q_{I_1+1,S} \\ \vdots \\ q_{I_2+DD_{max},S} \\ y_{I_1+1} \\ \vdots \\ y_{I_2+DD_{max}} \\ z_{I_1+1} \\ \vdots \\ z_{I_2+DD_{max}} \\ e_{I_1+1,S} \\ \vdots \\ e_{I_2+DD_{max},S} \end{bmatrix} \tag{80}$$

where $q_i$, $u_i$, $y_i$ and $z_i$ themselves are vectors indexed by production units, e.g.

$$q_i = \begin{bmatrix} q_{i,1} \\ q_{i,2} \\ \vdots \\ q_{i,k} \\ \vdots \end{bmatrix}, \qquad k \in K. \tag{81}$$

Master's Theses in Mathematical Sciences 2015:E43
ISSN 1404-6342

LUTFMA-3283-2015

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/