

PRODUCTION OPTIMIZATION FOR DISTRICT HEATING

Short-term planning of district heating grid in Gävle, Sweden

KARL BROGREN NICOLAS LINDGREN

The School of Business, Society and Engineering

Course: Degree project in Sustainable Energy Systems Course code: ERA403 Subject: Energy systems Points: 30 hp Program: Master of Science in Engineering -Energy System Handledare: Jan Skvaril Examinator: Anders Avelin Uppdragsgivare: Håkan Fjäder, EnviLoop AB Datum: 2019-06-07 E-post: kbn14005@student.mdh.se nln13004@student.mdh.se

ABSTRACT

Energy systems with a high portion of renewable energy from wind and solar power can suffer from fluctuations in production due to weak winds or cloudy weather, which may affect the electricity price. When producing heat and power in a combined heat and power plant, an additional heat storage tank can be used to store the heat surplus which is obtained when the power production is high, and the heat demand is low. To optimize heat and power production economically, short-term planning can be applied. Short-term planning covers the production in the near future of 1-3 days. The optimization in this degree project is based on the district heating production, which means that the heating demand always needs to be fulfilled. The district heating production is based on the weather. Therefore a suitable period for simulation is three days due to the accuracy of the weather forecasts are reasonable. The optimization is performed on the district heat system in Gävle, Sweden. The system comprises several different production units, such as combined heat and power plants, backup plants, and industrial waste heat recovery. Two different models are made, one using linear programming and one using mixed integer non-linear programming. The model stated as a linear programming problem is not as accurate as of the one stated as a mixed integer non-linear programming problem which uses binary variables. Historical input data from Bomhus Energi AB, a company owned together by the local heat and power supplier Gävle Energi AB and the pulp and paper manufacturer BillerudKorsnäs AB, was given to simulate different scenarios. The different scenarios have various average temperatures and in some scenarios are there some issues with the pulp and paper industry affecting the waste heat recovery. In all scenarios is the heat storage tank charged when the demand is low and then discharged when the demand increases to avoid starting some of the more expensive backup plants if possible. The simulation time varies a lot between the two approaches, from a couple of seconds to several hours. Particularly when observing scenarios with a rather high demand since the backup generators use binary variables which take a lot of time to solve.

Keywords: Short-term planning, Optimization, District heating, Linear programming, Mixed integer non-linear programming, Branch and bound, Heat storage, MATLAB, TOMLAB

PREFACE

We want to thank our supervisor Jan Skvaril at Mälardalens Högskola, for providing us with guidance throughout the work. We would also like to express our happiness for making this project possible as our degree project, thank you EnviLoop AB and Bomhus Energi AB. Inspiration for the project has been Erik Dotzauer and the work he has done in the area of study. Therefore we would like to thank him.

Västerås May 2019

Karl Brogren

Nicolas Lindgren



MÄLARDALENS HÖGSKOLA ESKILSTUNA VÄSTERÅS



SUMMARY

Renewable energy systems such as wind and solar power may affect fluctuations in the power grid caused by weak winds and cloudy weather. The fluctuations in the power grid can affect the electricity price. Producers at combined heat and power plants want to produce electricity when the price is high to increase the revenues. When producing electricity, a significant amount of heat is produced as a by-product. The heat is often inserted to a district heating grid. If the amount of produced heat exceeds the heating demand, the heat surplus must be stored to remain a high efficiency.

The purpose of this degree project is to develop a process model that can optimize how the production units should operate and how the heat storage tank should be charged and discharged to provide the best economic outcome. The model is based on a prognosticated heat demand, which must be fulfilled.

To get a better understanding of the problem as well as finding suitable approaches to solve the problem is a comprehensive literature study made. Most of the information is gathered from a similar research project, scientific papers, and books related to the field. The primary databases used to find scientific papers and research projects are Google Scholar, Research Gate, and Science Direct. A study visit to the different facilities in Gävle is made to gain a better understanding of the production units and the distribution system. To get relevant input data about the various production units, a meeting with the process engineers at BillerudKorsnäs AB, Bomhus Energi AB, and Gävle Energi AB is held. The input data have a central role in the development of the model.

The model is created in MATLAB. It runs the main script, which calls on several functions related to the production units and the optimization. Since the problem is rather complex, several different solvers are used in the optimization function. Which solver to use is highly dependent on the problem formulation. If the problem is stated as a linear programming (LP) problem, a more straightforward solver can be used, which reduces the accuracy of the model. If the problem, on the other hand, is stated as a mixed integer non-linear programming (MINLP) problem, the model can be more accurate but demands a more advanced solver.

To validate the model, a sensitivity analysis is made. The sensitivity analysis is performed by changing the input data and observe the result to control if it behaves in a correct and anticipated way. The different input data are related to different scenarios. The scenarios vary in heat demand and production status of the pulp and paper industry.

The optimized production process uses cheaper production units as a base load, and when the heat demand is low, the heat storage tank charges. When the heat demand is increased, the tank discharges to avoid starting the more expensive backup plants if possible. When comparing the simulation with and without heat storage, the simulation, including heat storage has a lower cost.

When the model is compared to the previous model used at Bomhus Energi AB, the result shows that for a specific scenario, the savings are around 400 000 SEK, both for the LP-model and the MINLP-model.

It is primarily the use of the backup units that differs between the linear and the non-linear approach. Which make the result of the linear and the non-linear model the same when the heating demand is low. Since the non-linear problem can be solved as a linear problem. The variation in run time for the simulation is rather significant between the two approaches, specifically when the demand is high. It can range from a couple of seconds for the linear solver to above 1 000 seconds for the non-linear solver.

The work concludes that the linear model is a more practical tool for the customer while being a bit less realistic when it comes to the result. The new model is an improvement from the old and helps to avoid unnecessary costs.

TABLE OF CONTENT

1	IN'	IRODUCTION7
	1.1	Gävle Energi AB 7
	1.2	Bomhus Energi AB 8
	1.3	Purpose10
	1.4	Research questions10
	1.5	Delimitations11
	1.6	Contribution to current research11
2	ME	THOD12
	2.1	Gathering of data12
	2.2	Modeling and simulation12
	2.3	Validation of result13
3	LI	ERATURE STUDY14
	3.1	Thermal storage14
	3.2	Optimization of a CHP-plant with a heat storage unit16
	3.3	Modeling of non-linear efficiencies18
	3.4	Non-convex & non-continuous problems19
	3.5	Branch and bound optimization20
4	М	DDELING23
	4.1	Existing model23
	4.2	Current model23
	4.3	Matlab24

	4	.3.1	Linprog24	ŀ
	4	.3.2	TOMLAB	ł
		4.3.2.	1. MinlpBB	1
	4	.4.1	Linear programming25	5
	4	.4.2	Mixed-integer non-linear programming28	3
	4.5	Scer	narios)
5	RE	SULT	۲31	I
	5.1	Line	ar vs. Nonlinear	}
	5.2	Heat	ing Demand35	;
	5.3	Elect	tricity Price	;
	5.4	Stora	age vs. No storage37	,
	5.5	Bom	hus Energi model vs. Minlp model39)
6	DI	SCUS	SION41	ł
7	CC	ONCL	USION	ł
8	FL	JRTHE	ER WORK45	;
R	EFEI	RENC	E LIST46	5
A	PPE	NDIX ⁻	1. LINEAR PROGRAMMING50)
A	PPE	NDIX	2. MIXED INTEGER NONLINEAR PROGRAMMING55	5
A	PPE	NDIX	3. NONLINEAR CONSTRAINTS (MINLP)63	\$
A	PPE		4. OBJECTIVE FUNCTION (MINLP)64	ŀ
A	PPE		5. BOMHUS VS MINLP65	5

INDEX OF FIGURES

Figure 1 Black liquor evaporator with three effect stages
Figure 2 Simplified block scheme of steam production for the district heating grid and
industries10
Figure 3 Flowchart of the simulation process13
Figure 4 Load shedding of a district heating system (Frederiksen & Werner, 2014)15
Figure 5 Pressureless heat storage tank connected to a district heating grid. (Frederiksen &
Werner, 2014)16
Figure 6 Example of convex sub-models, 9 points (Makkonen & Lahdelma, 2005)20
Figure 7 Enumeration tree related to the example. (Vanderbei, 2014)21
Figure 8 Minlp result for scenario I & II
Figure 9 Minlp/lp result for scenario III, IV & V
Figure 10 Comparison of scenario I minlp and lp result
Figure 11 Comparison of scenario II minlp and lp result
Figure 12 In/Decrease heat demand for scenario I
Figure 13 In/Decrease electricity price for scenario I
Figure 14 Comparison of production costs for different production units at different
electricity prices
Figure 15 Scenario III, storage vs. no storage
Figure 16 Previous vs. Minlp model
Figure 17 Total production cost comparison between the Bomhus model, the Minlp model,
and the Lp model40
Figure 18 Previous vs. Minlp model

INDEX OF TABLES

Table 1 Example of efficiencies at different loads	18
Table 2 Cost comparison scenario I different electricity prices	37
Table 3 Scenario III, storage vs. no storage	38
Table 4 Total production, Bomhus vs. Minlp (referring to Figure 16)	40

NOMENCLATURE

Denotation	Description	Unit
Р	Power	Watt [W]
Q	Heat	Watt [W]
t	Temperature	Celsius [°C]
η	Efficiency	Percent [%]
i	Timestep	Hour [h]

ABBREVIATIONS

Abbreviation	Description
BEAB	Bomhus Energi AB
СНР	Combined heat and power
Evap	Evaporator
FGC	Flue gas condenser
GEAB	Gävle Energi AB
HWB	Hot water boiler
HWC	Hot water condenser
IP	Integer programming
J	Johannes combined heat and power plant
LP	Linear programming
MILP	Mixed integer linear programming
MINLP	Mixed integer non-linear programming
MIP	Mixed integer programming
WH	Waste heat from the recovery boilers

DEFINITIONS

Definition	Description
Α	Inequality constraints matrix
A_{eq}	Equality constraints matrix
В	Inequality constraints vector
B _{eq}	Equality constraints vector
Black liquor	Black liquor is a waste product from digesting pulpwood, from the process of producing paper pulp.
f ^T	Transposed objective
Kraftliner	Paperboard made from virgin pulp
lb	Lower bound vector
Load shedding	Store heat when the demand is low, utilize stored heat through discharging when the demand is high.
Scenario I	Big operation problems with the industry, going into a period with cold weather.
Scenario II	Big operation problems with the industry, going into a period with warmer weather.
Scenario III	Regular operation, significant variations in temperature during night and day. Temperatures are below o°.
Scenario IV	Regular operation, warm weather.
Scenario V	Regular operation, significant variation in temperature during night and day. Temperatures are above 0°.
SOS	Special ordered set,
Thermal stratification	Keep the stored water layered by temperature, warm water in the top and cold water in the bottom.
ub	Upper bound vector
X	Decision variable

1 INTRODUCTION

The rising awareness about human-caused emissions and its effects on the environment and human health has led to investment towards a more environmentally friendly energy system (Karaoğlan & Durukan, 2016).

Energy systems with a high portion of renewable energy from wind and solar power can suffer from fluctuations due to cloudy weather or weak winds. The variations also affect the electricity price. (Pappala, Erlich, Rohrig, & Dobschinski, 2009; Garcia-Gonzales, Moraga.R, Santos.M, & Gonzales.M, 2008)

To eliminate the fluctuations in the grid, combined heat, and power (CHP) plants, together with a heat storage unit, is a practical solution. The heat and power production of a CHP-plant can easily be regulated to match the fluctuations in the grid. (Mueller, Tuth, Fischer, Wille-Haussmann, & Wittwer, 2013)

A high electricity production provides a high production of heat that could exceed the heat demand. With a heat storage unit, the heat surplus can be stored, and therefore, also maintain a high total efficiency, which characterizes a CHP-plant (Abdollahi, Wang, Rinne, & Lahdelma, 2014). CHP-plants can use biofuels to be more environmentally friendly than it would be using fossil-based fuels. A deregulated electricity market, which is existing in Sweden, makes the electricity production competitive and leads to a more volatile price (Swedish Competition Authority, 1996). Due to this, dynamical decision tools are essential to cover the heat demand while maximizing the profits.

Today most of the long-term contracts have been replaced by short-term contracts. The short-term contracts put certain requirements on the production, which must be optimized with high precision to be able to meet the varying electricity price while covering the heating demand (Energimarknadsinspektionen, 2015). To get the future heating demands, prognostication of the weather must be made since the demand for heat is entirely dependent on the weather. However, the weather forecasts for long periods are rarely as accurate as they need to be (Monache.D, 2010). Because of this, the prognostication must be valid and needs to be updated continuously to get the best possible accuracy.

In this degree project, a premade prognostication model is used to help improve the current production optimization tool where the heat storage unit is not considered. The district heating system to optimized is located in Gävle, Sweden, where Gävle Energi AB owns the district heating grid.

1.1 Gävle Energi AB

The population size of the municipality of Gävle is around 100 000 (SCB, 2019) which makes it the 17th biggest municipality in Sweden, based on the population size. From the Gavle creek that runs through Gävle, there are seven hydropower plants with the purpose to produce power

to Sweden's national grid. During a normal year, the power produced from these hydropower plants is 56 GWh. (Gävle Energi, 2019) In addition to these hydropower plants, there are two combined heat and power plants (Bomhus & Johannes), two heat only plants (Carlsborg & Ersbo) and a pulp and paper industry (BillerudKorsnäs) where heat is recovered from various processes to the district heating grid. These facilities supply Gävle with heat during the entire year. The base load for heat production is the waste heat from the industry, BillerudKorsnäs, and when this is not enough to cover the demand the CHP-plants can help to meet the demand (Gävle Energi, 2019). Johannes CHP-plant has a production capacity of 23 MW power and 77 MW heat with flue gas condensing included. (Bomhus Energi, 2019) The fuel used in Johannes is biofuel, consisting mainly of bark and recycled wood (Gävle Energi, 2019). A heat storage unit is also present at Johannes, which creates the possibility to store heat for later, increasing the flexibility of the CHP-plant. The storage unit can divert a part of the incoming district heating flow, heat it and then send it to the outgoing water in the district heating grid, where it is mixed with the water heated in the condenser at Johannes. During the coldest days or when the primary facilities are not available two back-up heat plants, Ersbo and Carlsborg can be used to meet the heating demand. Both back-up plants use bio-oil as their fuel. Ersbo has a capacity of 80 MW and Carlsborg 60 MW. (Bomhus Energi, 2019)

1.2 Bomhus Energi AB

Bomhus Energi AB (BEAB) was founded in 2010 as a result of a cooperation between the municipality owned Gävle Energi AB and the paper industry BillerudKorsnäs AB. Both parties own 50% each of BEAB. The CHP-plant is located in the same area as the paper industry. Gävle Energi AB and BillerudKorsnäs AB decided not to build one boiler each but instead build a conventional boiler which became BEAB with the purpose to secure the steam production needed for the paper industry, the sawmill located nearby, and supply heat to the district heating grid when possible and needed. The industries require steam at two pressure levels, 4 and 12 bar. (Bomhus Energi, 2014) BillerudKorsnäs produce single-use portion liquid packaging's, and white top kraftliner. Kraftliner is a paper product that is used in the outermost layer of corrugated fiberboard (Skogen, n.a.). The total production is 740 000 tons every year. (BillerudKorsnäs, 2019) The business is competitive and requires a high-quality product, making it important to deliver the steam it needs (Confederation of European paper industries, 2006; Karikallio, Mäki-Fränti, & Suhonen, 2011). Electricity is also produced from Bomhus but is seen as a co-product to increase the total efficiency since the produced heat is the primary product. The CHP-side consists of a biofueled boiler and a turbine from Siemens. The turbine has a nominal power of 92 MW, and maximum heat production from the condenser is 150 MW with flue gas condensing not included. The primary fuel used for the boiler is bark. (Bomhus Energi, 2019)

Besides the bio-fueled boiler, there are two recovery boilers and two electrical boilers that can produce steam. The recovery boilers are a part of the paper industry and help the chemical cycle by recovering chemicals from the black liquor which is a waste from digesting pulpwood, by combusting the organic substances that comes from the wood chips in the treatment process (Bomhus Energi, 2019). Another process where heat is recovered is from the black liquor evaporators. After the black liquor has been separated from the pulp fiber from the digesters and pulp wash, the black liquor contains too much water to be used as fuel in the recovery boilers (Bomhus Energi, 2019). Bergsten (2015) describes the process as the following, to decrease the moisture level of the black liquor, a series of evaporators is used. These evaporators are called effect stages. One effect stage is made up of one or more evaporators operating with the same pressure level. The most common approach is to use subatmospheric pressure, this to make the evaporation process easier, the further it gets. In the first effect stage steam is used to evaporate the moisture in the black liquor, then the steam is directed away, and its heat is recovered to the district heating grid. The next effect steps use the evaporated black liquor as the hot fluid. Therefore, a lower pressure is required in the other effect stages. A higher number of stages require a lower amount of steam compared to a lower number of effect stages. The sub-atmospheric pressure is created by a condenser that cools the liquor vapor in the last effect step. Figure 1 illustrates an example of three effect stages. From the condenser that chills the liquor vapor medium temperate water is produced. For the highly contaminated liquor condensate, a large portion of combustible organic material that can be separated from the water using a distillation process, this is done in a stripping column.



Figure 1 Black liquor evaporator with three effect stages

The electric boilers have a total production capacity of 80 MW, and its primary purpose is to be used as a backup when needed but, it can also be one of the more cost-effective ways to produce steam when the electricity is cheap. The paper industry also has an oil-fired backup boiler with a capacity of 110 MW. In the system, there is a steam accumulator that can store excess steam that can be used for the industry or to cover the heating demand for the district heating grid. Figure 2 shows an overview of the system described in this section. (Bomhus Energi, 2019)



Figure 2 Simplified block scheme of steam production for the district heating grid and industries

1.3 Purpose

The purpose of the project is to develop a process model that can, from a prognosticated heat demand, provide guidance and decision support of how the facilities should be operated to provide the best possible economic outcome, in terms of minimizing the production cost which involves revenues from sold electricity. Which also includes the heat storage unit as well, when to charge it and when to discharge it.

1.4 Research questions

- How should the heat storage unit in Johannes be operated, in terms of charging and discharging, to get the best economic outcome while meeting the heat demand during the simulation period of 3 days?
- What is the optimal mix of boiler loads to meet the heat demand and gives the lowest production cost and highest profit?

1.5 Delimitations

The power production from the CHP-plants is not researched since the heat production is the main product, and the electricity is seen as a by-product. Forecasting of the heat demand and electricity price is already developed by Gävle Energi and is not studied. The simulated period is delimited to 3 days since the heat demand forecast is based on the weather forecast. In order to meet the heating demand, a specific feed temperature should be required. However, in this study, only the needed energy is observed. Some delimitations in costs for additional software are taken into consideration. Due to the end model being developed for a customer, the choice of software must be accepted by the end user and be within a reasonable price.

1.6 Contribution to current research

The contribution of this degree project is to provide a simulation model related to the district heating grid in Gävle. The modeling methodologies that are described in the literature study are not immediately suitable for the problem related to this degree project, in terms of planning period and number of production units. Therefore, a new approach is undertaken to solve the problem for this specific district heating grid.

2 METHOD

In this section, is the methodology about the gathering of data and the simulation model explained.

2.1 Gathering of data

A comprehensive literature study was made in order to get a better understanding of the problem as well as finding different methods to solve the problem. Most of the information about suitable approaches to solve the problem is gathered from similar research projects and scientific papers found in the databases Google Scholar, Research Gate, and Science Direct. The general information about energy storage and basic mathematics is gathered from books. In order to find relevant scientific papers, some specific search parameters were used, knowing it is a widely used term in the problem area. Unit commitment, economic dispatch, short-term planning of district heating, and cogeneration are some phrases that were used commonly to find information. After searching for information, some names were seen a lot in different papers handling the same issues, giving the impression they have had a significant influence on the industry. More papers were found by using the frequently appearing names.

A study visit to the different facilities was made to gain a better understanding of the production units and the distribution system. Relevant input data about the various production units were gathered during a meeting with the process engineers at BillerudKorsnäs AB, Gävle Energi AB, and Bomhus Energi AB. The collected data have a central role in the development of the model.

2.2 Modeling and simulation

The model was created with MATLAB. The base model runs the main script which reads the input data including the prediction of the district heating demand and electricity market price from an excel file, then it calls on functions for all production units and one optimization function, the process is explained in Figure 3 below. Since the problem is rather complex, several different solvers are used in the optimization function. Which solver to use in the optimization function is highly dependent on the problem formulation. If the problem is simplified to a linear programming (LP) problem, a more straightforward solver can be used, which reduces the accuracy of the model. If the problem, on the other hand, is stated as a mixed integer non-linear programming (MINLP) problem, the model can be more accurate, and a more advanced solver is needed.



Figure 3 Flowchart of the simulation process.

2.3 Validation of result

Since there is no data to compare the result with, other measures must be used to ensure that the result is the global optimum, and not a local. If the objective function is convex, only one optimum exists. Validation of the model was done by using the same approach as previous projects with similar scope, and by performing a sensitivity analysis. The sensitivity analysis was performed by changing input data and observe the result to control if it behaved the correct and anticipated way.

3 LITERATURE STUDY

In the following chapter will the technical information relevant to this project be presented. A reference to Erik Dotzauers article from 1997 is used as a base for this project. This is motivated by that the mindset on how to simulate/operate a plant is not changed since. Therefore, the same constraints can be used to get a valid simulation model. Also, it is confirmed by recent articles using the same methodology. Chenghong, Da, Junbo, Xitian, and Qian (2015) confirms that a CHP plant is either operated after the electricity or heat load. In this work is it the heat load that decides the production, Dotzauer (1997) uses the same strategy. All images with references have been approved by their respective authors to be used in this work.

3.1 Thermal storage

According to Frederiksen and Werner (2014), large scale thermal storage is often divided into two methods, short-term storage, and seasonal storage. The seasonal storage method is under progress and is often related to thermal solar power. Short-term storage is, however, a well-established method whose purpose is to:

- Push the load from a period when the demand is high to a period where the demand is lower.
- Reduce rapid load changes that can harm the production unit.
- Avoid energy losses which are related to frequent start and stops of the production unit.
- Increase the production of electricity when, for example, the demand for cooling is higher than the demand for heat.

In Figure 4, the load shedding for a district heating system presented. The storage tank discharges when the heating demand is higher than the available heat production.



Figure 4 Load shedding of a district heating system (Frederiksen & Werner, 2014)

In a district heating system is the most common method for thermal storage, the short-term storage method which bases on thermal stratification. Thermal stratification occurs when the temperature of the water increases which decreases the density of the water, which in turn makes the mass of hot water lay on top of the mass of cold water (Frederiksen & Werner, 2014). In order to avoid stratification degradation and to improve the storage cycle performance must the injection, recovery, and holding of heat be managed carefully (Dincer & Rosen, 2012).

Figure 5 which according to Frederiksen and Werner (2014) explains a pressureless hot water storage tank (1) which is connected to a district heating grid. The storage tank is connected in series with a combined heat and power plant (12) and a backup boiler (13). In the figure, two parallel circuits are presented, in order to charge the tank, open the charging valve (6) and start up the charging pump (3). To discharge the tank, open the discharge valve (4) and start the discharging pump (4). The water level in the tank remains stable by the circuit including a pump (10) and a valve (11), when the water level is to low the pump startup and pump water from the reservoir tank (9), and when the water level is too high the valve is opened in order to transport water to the reservoir.



Figure 5 Pressureless heat storage tank connected to a district heating grid. (Frederiksen & Werner, 2014)

3.2 Optimization of a CHP-plant with a heat storage unit

When planning district heating production, it is critical that there is a time perspective considered, especially if there is a heat storage unit involved. According to Dotzauer (1997), the time perspective can be divided into three levels, long-, medium-, and short-term production planning. The long-term planning involves decisions about future investments, including production capacities and expanding the district heating grid in a coming period of 5-25 years. Medium-term planning handles fuel and electricity contracts in the next coming 1-5 years. Short-term planning is about finding the optimal operation conditions for the facilities in a period from 1 – 7 days. (Dotzauer, 1997; Salgado & Pedrero, 2008) Commonly it is divided into subproblem for each production unit, and to increase the accuracy of the planning the period to be solved should be for the upcoming 24 hours (Dotzauer, 1997). Frederiksen and Werner (2014) describe the benefits of short-term planning as is it giving lower marginals for fuel consumption and a higher precision electricity production of a CHP-plant. Today the market for electricity requires that all producers of power must report their planned production for the next coming day. This helps to balance the supply and demand. (Frederiksen & Werner, 2014) It is crucial that the unit is being operated in an optimal condition in order to reduce cost, especially plants where the production cost is high (Dotzauer, 1997; Frederiksen & Werner, 2014), for example, oil-fired boilers. For a plant with a high production cost, a lot of money can be saved daily (Dotzauer, 1997). The optimal operating conditions also help to reduce unwanted emissions, which also help reduce costs. With an increasing number of production units and heat storage units, the complexity and difficulty to

find the optimal operating conditions also increase rapidly. From Erik Dotzauers (1997) analysis for optimal operation, the below-described aspects are what complicates the problem and must be taken into consideration.

- With several production units, it can be hard to decide which ones that should be used as a baseload, it limits the minimum operation- and downtimes of the plants. The minimum operation time is the time that the plants must be operated if started, and minimum downtime is the time the plant must be off if the operation is shut down. This must be considered because there is a start-up cost for the plants that affect the optimization. Dotzauer calls this problem of which unit should be operated and which should be off "the unit commitment problem". The term is widely used in the sector and is extensively researched in its community (Pandžić & Luburić, 2019; Fossati, 2012).
- If any of the units that are optimized are combined heat and power plants, further problems appear, such as deciding the electricity production. In Sweden, the electricity is sold on an open market, Nord Pool (Energimarknadsbyrån, 2019), and the general goal is to maximize electricity production when the price for electricity is high. A CHP-plant has its highest overall efficiency at a proportional production of heat and power. This means that when electricity production is high, then heat production also is at high levels. If the produced heat exceeds the demand, problems may arise.
- With a heat storage unit, the production flexibility increases, due to that the excess heat from high electricity production can be stored for later when the electricity price is low, and the operation wants to avoid producing a small amount of electricity and high amount of heat to keep the total plant efficiency high. This tactic helps to even out peaks of the heat demand. However, when performing the optimization, this instantly increases the difficulty because a highly time-dependent aspect is introduced.
- The outline of the district heating grid has a significant influence on how the plant can produce heat since a high incoming temperature decreases the efficiency of a plant drastically, therefore it is essential that the heating grid is not connected in series but rather to use substations to deliver and distribute the heat.

To solve this complex problem, a combination of mathematical models and fitting algorithms are required. Bellqvist and Olofsson (2012) used the software GAMS and reMIND to solve the problem of similar nature at Luleå Energi AB and Lulekraft. The model made in GAMS used mixed integer non-linear programming (MINLP) and the model in reMIND mixed integer linear programming (MILP). Vennström (2014) at Umeå University also used the MILP to solve such a problem. Dotzauer (1997) has used Matlab combined with toolboxes from TOMLABS NPSOL to solve a short-term planning problem. Gopalakrishnan and Kosanovic (2015) used a genetic algorithm to solve their operational planning of a CHP with a mixed integer nonlinear problem.

To help others, Dotzauer (1997) constructed a guide to help decide which method to use, and it is dependent on the following characteristics:

- Deterministic or stochastic
- Continuous or discrete

- Linear or non-linear
- Convex or non-convex

A linear problem that has both continuous and integer variables are viewed as a MILP, and if the problem is non-linear with continuous and integer variables makes it a MINLP. (Klanšek, 2014) Below a couple of methods that can be used to solved MILP and MINLP are shown (Dotzauer, 1997).

- Dynamic programming
- Heuristic methods
- Branch and bound
- Lagrangian relaxation

3.3 Modeling of non-linear efficiencies

By using non-linear efficiencies, the credibility and accuracy of an optimization increases, compared to constant or linear efficiencies, this because it better reflects the real efficiencies (Tveit, Savola, & Fogelholm, 2005). The efficiencies are highly dependent on the installed capacity and the load, therefore, both these variables can be seen as the main controlling variables for the efficiencies (Milan, Stadler, Cardoso, & Mashayekh, 2015). Dotzauer (1997) shows how a secondary polynomial equation can be approximated to describe non-linear efficiencies. This requires empirical data to be available. To get empirical data, the plant can be operated in an off-design mode (Savola & Keppo, 2005). Operating the plant in off-design mode means that different parameters are changed to see how the rest of the system is affected, while still meeting a production target (Kupecki, 2015). Milan et al. (2015) describe two methods for non-linear modeling efficiencies. Both ways use linearization of the non-linear curves. The first approach uses binary variables. From the efficiency curve, a matrix is created where different loads are connected to particular efficiencies, Table 1.

The binary variables create classes, and every binary variable is related to a specific value in the matrix. When the actual load is known, the efficiency is found by finding the nearest located load in the matrix. This does not give the exact efficiency because interpolation between tabulated values does not work with the binary variables. However, it is a simple solution to avoid using constant efficiencies.

		Load [%]									
Installed capacity [MW]	0	10	20	30	40	50	60	70	80	90	100
η, Boiler 1 [%]	0	31	36	42	47	53	59	65	70	74	78
η, Boiler 2 [%]	0	33	38	44	49	55	61	67	72	76	80

Table 1 Example of efficiencies at different loads

The second method gives a more precise result since it allows for interpolation to be made between the tabulated values. Instead of using binary variables, Special-Ordered-Set (SOS) variables are used. SOS variables are a set of variables in consecutive order. Compared to other variables, only one (SOS1) or two (SOS2) adjacent elements in the ordered set can be set to a non-zero value. SOS variables can be used as weighting factors in linearization methods, by setting its upper and lower bounds to one and zero, respectively. Using the variables as weighting factors, a tabulated, or an interpolated value between two loads in the table can be retrieved. However, SOS-variables requires a MILP-method that supports this type of variable. Milan et al. (2015) use the CPLEX solver in the software GAMS to perform the optimization. In order to find the correct boiler (row) in Table 1, SOS1 is used, and to perform the interpolation SOS2 is used. The most accurate way of modeling the non-linear efficiency is to curve fit an equation from empirical data (Savola & Keppo, 2005).

3.4 Non-convex & non-continuous problems

There are problems whose characteristics cannot be assumed to be convex, a problem like this is, for example, modeling of non-linear efficiencies (Makkonen & Lahdelma, 2005). If the model has non-linear efficiencies, it leads to the production mix of heat and power being nonconvex. A non-convex model also results in there being local optimums, which can be a problem (Chun-lung, 2007). The characteristics for a modern CHP-plant is often noncontinuous due to reasons such as backpressure production, the possibility to use different types of fuels and a high variety of operations modes, this in turn also makes it non-convex (Makkonen & Lahdelma, 2005). There are several ways to simplify this problem. One method is to divide the model into several convex sub-models. The sub-models are then modeled to be alternative components, and only one sub-area can be operated at a time (Rong & Lahdelma, 2007), Lien and Amato (2004) call this convex decomposition. Convex decomposition is a common way of handling non-convex problems and helps most algorithms to perform better (Lien & Amato, 2004). Makkonen and Lahdelma (2005) describe an example of a CHP-plant with a backpressure turbine with a bypass control valve, condenser, and an additional cooling unit. Altogether, this creates nine specific points, Figure 6.



Figure 6 Example of convex sub-models, 9 points (Makkonen & Lahdelma, 2005)

Figure 6 shows the heat produced as a function of power production, which Makkonen and Lahdelma (2005) explains as the following. Point 1 - 2 is the backpressure line, meaning that if the heat production should increase the bypass valve should be opened accordingly to get the wanted increase of heat, this also leads to a decreased power production due to less steam passing the turbine. Opening the bypass valve pushes the state towards points 8 and 9. However, in a reversed situation, increased power production, the current state is pushed downwards to point 3. This production mix lowers the total efficiency. If even more power is required and the cooling capacity of the condenser is not enough, the additional cooling unit is needed, the production state is pushed towards four to seven, and even further reduces the total efficiency. The total area of the graph in Figure 6 is non-convex, therefore by dividing the area into smaller parts it can be made convex, each area represents a sub model (Makkonen & Lahdelma, 2005).

3.5 Branch and bound optimization

According to Vanderbei (2014), many real-world problems can be modeled as linear programs except that some or all variables are constrained to be integers. One technique for solving problems described as integer programming (IP) problems or mixed integer programming (MIP) problems is the branch-and-bound method. The branch-and-bound method can solve a potentially large number of linear programming (LP) problems in its search for an optimal solution. The algorithm starts by ignoring the integer constraints to solve the problem and hope that the solution vector satisfies the integer constraints. The hopes for this first solution are almost always unfulfilled, and a backup strategy is required. The backup strategy is to develop a tree of linear programming subproblems. Consider the following example.

$$\max_{x_1, x_2} \quad 17x_1 + 12x_2$$

s.t. $10x_1 + 7x_2 \le 40$
 $x_1 + x_2 \le 5$
 $x_1, x_2 \ge 0$
 $x_1, x_2 \text{ integers}$

The linear programming problem obtained when the integer constraints are ignored, is called the LP-relaxation. The solution to the LP-relaxation is at $(x_1, x_2) = (5/3, 10/3)$ which gives the optimal objective value 205/3 = 68,33. By rounding each component of this solution to the nearest integer gives $(x_1, x_2) = (2, 3)$, which is not feasible. The closest feasible integer solution to the LP-relaxation is $(x_1, x_2) = (1, 3)$ which is not the optimal solution. As mentioned, x_1 for the LP-relaxation problem is 5/3. Therefore, the optimal solution to the integer problem satisfies either $x_1 \le 1$ or $x_1 \ge 2$. Let P₁ denote $x_1 \le 1$ and P₂ denote $x_1 \ge 2$. A tree of LP subproblems can now be made as in Figure 7 below, this is called the enumeration tree. The algorithm explores branches of the tree, and every branch is checked against an upper and lower bound, hence the name Branch-and-Bound.



Figure 7 Enumeration tree related to the example. (Vanderbei, 2014)

By adding the $x_1 \le 1$ constraint to the LP-relaxation P_1 is achieved. Since the solution to P_1 is an integer, it is said to be the best-so-far solution. However, since the solution to P_2 is higher than

the one for P_1 , the procedure must continue. The process is moving on until the last step where two integer solution is achieved, which is in P_7 and P_8 where P_8 is highest and said to be the optimum. P_7 must not be an integer but to stop the procedure, should it be lower than P_8 .

According to Leyffer (1999), the same methodology can be used for solving mixed integer nonlinear programming (MINLP) problems. All integer constraints are relaxed, and the resulting nonlinear programming problem is solved. If the solution is an integer solution, the algorithm stops. If not, the algorithm continues by creating an enumeration tree with branches that are checked against lower and upper bounds. The branch and bound algorithm is a global optimization method and Ratschek and Rokne (1995) describes it as almost being a brute search for the optimum, and therefore almost guaranteeing that it is a global optimum, however, due to its brute search characteristic it is time-consuming and exponential.

The branch and bound algorithm have been used by others to solve similar production problems, for example, Lahdelma (2006) uses an envelope-based branch and bound algorithm to solve a production planning problem with non-convex characteristics with success. The problem was defined as a mixed integer linear problem.

4 MODELING

The following chapter describes the process of how the models are constructed and the methods used to gain the best result in terms of accuracy, time, and complexity of the optimization. To contribute with a new approach, the branch and bound based Matlab toolbox created by Kenneth Holmström called TOMLAB is used to perform the optimization. TOMLAB was used by Dotzauer (1997) but with the dynamic optimization method.

4.1 Existing model

The currently used model that is to be improved is done in Excel and was developed by Håkan Yderling at BillerudKorsnäs. It is based on the mentioned heat demand forecast, electricity price forecast model, and production planning for the industry, how much is expected to be produced. The equations used for efficiencies, district heating return temperature, operational costs are curve fitted to give the most accurate result. The basic structure of this model is to create two lists, one that shows how much production each unit can produce in the specific hours, and the other with the particular costs during that hour. From this, a priority list can be made, and then the cheapest ones during each hour are operated to get the lowest possible costs while meeting the heating demand. The way the heat storage tank is operated is to compare how much it would cost to store 1 MW of heat, and how much would be saved by discharging 1 MW of heat and then calculates a marginal cost. This is not optimal cause it does not consider the time aspect as it should.

4.2 Current model

From meetings with Gävle Energi, Bomhus Energi and BillerudKorsnäs performance requirements were collected to help develop the model in a way that satisfies all three actors. From dialogs with Håkan Yderling, the process engineer, at BillerudKorsnäs, it was decided to use the same curve fitted equations as in the existing model. This because they are accurate, and making new equations using the curve fit method would give similar results. Hence it is unnecessary. It was said that there are close to no startup costs for any of the production unit, same with the heat losses for the storage unit, and therefore, not be considered in the optimization done in the project. When deciding which software to construct the model in, an economic aspect must be considered, since the customer should approve of purchasing it. Gävle Energi already has a Matlab license but in order to do the optimization either the MATLAB optimization toolbox or the external TOMLAB toolbox is required. The Matlab optimization toolbox costs 11.700 SEK. TOMLAB has a base cost of 1.250 USD, and the MINLP solver cost 2.540 USD, making it a total of 35 064 SEK (TOMLAB, 2019) using the currency rate 9.25 (11th April 2019) (Xe, 2019). This shows that different solvers come at a large variety of costs. However, to find out which method and approach that gives the best and most accurate real-life operation result at a reasonable time, both approaches are modeled.

4.3 Matlab

The section below describes the different methods used and experimented with to get a result fitted for the customer.

4.3.1 Linprog

Linprog is a function included in the Matlab optimization toolbox. It is used for solving linear programming problems. There is also a function called intlinprog, which is similar, but it allows for integers to be used. However, if integers are to be used, the solver would be required to solve a non-linear problem which it cannot (MathWorks, 2019).

4.3.2 TOMLAB

TOMLAB is developed for a general purpose of development and modeling in Matlab for research, teaching, and practical uses. TOMLAB is created by Kenneth Holmström and is used by several large companies and universities all over the world such as ABB, US Air Force Control Lab, IBM Watson Research Center, NASA, and more. Holmström decided to create TOMLAB because he saw a lack of advanced, robust, and reliable tools for optimization algorithms. Using TOMLABs state-of-the-art optimization software Matlab can be used to solve the complex optimization problems that otherwise would be difficult (TOMLAB, 2019).

4.3.2.1. MinlpBB

MinlpBB solves problems of MINLP type by using the branch-and-bound method. It can be used for large, sparse or dense mixed-integer linear, quadratic, and non-linear programming problems. For the optimization to work the second order information from the nonlinear constraints and objective function are required (TOMLAB, 2019).

4.4 Model characteristics

As mentioned above, the simulation model is built in MATLAB by using different types of solvers. The problem formulation is different for each solver. The general approach of the model is that it first collects data from an input file in the form of an excel spreadsheet. From the gathered data, it first calculates the availability of each unit to control that the unit can be used to produce heat, it could be that there is scheduled maintenance or an unexpected problem that appeared. From the availability, the available energy is calculated, which is how much energy each unit can produce, which then is used to calculate the available production. The available production is what decides how much each unit realistically can produce each timestep.

Before the optimization can start, the production prices for all units and timesteps need to be calculated. The evaporators for the black liquor have three sets of which heat can be extracted. However, when performing the modeling, it is only divided into two variables, where its total available production is 40 MW. It is set to two variables only because there are two different production costs for the evaporators. All production over a specific limit of power costs more. The cost for extracting heat from the black liquor evaporators are dependent on two equations, for the first variable. The cost for the flue gas condenser and steam of 4 bar decides the production costs for the first equation and the second is dependent on the electricity price. The equations are curve fitted, and the lowest cost calculated from the equations is what becomes the price, this is applied to all the cost equation. Therefore, only the dependent factors are described. The cost for the second evaporator variable is dependent on the cost for the first variable.

For the solid fuel to BEAB's bio-fired boiler to be used as heat in the hot water condenser the cost is calculated from the cost to produce steam with the pressure of 4 bar and the cost for the flue gas condenser. The cost for the steam of 4 bar is derived from the boiler efficiency, electricity price, fuel cost, electricity certificate, and cost for producing steam of 120 bar. The efficiency of the boiler is dependent on the load. Operational cost for the electric boiler is calculated from the cost of heat from solid fuel. The cost for the excess steam which is stored in the steam accumulator is set to a specific and constant price. The same is done for the waste heat, flue gas condenser, and all backup units.

For Johannes, the approach is a little bit different because it has been split into four components with a convex characteristic due to it being non-convex with the backpressure functionality. The four convex sub-models are the minimum load, turbine condenser and flue gas condenser, direct condenser, and flue gas condenser, and only direct condenser. The efficiency for Johannes is modeled as constant, due to lack of data. To make sure the minimum load is always operating its cost is set to be zero. Direct condensing happens when the turbine is skipped in cases where the heat demand is high. The cost for the turbine and flue gas condenser is calculated from the fuel price, electricity price, and electricity certificate, and the same is for the direct and flue gas condenser. The cost for only direct condensing is derived from the electricity and electricity certificate price, due to that, the heat that is produced must cover the income from not producing electricity. The turbine condenser can be operated at the same time as the direct condenser. Hence, only a part of the flow is extracted before the turbine.

4.4.1 Linear programming

When using the solver "linprog", the problem must be formulated as a linear programming problem as following:

$$\min_{x} c^{T}x \quad (1)$$

s.t. $Ax \le b \quad (2)$
 $x \ge 0 \quad (3)$

Where function (1) is a linear function of x which is a vector of all decision variables, the objective is to minimize the function value. Therefore, it is called the objective function. Function (2) and (3) are the constraints related to the objective function. The constraints are either equality constraints or inequality constraints, associated with some linear combination of the decision variable. (Vanderbei, 2014)

When using linear programming to solve the problem is stated as the following:

$$\min_{x} f^{T}x \quad (4)$$
s.t. $A_{eq}x = b_{eq}$ (5)
 $Ax < b$ (6)
 $lb \le x \le ub$ (7)

Where x is the vector including all decision variables associated with the storage tank, heat demand and all production units as:

$$x = \begin{bmatrix} Q_{start} \\ Q_{stored} \\ P_{charge/discharge} \\ P_{demand} \\ P_{J.DC,FGC} \\ P_{J.DC} \\ P_{J.DC} \\ P_{J.min} \\ P_{WH} \\ P_{Evap 1} \\ P_{Evap 2} \\ P_{HWB} \\ P_{Carlsborg} \\ P_{Ersbo} \\ P_{FGC BEAB} \\ P_{HWC D} \\ P_{HWC F} \\ P_{HWC F} \\ P_{HWC F} \\ P_{HWC F} \end{bmatrix}$$

The vector f is the vector in which all operational costs associated with the productions are included. There are no operational costs related to storage and heat demand. Therefore, the elements in the first four rows are set to zero. The equality constraints are represented by the general equations 8 - 9:

$$Q_{stored}(i-1) + P_{charge/discharge}(i) = Q_{stored}(i) \quad (8)$$

$$\sum P_{Johannes}(i) - P_{charge/discharge}(i) + P_{WH}(i) + \sum P_{Evap}(i) + P_{HWB}(i) + P_{Carlsborg}(i)$$

$$+ P_{Ersbo}(i) + \sum P_{HWC}(i) = P_{demand}(i) \quad (9)$$

$$\sum P_{Johannes}(i) = P_{J \cdot DC, FGC}(i) + P_{J \cdot TC, FGC}(i) + P_{J \cdot DC}(i) + P_{J \cdot min}(i)$$

where i is the time step

The general equations for the equality constraint used to form a matrix which is necessary to cover all timesteps in the simulation. A simplified equality matrix is presented below.

$$A_{eq} = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}$$

The matrix is based on three timesteps and is divided into six sections by the dashed lines in red. The first section represents the start value of the energy stored in the tank, and it consists of only one column since the start value is related to the first timestep. Section two represents the stored energy in the tank, the storage in timestep i, in equation 8 is moved from the right-hand side to the left-hand side and therefore set as negative, the stored energy in the previous timestep is set as positive. Section three represents the charging, which is positive in the storage constraint and negative in the demand constraint. In the fourth section is the discharging represented, which is negative in the storage constraint and positive in the storage constraint and therefore, set to zero. In the demand constraint is the production set as positive. The sixth and final section represents the demand which is moved from the right-hand side in the demand constraint and therefore set as negative. Since the variables on the right-hand side are moved to the left-hand side for both the storage constraint and the demand constraint and therefore set as negative. Since the variables on the right-hand side are moved to the left-hand side for both the storage constraint and the demand constraint is the quality vector set as the zero vector as:

$$b_{eq} = \begin{bmatrix} 0\\0\\0\\0\\0\\0\end{bmatrix}$$

The inequality constraints are used to make sure that heat produced from a specific producer charges the tank. This is not a problem when discharging the heat storage, equation 10.

$$P_{charge/discharge}(i) \le \sum P_{Johannes}(i) (10)$$

The inequality matrix is based on the same concept as the equality matrix. It is crucial that the number of columns is the same and that the sections agree. The inequality vector is set to the zero vector since the right-hand side is moved over to the left-hand side.

The matrixes that represent the lower and upper bound is made for each decision variable. Since the demand is known in every timestep is the lower and upper bound set to the demand. The lower and upper bound for the charging and discharging is set to 0 MW and 40 MW respectively. The lower and upper bound of the storage tank is set to 50 MW and 350 MW. The upper bound for each production unit is set to its maximum load, the lower bound must be set to 0 MW in order to avoid a production unit to produce its minimal load when it is turned off. This is a simplification of the problem since in the real world a production unit cannot produce in the range between zero and the minimal load. To make the model support the minimal load, another binary variable must be implemented. However, this makes the problem non-linear and therefore, is not taken into consideration for the simplified linear problem. The same occurs if a minimum operating or downtime is introduced.

A way to make the result smoother without the loads going from low to high or reversed, ramping functions are used to create a limitation to how much the load can be increased or decreased between every timestep. The constraint is shown in equation 11.

$$P_{decrease} \le P(i) - P(i-1) \le P_{increase}$$
 (11)

This constraint is only implemented for the backup units, Carlsborg, Ersbo, and HVP since the other loads are more flexible in terms of changes to the operating load.

Logically the optimization tries to empty the remaining energy in the heat storage unit at the end of the simulation. To avoid draining the tank in the end, a function that calculates the average heat demand in the six last hours is used to decide how much energy should be stored at the end of the simulation. This was brought up by the operating personnel at Gävle Energi. For an average heat demand of 200 MW, the stored energy should be at least 200 MWh, and 170 MWh for an average between 150 and 200 MW. A set of threshold values is used to control the stored energy at the tank at the end, in order to avoid emptying it at the end of the simulation. The threshold values are compared to the average heat demand of the last six hours and decide what the minimum allowed stored energy would be at the end of the simulation. To see how to implement the specifics mentioned in this chapter into Matlab, see Appendix 1.

4.4.2 Mixed-integer non-linear programming

By expanding the set of decision variables by introducing integer variables in order to turn on and turn off the production units allowing for minimum production levels, minimum operating and down times, the problem is turned into a mixed-integer non-linear programming problem. The constraints in the linear problem are similar to the ones used in the MINLP. The decision variable then becomes the vector below. Binary decision variables are only required for the production units, which have a minimum load and/or a minimum operating time and minimum downtime. Because of Johannes already being divided into sub-models, of which one is for the minimum load it does not require an on/off variable to describe its status.

$$x = \begin{bmatrix} Q_{start} \\ Q_{stored} \\ P_{charge/discharge} \\ P_{demand} \\ P_{J_{DC,FGC}} \\ P_{J_{TC,FGC}} \\ P_{J_{DC}} \\ P_{J_{DC}} \\ P_{J_{DD}} \\ P_{Evap 1} \\ P_{Evap 1} \\ P_{Evap 2} \\ P_{HWB} \\ P_{Carlsborg} \\ P_{Ersbo} \\ P_{FGC Beab} \\ P_{HWC D} \\ P_{HWC C} \\ P_{HWC C} \\ P_{HWC C} \\ T_{Johannes} \\ T_{Carlsborg} \\ T_{Ersbo} \\ T_{HWB} \end{bmatrix}$$

The non-linear equality constraint now becomes as equation 12. Constraints that affect production units which do not have a binary variable connected to it remains the same.

$$P_{demand}(i) = P_{charge/dicharge}(i) + \sum_{P_{Johannes}} P_{Johannes}(i) * T_{Johannes}(i) + P_{WH}(i) + \sum_{P_{Evap}} P_{Evap}(i) + P_{HWB}(i) * T_{HWB}(i) + P_{Carlsborg}(i) * T_{Carlsborg}(i) + P_{Ersbo}(i) * T_{Ersbo}(i) + \sum_{P_{HWC}} P_{HWC}(i) (12)$$

It is no coincident that the backup plants are the ones who require binary variables to control their minimum operating time because they are the plants that most often would be required to be operating for just a quick moment when the heat load is high. In real life, operating a plant for only one hour is not a choice, and therefore, it is crucial to implement constraints to get a result that is reflected by realistic operating conditions. The minimum operating time is implemented by using a constraint, shown in equation 13. This constraint is based on problems that Dotzauer (2002), Zendehdel, Karimpour, and Oloomi (2008) have described, which has a similar objective.

$$\begin{cases} T_j - T_{j-1} \le T_i, & where \ j = i - t_{min}^{up} + 1, \dots, i - 1 \quad (13) \\ T_{j-1} - T_j \le 1 - T_i, where \ j = i - t_{min}^{down} + 1, \dots, i - 1 \quad (14) \end{cases}$$

To give a better understanding of how the constraint work, it is set up as a matrix. In Gävle the backup units have a minimum operating time of three hours, creating the matrix, with nine timesteps (i = 9), displayed underneath.

ſ	0			•	•	•			0		0
ł											0
ł	0								0		0
İ	-1	1	0	-1	0	0	0	0	0		0
I	0	-1	1	0	-1	0	0	0	0	$*x \leq$	0
ļ	0	0	-1	1	0	-1	0	0	0		0
	0	0	0	-1	1	0	-1	0	0		0
	0	0	0	0	-1	1	0	-1	0		0
l	0	0	0	0	0	-1	1	0	-1-		l ₀

The optimization code for minlp in Matlab can be seen in Appendix 2 -4, to give an example on how to implement this in the Matlab syntax.

4.5 Scenarios

Different real scenarios are tested with the two different simulation models to control that they work as intended independently of the input data. One scenario (Scenario I) is when the paper pulp industry has had some problems and therefore cannot provide the waste heat nor heat from the black liquor boilers during the winter period when the forecast predicts that the weather is going to be cold. Scenario II is similar to scenario I. Issues regarding the factory are present, but with warm weather predicted. Scenario III handles normal operation with no present issues. However, it has a large temperature difference during night and day with an average temperature below zero. The fourth scenario (Scenario IV) operates during normal conditions with warm weather. Scenario V can be compared to scenario III but with average temperatures being above zero. For the more complex and harder solved scenarios, the demand is decreased and increased to analyze how the model handles the optimization. This is made for the electricity price as well. All scenarios are tested with a period of 72 hours.

5 RESULT

In this section results from each scenario are displayed with a time frame of three days, 72 hours. Some of the scenarios have heat demand or electricity price increased and/or decreased.

Figure 8 shows the scenarios where the weather is cold. From the graphs, it can be observed that the resulting heat production in each hour covers the particular demands (thin black line). When it goes below zero, the heat storage unit is being charged, and therefore, a higher total production must be achieved, and when it discharges, it goes on the positive side of the y-axis. Both scenarios use cheaper production alternatives and try to avoid using expensive backup plants as much as possible. Scenario I must operate the HWB for approximately 40 % of the total simulation period while scenario II only uses it for a couple of hours, during the peak demand. In both scenarios, the direct condenser is used when the demand reaches a higher level and stops the entire electricity production in Johannes CHP. A higher capacity for the discharging of the stored energy might have been able to help avoid using the HWB at all for scenario II. Scenario I charges the heat storage in the end due to the implemented function to check the average heat demand in the last six hours and depending on the mean value a certain energy level is required in the tank, and in this scenario, it forces the charging in the end to fulfill the requirement.



Figure 8 Minlp result for scenario I & II

Scenario III, IV, and V are displayed in Figure 9. The third and fourth scenario operates during warmer weather compared to scenario I and II and is visible through the lower heat demand. However, scenario V is close to scenario I and II studying the heat demand. These three scenarios have one thing in common, none of the backup plants are being used, which makes the solutions precisely the same as the result from the linear programming approach. This is mainly due to the lower heat demand, but also due to the better alternatives being available for heat production.



Figure 9 Minlp/lp result for scenario III, IV & V

5.1 Linear vs. Nonlinear

Figure 10 compares the result for scenario I, using the linear and nonlinear approach. The first thing that can be concluded is that the production overall is similar, with minor differences in the middle mainly. The guess for the nonlinear approach is the result of the linear method. By using the nonlinear method, a smoother and less spikey result is withheld, and in this specific scenario, the total cost for all production is the same, meaning that the area of the orange area in the two graphs is precisely the same even though being used in different ways. It is because of the minimum up and downtime the operations differ, in the linear method it can be turned off however, it wants, but not in the nonlinear method. The simulation time for the linear result was a couple of second (<10s) while the nonlinear took 1331 second performing 246 iterations to find the optimum that satisfies all constraints.



Figure 10 Comparison of scenario I minlp and lp result.

For scenario II the linear and nonlinear results differ very little, Figure 11. In the linear result, the HWB is only used during one hour at a load (21.4 MW) that is below the minimum allowed load (25 MW). It because of this, the results of the two methods differ, whenever a backup plant is required to cover the heat load a new set of constraints is introduced and must be satisfied. After approximately 40 % of the simulation period, the graphs are precisely the same due to the backup plants not being required.

Figure 11 Comparison of scenario II minlp and lp result

5.2 Heating Demand

Figure 12 shows a comparison between two modified versions on scenario I and the regular in the middle. The modified version on the top has a 50 % increased heating demand, and the other version has decreased the demand by 50 %. Through comparison, the flexibility of the heat storage unit can be observed. The discharging pattern for all versions has some similarities. For example, the regular and increased versions are similar in the start and end. They are similar in the end due to a constraint saying there must be at least 200 MWh of energy stored in the tank at the end of the simulation. In the increased and decreased version, the discharging happening in the middle section of the period is also similar.

Figure 12 In/Decrease heat demand for scenario I.

5.3 Electricity Price

Figure 13 shows a comparison between an increased (30 %), decreased (30 %), and regular electricity price for scenario I. It shows that while the electricity price is high, the CHP Johannes decides to maintain the electricity production for almost the entire period even though the heat demand is high. It is also because the price for the direct condenser must cover the revenue the sold electricity otherwise would provide. Which is not the case in the regular and decreased version. For all three versions, the total sum of heat production from the electrical boiler is the same. The regular and decreased versions all have the same amount of production for each respective unit, but still uses a different strategy when it comes to the

Figure 13 In/Decrease electricity price for scenario I.

hourly operation. The only thing that differentiates the increase from the other is that the electricity production at Johannes CHP plant and the use of backup unit HWB.

By comparing the total price difference in Table 2, the total cost is profoundly affected by the electricity price. If the scenario had a smaller heat demand, the difference in price would not be as significant.

Electricity price	Cost [SEK]
30% Increase	3 919 778
Regular	3 507 734
30% Decrease	3 070 407

Table 2 Cost comparison scenario I different electricity prices

The units that are affected by a change in electricity price are the black liquor evaporation, hot water condenser, and Johannes CHP plant. The variation in heat production prices is according to Figure 14.

Figure 14 Comparison of production costs for different production units at different electricity prices.

5.4 Storage vs. No storage

Figure 15 shows the difference for scenario III with storage and if the system did not include a heat storage unit. At first, the system without storage seems to have a more even production, but that is not the case because the storage unit is at the bottom of the graph, and therefore causes all other units to follow its pattern up and down.

Figure 15 Scenario III, storage vs. no storage

The most interesting thing when comparing the system if it would not have heat storage available is how the total cost differs. Hence, the same scenario was used but with an increased and decreased heat demand. Table 3 shows the difference in price from the comparison.

Electricity price	Storage [SEK]	No storage [SEK]	Difference [SEK]
50% Increase	1 432 184	1 574 337	142 153
Regular	513 619	537 487	23 868
50% Decrease	70 855	77 279	6 424

Table 3 Scenario III, storage vs. no storage

5.5 Bomhus Energi model vs. Minlp model

The improvements from the minlp compared to the previous constructed by Bomhus Energi can be observed in Figure 16 and Figure 18 (see Appendix 5). It gives a clear visualization of how the minlp tries to avoid using the expensive backup plants, while Bomhus model is not as successful in that challenge.

Figure 16 shows that by using the heat storage smartly and efficiently, it can be completely avoided to start up a second backup plant, Carlsborg in this case. The minlp also takes every chance it can to charge the heat storage while Bomhus model does not.

In both Figure 16 and Figure 18, Bomhus does not seem to prioritize HWC solid fuel but rather use the electrical boilers instead, this is not what the minlp model does. The electrical boiler is more expensive than using a solid fuel boiler unless the electricity price is low. There is also a different priority on how to operate Johannes, when to not produce electricity and when to.

[MWh]	Bomhus	Minlp
Waste heat	11	11
FGC BEAB	1909	1812
<i>E5</i>	0	0
<i>E4</i>	0	0
E3	0	0
HWC Excess	0	0
HWC Solid fuel	1136	1891
HWC Electric	2967	3272
HWB	2532	1593
J. FGC	1383	1383
J. TC	2472	1653
J. DC	1968	2783
Carlsborg	22	0
Ersbo	0	0

Table 4 Total production, Bomhus vs. Minlp (referring to Figure 16)

As presented in Table 4, the amount of production in different production units varies. The difference in production cost varies by around 400 000 SEK, according to Figure 17.

Figure 17 Total production cost comparison between the Bomhus model, the Minlp model, and the Lp model.

6 **DISCUSSION**

The following chapter contains an in-depth analysis of the result regarding the different scenarios and methods. Due to no specific validation method found for branch and bound optimization, the result had to be studied carefully to ensure its quality. However, Ratschek and Rokne (1995) describe the optimization method to always find the global optimums due to it being similar to a brute force search for the optimum. Meaning that it will try every result that satisfies the constraints and compare it to find the optimum. This comes to a downside since there are a lot of feasible solutions. Therefore, the time to find the optimum is highly dependent on the initial guess and is highly time-consuming in many cases. Lahelma (2006) can get the optimal solution using the branch and bound algorithm within a reasonable time. However, the number of units and the complexity of the constraints are not as advanced as in this study and therefore making it easier to find the result. In the scenarios where the same solution is given from the minlp and lp model, the initial guess from the lp solution. This modification did not affect the result but made a significant impact on the simulation time.

The simulation time varies a lot depending on the heat demand and the status of the pulp and paper industry. The run time for a simulation can range from only a couple of seconds to several hours. As Dotzauer (1997) mentions, the simulation time drastically increases as the number of units, and time steps increase. The simulation time is also affected by the number of constraints, some constraints can shorten the time since it limiting the feasible operation zone, and some increases the complexity, making it harder to find an optimal solution (Dotzauer, 1997). For a scenario with a rather low heat demand, the linear programming solver is enough to solve the problem since there is no need to use the production units that require the nonlinear solver. However, the issue regarding the simulation time is not as simple as that. To understand why it is like that the additional constraints must be observed. Basically, in scenarios where the guess made from the linear solution differs a lot from the nonlinear, the simulation time increases a lot in many cases. Which is often related, as mentioned to scenarios when the heat demand is at higher levels. An example on this is when the linear programming, which does not have a minimum up/downtime for the backup units or load, wants to turn on and off these units to just top the production. The solution for this case will then not satisfy the constraints for the nonlinear method, and therefore, the further away the guess is, the more solutions must be studied to find the global optimum. Because the final product is supposed to be used as a decision tool, the simulation time is highly essential to ensure that the plants are being operated optimally. A simulation time that is 1 hour is too long because when the simulation is done the timeframe of the simulation might already have passed in real life, the maximum time that was given from Bomhus was 3-5 minutes, which is obtained with the more simplistic approach, and sometimes with the minlp model. This then leads to that the actual operation differs at the start, and therefore, it is hard to follow the plan withheld from the model. Eventually, this leads to the model being unusable for the customer.

Looking at the result, it seems to operate in an intended way, first starting the cheaper production units and tries to avoid using the expensive backup plants. However, the hard thing

to control is how the storage tank is used, and if it is done in the most cost-efficient manner. It is our belief that it handles it well because when studying the result, a pattern can be seen on how it operates, similar to what Dotzauer (1997) and Frederiksen and Werner (2014) describes in their studies. Whenever the demand goes up, it goes from charging to discharging, and when the demand goes downwards the start of the stacked graph goes below zero (charging), this is occurring for all scenarios. It can also be seen as the system trying to keep an even production rate throughout the simulation period and that the main controlling factor for the operational strategy is the heat demand, which seems logical. By using previously tested methods for both constraints and algorithms, the operation of the heat storage can certainly be assumed to have a better performance than the old simulation model.

In Figure 10, it can be noticed that the turbine is being turned on and off a lot, this is not good since this will affect the lifetime of the turbine. This should be fixed by implementing a constraint similar to the ones used for the backup units regarding the minimum up- and downtime. In the real world, it would be preferable to instead lower the power production instead of shutting if of completely.

To be sure about the heat demand as a central controlling factor, the impact of electricity price also must be considered. In chapter 5.3, it is made clear that the electricity price has a significant impact as well. The units that are dependent on the electricity price is Johannes, the electrical boilers (HWC Electric), solid fuel (HWC Solid fuel), and black liquor evaporators (Evap). At first, it is confusing why in the increased version, the system would choose to continue to produce power at Johannes instead of producing more heat by using the direct condenser, this while using the electrical boilers at maximum capability. This can be explained by what defines the price for the electrical boilers and the direct condenser. The price of operating electrical boilers is only dependent on the electricity price. The direct condenser must cover the revenues that would be made from having the power production, which is including the electricity certificate, making it a more significant factor than just the electricity price which the electrical boilers are dependent on. This is questionable and does not seem like a feasible approach. However, it is given from Bomhus, and the instruction was to use the same prices as the previous model. This issue does not disturb the logic of the actual optimization only affects the result, meaning that if it was to be changed the model itself does not need to change only its input. Frederiksen and Werner (2014) describes an operational strategy that the unit will aim to produce as much electricity as possible at high electricity market prices, and through this also getting a good heat production and total plant efficiency, which is what happens at Johannes. However, instead of producing power in scenarios where the heat demand is high, it is more profitable to produce heat only instead of starting a backup unit.

In order to show the possibilities that are enabled by having heat storage and using it in a smart way, Table 3 is used. The scenario that was used for this result is not an extreme case where the demand is high, but a regular demand like many days during an entire year. It shows that a very small amount can be saved whenever the more expensive units can be off, and the cheaper alternatives can be used to provide the heat required. It also shows that much money can be saved during a small timeframe, in this case, 142 000 SEK, and this can elevate to

significant high numbers whenever the fossil-based units must help, which is not the case in the result.

In order to not drain the heat storage tank in the last timestep which is the most cost-effective operation since the start level in the tank can be considered as "free" heat, an equation based on the average heat demand is used. Depending on the average heating demand during the six final hours of the simulation, the final value of the heat storage tank is set to a fixed value. The fixed value varies between the intervals of the average heating demand, if the average heating demand is high, then the fixed value is high, and if the average heating demand is low, then the fixed value is low. The reason to have this equation is to achieve a safer operation of the heat storage tank. If there are some unexpected issues with some of the base load production units, then it can be valuable to have a higher amount of heat stored in the tank to provide heat to the grid until the issues are fixed. It is possible for the operation technicians to regulate the intervals and fixed values, which is good cause different season would require different values. Another possibility to handle this would be to check at what time of the day the simulation would end, and if it is near the usual peak hours, it should store energy or vice versa.

The linearization of the Johannes CHP seems to be working well. This can be seen in the result be analyzing the ratio between the direct condenser and the turbine condenser. The production from the turbine condenser is being lowered while the direct condenser is turned on, as well as being shut down when the direct condenser is being used on maximum capacity, this is the same functionality as described by Makkonen and Lahdelma (2005). This also helps to ensure a global optimum to be found by the algorithm (Lien & Amato, 2004).

When the simulation model is compared to the previous model used at Bomhus Energi, it shows that the costs are decreased by around 400 000 SEK, Figure 17. The comparison is made for a scenario with rather high heat demand. The yield is not necessary as significant for scenarios with a lower heat demand since the more expensive production units do not have to be used. The fact that the fuel prices and electricity prices that the estimation is based on are specific for scenario I, also make the possible savings specific for scenario I.

7 CONCLUSION

The purpose of this degree project is to develop a simulation model that optimizes heat production units in the municipality of Gävle, Sweden. The heat production units include combined heat and power plant, backup plants, industrial waste heat recovery, and a heat storage tank. The main idea is to use the heat storage tank to operate heat production to optimize the economic outcome.

The result concludes the following. The heat storage tank is discharged when the demand is increasing to a peak and then charged when the demand is decreasing. The reason for this behavior is to avoid using the more expensive backup plants when the heat demand peaks. A critical part of the model is the simulation time, Bomhus Energi requested a model that is able to compute a result within 5 minutes. For the mixed integer non-linear model is the simulation time an issue when handling scenarios with relatively high heat demand. This is due to the non-linear constraints, which are related to the backup plants that are needed to operate the backup units. The simulation time can vary between different scenarios, from less than 30 seconds to several hours. Therefore, it is recommended for Bomhus Energi to use the linear model, which can solve any simulated scenario in less than one minute. When the two models (minlp & lp) made in this report are compared to the model that Bomhus Energi uses to plan the production. A potential saving of 400 000 SEK can be achieved by producing heat according to the non-linear or linear model instead of using the Bomhus Energi model. The result of the comparison is only related to scenario I since the electricity price, fuel prices and boiler efficiencies are specific for that scenario.

8 FURTHER WORK

To further improve the model and its result, some aspects can be considered by analyzing the result and the method. One way to make the model more dynamic and user-friendly a time aspect of how the storage unit is used at the end of the simulation. For example, it should consider the time of the day at the end of the simulation to know if there will be coming a large peak or if not. Also, the time of the year has an impact due to the temperature differences, for example, during the summer period, it might not be as important to have the same buffer as during the wintertime.

The model can be done more detailed, in terms of using mass flows, pressures. Which also would make the model optimize the specifics of the plants to make them operate in an optimal way to reach the required power output. The curve fitted equations could also be studied to get their accuracy as well as the prognosis of the heat demand and electricity price. In order to make the CHP Johannes operate even smoother in terms of switching on and off the turbine, this will also help to extend the lifetime of it. This could be done in the same way as the minimum uptime and downtime for the backup units.

Other software could be interesting to use and see how the performance is affected while using the same methodology. Different algorithms can also provide insight into how well the model performs compared to other, for example, the software GAMS with its CPLEX solver could be used.

REFERENCE LIST

- Abdollahi, E., Wang, H., Rinne, S., & Lahdelma, R. (2014). *Optimization of energy* production of a CHP plant with heat storage. IEEE.
- Bellqvist, D., & Olofsson, D. (2012). *Driftoptimering av Luleås fjärrvärmesystem*. Luleå: Luleå Universitet.
- Bergsten, D. (2015). *Modellering och analys av sekundärvärmesystemet på Iggesunds bruk*. Umeå: Umeå Universitet.
- BillerudKorsnäs. (2019, April 3). *BillerudKorsnäs: Om BillerudKorsnäs: Gävle*. Retrieved from BillerudKorsnäs: https://www.billerudkorsnas.se/om-billerudkorsnas/vara-produktionsanlaggningar/gavle
- *Bomhus Energi*. (2014). Retrieved from Om företaget: http://www.bomhusenergi.se/ombomhus-energi
- Bomhus Energi. (2019, April 3). *Bomhus Energi: Teknisk Data*. Retrieved from Bomhus Energi: http://www.bomhusenergi.se/tekniskdata
- Chem. (2019, Mars 19). *Chem: Programming in FORTRAN*. Retrieved from Chem: http://www.chem.ox.ac.uk/fortran/fortran1.html
- Chenghong, G., Da, X., Junbo, S., Xitian, W., & Qian, A. (2015). *Optimal Operation of Combined Heat and Power System Based on Forecasted Energy Prices in Real-Time Markets§.*
- Chun-lung, C. (2007). Non-convex economic dispatch: A direct search approach. *Energy Conversion and Management*, 219-225.
- *Confederation of European paper industries*. (2006). Hämtat från Competitiveness and Europe's pulp and paper industry: The State of Play: http://www.cepi.org/system/files/public/documents/publications/competitivenessa ndtrade/2006/2006Competitiveness%20and%20Europe%20s%20Pulp%20%20Pape r%20Industry%20The%20State%20of%20Play.pdf
- Dincer, I., & Rosen, M. A. (2012). *Exergy: Energy, Environment and Sustainable Development*. Elsevier Science & Technology.
- Dotzauer, E. (1997). Algorithms for Short-Term Production-Planning of Cogeneration Plants. Linköping: Linköping University.
- Dotzauer, E. (2002). *Produktionsplanering av el och värme Matematiska modeller och metoder*. Västerås: Mälardalens Högskola.
- Energimarknadsbyrån. (2019, January 31). *Energimarknadsbyrån: Så fungerar elemarknaden*. Retrieved from Energimarknadsbyrån:

https://www.energimarknadsbyran.se/el/elmarknaden/sa-har-fungerar-elmarknaden/

- *Energimarknadsinspektionen.* (2015, May 27). Retrieved from An electricity market in transistion: https://www.ei.se/sv/Publikationer/Rapporter-och-PM/rapporter-2015/an-electricity-market-in-transition-demand-flexibility-and-preference-heterogeneity/
- Fossati, J. (2012). Unit commitment and economic dispatch in micro grid.
- Frederiksen, S., & Werner, S. (2014). Fjärrvärme och fjärrkyla. Lund: Studentlitteratur.
- Garcia-Gonzales, J., Moraga.R, R., Santos.M, L., & Gonzales.M, A. (2008). *Stochastic joint optimization of wind generation and pumped-storage units in an electricity market*. IEEE.
- Gill, P., Murray, W., Saunders, M., & Wright, M. (1986). User's guide for NPSOL 5.0: A FORTRAN package for nonlinear programming.
- Gopalakrishnan, H., & Dragoljub, K. (2015). Operational planning of combined heat and power plants through genetic algorithms for mixed 0-1 nonlinear programming. *Computer and Operations Research*, 51-67.
- Gävle Energi. (2019, April 3). *Gävle Energi: Om fjärrvärme*. Retrieved from Gävle Energi: https://www.gavleenergi.se/om-oss/om-vara-tjanster/fjarrvarme/
- Karaoğlan, S., & Durukan, T. (2016). Effect of environmental awareness on willingness to pay for renewable energy. *International Journal of Business and Management Invention*, 42-48.
- Karikallio, H., Mäki-Fränti, P., & Suhonen, N. (2011). Competition in the global pulp and paper industries An evaluation based on three approaches. *Journal of Forest Economics*, 91-104.
- Klanšek, U. (2014). A comparison between milp and minlp approaches to optimal solution of nonlinear discrete transportation problem.
- Kupecki, J. (2015). *Off-design analysis of a micro-CHP unit with solid oxide fuel cells fed by DME*. Elsevier Ltd.
- Lahdelma, R. (2006). An efficient envelope-based Branch and Bound algoritm for nonconvex combined heat and power producton planning. *European Journal of Operational Research*.
- Leyffer, S. (1999). Integrating SQP and Branch-and-Bound for Mixed Integer Nonlinear Programming. Dundee: Kluwer Academic Publisher.
- Lien, J.-M., & Amato, N. (2004). Approximate Convex Decomposition. New York.

- Makkonen, S., & Lahdelma, R. (2005). *Non-convex power plant modelling in energy optimisation*. Elsevier.
- MathWorks. (2019, April 3). *MathWorks: Documentation: fmincon*. Retrieved from MathWorks: https://se.mathworks.com/help/optim/ug/fmincon.html?s_tid=srchtitle
- MathWorks. (2019, April 3). *MathWorks: Documentation: ga*. Retrieved from MathWorks: https://se.mathworks.com/help/gads/ga.html?s_tid=doc_ta
- Milan, C., Stadler, M., Cardoso, G., & Mashayekh, S. (2015). *Modeling of non-linear CHP efficiency curves in distributed energy systems*. Elsevier Ltd.
- Monache.D, L. (2010). Kalman Filter and Analog Schemes to Postprocess Numerical Weather Predictions. *AMS100*.
- Mueller, S., Tuth, R., Fischer, D., Wille-Haussmann, B., & Wittwer, C. (2013). *Balancing fluctuating renewable energy generation using congeneration and heat pump systems.*
- Pandžić, H., & Luburić, Z. (2019). FACTS devices and energy storage in unit commitment. International Journal of Electrical Power & Energy Systems, 311-325.
- Pappala, V. S., Erlich, I., Rohrig, K., & Dobschinski, J. (2009). A stochastic model for the optimal operation of a wind-thermal power system. IEEE.
- Ratscheh, H., & Rokne, J. G. (1995). Interval Methods. i P. M. Pardalos, & R. Horst, *Handbook of Global Optimization* (ss. 751-828). Dordrecht, Netherlands: Springer.
- Rong, A., & Lahdelma, R. (2007). *An efficient envelope-based Branch and Bound algorithm for non-convex combined heat and power production planning.*
- Salgado, F., & Pedrero, P. (2008). *Short-term operating planning on cogeneration systems: A survey.* Elsevier.
- Savola, T., & Keppo, I. (2005). Off-design simulation and mathematical modeling of smallscale CHP plants at part loads. Elsevier Ltd.
- SCB. (2019, April 3). *SCB: Folkmängd, topp 50*. Retrieved from SCB: https://www.scb.se/hitta-statistik/statistik-efter-amne/befolkning/befolkningenssammansattning/befolkningsstatistik/pong/tabell-och-diagram/topplistorkommuner/folkmangd-topp-50/
- Skogen. (n.a.). Hämtat från Kraftliner Glossary: https://www.skogen.se/glossary/kraftliner
- Swedish Competition Authority. (1996, November). Retrieved from Deregulation of the Swedish electricity market: http://www.konkurrensverket.se/globalassets/english/publications-anddecisions/deregulation-of-the-swedish-electricity-market.pdf

- *TOMLAB*. (2019, April 11). Retrieved from Commercial price list: https://tomopt.com/tomlab/products/prices/commercial.php
- TOMLAB. (2019, April 3). *TOMLAB: About TOMLAB*. Retrieved from tomopt: https://tomopt.com/tomlab/about/
- TOMLAB. (2019, April 3). *TOMLAB: Products: minlpBB*. Retrieved from tomopt: https://tomopt.com/tomlab/products/minlp/solvers/minlpBB.php
- Tveit, T.-M., Savola, T., & Fogelholm, C.-J. (2005). *Modelling of steam turbines for mixed integer nonlinear programming (MINLP) in design and off-design conditions of CHP plants*. Helsinki University of Technology.
- Vanderbei, R. J. (2014). *Linear Programming Fundations and Extensions*. New Jersey, USA: Springer.
- Vennström, P. (2014). *Modellering och optimering av Lycksele kraftvärmeverk*. Umeå: Umeå Universitet.
- *Xe*. (2019, April 11). Retrieved from Currency converter: https://www.xe.com/sv/currencyconverter
- Zendehdel, N., Karimpour, A., & Oloomi, M. (2008). Optimal Unit Commitment Using Equivalent Linear Minimum Up and Down Time Constraints. *2nd IEEE International Conference on Power and Energy*. Johor Baharu.

Appendix 1. Linear programming

88	Retrieve needed	d data				
	DH demand	= input.demand;				
	J DK pris	= Priser output.J DK;				
	J DK RGK pris	= Priser output.J DK RGK;				
	J_TK_RGK_pris	= Priser_output.J_TK_RGK;				
	J_Min_pris	= Priser_outp	= Priser_output.J_Min;			
	J_DK_mp	= Johannes_ou	= Johannes_output.DK_mp;			
	J_DK_RGK_mp	= Johannes_ou	= Johannes_output.DK_RGK_mp;			
	J_TK_RGK_mp	= Johannes_ou	= Johannes_output.TK_RGK_mp;			
	J_Min_mp	= Johannes_output.Min_mp;				
	SV_pris	= Priser_output.SV;				
	SV_mp	= SV_output.SV_mp;				
	IND_1_pris	= Priser_output.IND_1;				
	IND_2_pris	= Priser_output.IND_2;				
	IND_1_mp	= IND_output.IND_1_mp;				
	IND_2_mp	= IND_output.IND_2_mp;				
	HVP_pris	= Priser_output.HVP;				
	HVP_mp	= HVP_output.HVP_mp;				
	Carlsborg_pris	s = Priser_outp	put.Carlsborg;			
	Carlsborg_mp = Carlsborg_output.Carlsborg_mp;					
	Ersbo_pris	= Priser_outp	= Priser_output.Ersbo;			
	EISDO_mp	= Ersbo_outpu	= Ersbo_output.Ersbo_mp;			
	RGR_PIIS PCK_mp	= PCK output	= Priser_output.RGK;			
	HUK D pris	= Priser outr	. NGR_MP,			
	HVK F pris	= Priser outr	out HVK F pris:			
	HVK_I_pris	= Priser outr	out HVK E pris:			
	HVK_D_pris	= HVK output	<pre>= FILSEL_OULPUL.HVK_E_PILS; = HVK_output.HVK_D_mp; = HVK_output_HVK_E_mp;</pre>			
	HVK F mp	= HVK_output				
	HVK E mp	= HVK output. HVK E mp;				
	%% Constraints					
	%% Linear constraints					
	%Define matrixes for constraint					
	n eq cons =	= 2;				
	%Storage const	craint				
	S start =	<pre>zeros(n,1);</pre>	%Initial value for	storage unit		
	Ss =	zeros(n,n);	%Storage			
	S ^c =	zeros(n,n);	%Charge/Discharge			
	S_d =	zeros(n,n);	%Demand			
	S_J_DK_RGK =	<pre>zeros(n,n);</pre>	%Johannes DK RGK			
	S_J_TK_RGK =	<pre>zeros(n,n);</pre>	%Johannes TK RGK			
	S_J_DK =	zeros(n,n);	%Johannes DK			
	S_J_min =	zeros(n,n);	%Johannes min load			
	S_sv =	zeros(n,n);	%Waste heat			
	S_ind_1 =	zeros(n,n);	%Evap 1			
	S_ind_2 =	zeros(n,n);	%Evap 2			
	S_HVP =	zeros(n,n);	%HVP			
	S_Carlsborg =	zeros(n,n);	%Carlsborg			
	S_Ersbo =	<pre>zeros(n,n);</pre>	%Ersbo			
	S_RGK =	zeros(n,n);	%Flue gas cond.			
	S_HVK_D =	zeros(n,n);	%Excess steam			
	$S_HVK_F =$	zeros(n,n);	%Solid fuel			
	$S_HVK_E =$	zeros(n,n);	%Electric boilers			
	*Demand const:	caint				
	D_start =	<pre>zeros(n,1);</pre>	%Initial value for	storage unit		

```
Dѕ
           = zeros(n,n);
                          %Storage
Dс
           = zeros(n,n);
                          %Charge/Discharge
Dd
           = zeros(n,n);
                          %Demand
D J DK RGK = zeros(n, n);
                          %Johannes DK RGK
D J TK RGK = zeros(n, n);
                          %Johannes TK RGK
D J DK
          = zeros(n,n);
                          %Johannes DK
D J min
          = zeros(n,n);
                          %Johannes min load
          = zeros(n,n);
                          %Waste heat
D sv
          = zeros(n,n);
D ind 1
                          %Evap 1
          = zeros(n,n);
D ind 2
                          %Evap 2
         = zeros(n,n);
D HVP
                          %HVP
D Carlsborg = zeros(n,n);
                          %Carlsborg
         = zeros(n,n);
D Ersbo
                          %Ersbo
           = zeros(n,n);
D RGK
                          %Flue gas cond.
          = zeros(n,n);
D HVK D
                          %Excess steam
D HVK F
          = zeros(n,n);
                          %Solid fuel
        = zeros(n,n);
                          %Electric boilers
D HVK E
%Loop in the correct values into the predefined matrixes
for i = 1:1:n
 if i == 1
    S start(i,i)
                     = 1;
                     = -1;
    S s(i,i)
                    = 1;
    S c(i,i)
                     = -1;
    D c(i,i)
    D d(i,i)
                     = -1;
    D J DK RGK(i,i)
                    = 1;
   D_J_TK_RGK(i,i)
D_J_DK(i,i)
                    = 1;
                     = 1;
    D J min(i,i)
                    = 1;
    D sv(i,i)
                     = 1;
    D_ind_1(i,i)
                     = 1;
    D ind 2(i,i)
                     = 1;
    D HVP(i,i)
                     = 1;
    D Carlsborg(i, i) = 1;
                    = 1;
    D Ersbo(i,i)
                     = 1;
    D RGK(i,i)
    D HVK D(i,i)
                    = 1;
    D_HVK_F(i,i)
                     = 1;
    D HVK E(i,i)
                     = 1;
else
    S s(i,i-1)
                     = 1;
                     = -1;
    S_s(i,i)
                     = 1;
    S_c(i,i)
    D_c(i,i)
                     = -1;
    D d(i,i)
                     = -1;
    D J DK RGK(i,i)
                    = 1;
                    = 1;
    D J TK RGK(i,i)
                     = 1;
    D J DK(i,i)
                     = 1;
    D J min(i,i)
    D sv(i,i)
                     = 1;
    D ind 1(i,i)
                     = 1;
    D ind 2(i,i)
                     = 1;
    D HVP(i,i)
                     = 1;
    D Carlsborg(i, i) = 1;
    D Ersbo(i,i)
                    = 1;
    D RGK(i,i)
                    = 1;
    D HVK D(i,i)
                    = 1;
    D HVK F(i,i)
                     = 1;
    D HVK E(i,i)
                     = 1;
```

end

```
end
%Merge the matrixes
Aeq =
```

[S_start,S_s,S_c,S_d,S_J_DK_RGK,S_J_TK_RGK,S_J_DK,S_J_min,S_sv,...

```
S ind 1, S ind 2, S HVP, S Carlsborg, S Ersbo, S RGK, S HVK D, S HVK F, S HVK E;...
```

```
D_start,D_s,D_c,D_d,D_J_DK_RGK,D_J_TK_RGK,D_J_DK,D_J_min,D_sv,...
```

```
D ind 1,D ind 2,D HVP,D Carlsborg,D Ersbo,D RGK,D HVK D,D HVK F,D HVK E];
    Beq = zeros(n*n eq cons,1);
    %% Inequality constraints!
    n_ineq_cons = 1;
    %Charge constraint
    Y start
              = zeros(n,1);
                               %Initial value for storage unit
    Yз
               = zeros(n,n);
                              %Storage
    Yс
               = -eye(n,n);
                               %Charge/Discharge
    Υd
                               %Demand
                = zeros(n,n);
    Y J DK RGK = eye(n, n);
                                %Johannes DK RGK
    Y J TK RGK = eye(n, n);
                                %Johannes TK RGK
    YJDK
                               %Johannes DK
               = eye(n,n);
    Y J_min
               = eye(n,n);
                               %Johannes min load
                              %Waste heat
    Y sv
               = zeros(n,n);
    Y_ind_1
               = zeros(n,n);
                               %Evap 1
    Y_ind_2
                               %Evap 2
               = zeros(n,n);
    Y HVP
               = zeros(n,n);
                                %HVP
    Y_Carlsborg = zeros(n,n);
                                %Carlsborg
              = zeros(n,n);
                                %Ersbo
    Y Ersbo
    Y RGK
                = zeros(n,n);
                                %Flue gas cond.
    Y HVK D
               = zeros(n,n);
                                %Excess steam
    Y HVK F
               = zeros(n,n);
                                %Solid fuel
    Y HVK E
               = zeros(n,n);
                                %Electrical boilers
    %Merge the matrixes
    A = -[Y_start,Y_s,Y_c,Y_d,Y_J_DK_RGK,Y_J_TK_RGK,Y_J_DK,Y_J min,...
          Y sv,Y ind 1,Y ind 2,Y HVP,Y Carlsborg,Y Ersbo,Y RGK,Y HVK D,...
          Y_HVK_F,Y_HVK_E];
    B = -zeros(n*n ineq cons,1);
    %% Objective function
    f s
               = zeros(1, n+1);
                                                 %No cost for storage
    f c
                = zeros(1,n);
                                                 %No cost for
charge/discharge
    f d
                                                 %No cost for the demand
               = zeros(1,n);
    f DK RGK
               = ones(1,n).*J DK RGK pris(:)';
                                                 %Cost for Johannes DK RGK
    f TK RGK
               = ones(1,n).*J_TK_RGK_pris(:)';
                                                 %Cost for Johannes TK RGK
    f DK
               = ones(1,n).*J_DK_pris(:)';
                                                 %Cost for Johannes DK
                                                 %Cost for Johannes min
    f min
               = ones(1,n).*J Min pris(:)';
load
    f sv
               = ones(1,n).*SV pris(:)';
                                                  %Cost for Waste heat
    f ind 1
               = ones(1,n).*IND 1 pris(:)';
                                                  %Cost for Evap 1
    f ind 2
               = ones(1,n).*IND 2 pris(:)';
                                                  %Cost for Evap 2
    f HVP
               = ones(1,n).*HVP pris(:)';
                                                  %Cost for HVP
    f Carlsborg = ones(1,n).*Carlsborg pris(:)';
                                                 %Cost for Carlsborg
    f Ersbo
              = ones(1,n).*Ersbo pris(:)';
                                                  %Cost for Ersbo
    f RGK
               = ones(1,n).*RGK pris(:)';
                                                  %Cost for RGK BEAB
    f HVK D
               = ones(1,n).*HVK D pris(:)';
                                                  %Cost for excess steam
    f HVK F
               = ones(1,n).*HVK F pris(:)';
                                                 %Cost for solid fuel
    f HVK E
               = ones(1,n).*HVK E pris(:)';
                                                  %Cost for electrical
boilers
    %Merge all objectives into one objective function
```

f = [f_s,f_c,f_d,f_DK_RGK,f_TK_RGK,f_DK,f_min,f_sv,f_ind_1,...

```
f ind 2,f HVP,f Carlsborg,f Ersbo,f RGK,f HVK D,f HVK F,f HVK E];
    %% Lower and upper bounds
    Storage start = input.Ack start;
                                          %Set storage initial value
    lb start s
                = Storage start;
                                          %Force initial storage value
                 = Storage start;
                                          %Force initial storage value
    ub start s
                  = ones(1,n)*20;
                                                   %Lower bound for storage
    lb s
                  = ones(1,n)*350;
                                                   %Upper bound for storage
    ub s
    %Decide the end level for the storage unit, depending on the demand
    if mean(DH demand(1, n-6:n)) > 200
        1b s(1, n) = 200;
        ub s(1, n) = 200;
    elseif mean(DH demand(1, n-6:n)) > 150
        lb s(1,n) = 170;
        ub s(1, n) = 170;
    else
        lb s(1,n) = Storage start;
        ub s(1,n) = Storage start;
    end
                  = ones(1,n)*-40;
                                                  %Lower bound charging
    lb c
    ub c
                                                  %Upper bound charging
                  = ones(1,n)*40;
    lb d
                  = DH demand;
                                                  %Force demand
                   = DH demand;
                                                  %Force demand
    ub d
                  = zeros(1,n);
                                                   %Lower bound Johannes DK +
    lb J DK RGK
RGK
                   = ones(1,n).*J_DK_RGK_mp(:)'; %Upper bound Johannes DK +
    ub_J_DK_RGK
RGK
    lb_J_TK_RGK
                   = zeros(1,n);
                                                   %Lower bound Johannes TK +
RGK
    ub J TK RGK
                   = ones(1,n).*J TK RGK mp(:)'; %Upper bound Johannes TK +
RGK
    lb J DK
                   = zeros(1, n);
                                                   %Lower bound Johannes DK
                  = ones(1,n).*J_DK_mp(:)'; %Upper bound Johannes
= ones(1,n).*J_Min_mp(:)'; %Forces minimum load
= ones(1,n).*J_Min_mp(:)'; %Forces minimum load
    ub_J_DK
                                                   %Upper bound Johannes DK
    lb J min
                  = ones(1,n).*J Min_mp(:)';
                                                   %Forces minimum load
    ub_J_min
                  = zeros(1,n);
                                                   %Lower bound waste heat
    lb_sv
                  = ones(1,n).*SV_mp;
    ub_sv
                                                   %Upper bound waste heat
               = zeros(1,n);
= ones(1,n).*IND_1_mp;
= zeros(1,n).*
    lb ind 1
                                                   %Lower bound evap 1
    ub_ind_1
                                                  %Upper bound evap 1
                  = zeros(1,n);
    lb_ind_2
                                                   %Lower bound evap 2
    ub ind 2
                  = ones(1,n).*IND 2 mp;
                                                  %Upper bound evap 2
    lb_HVP
ub_HVP
                  = zeros(1,n);
                                                   %Lower bound HVP
                  = ones(1,n).*HVP_mp;
                                                  %Upper bound HVP
               - ones(1, n).*
= zeros(1, n);
= ones(1, n).*
    lb Carls
                                                   %Lower bound Carlsborg
                 = ones(1,n).*Carlsborg_mp;
    ub Carls
                                                  %Upper bound Carlsborg
                 = zeros(1,n);
    lb Ersbo
                                                   %Lower bound Ersbo
                 = ones(1,n).*Ersbo mp;
    ub Ersbo
                                                  %Upper bound Ersbo
                  = zeros(1,n);
    lb RGK
                                                  %Lower bound BEAB RGK
    ub RGK
                 = ones(1,n).*RGK mp;
                                                  %Upper bound BEAB RGK
                = zeros(1,n);
    lb HVK D
                                                  %Lower bound excess steam
    ub HVK D
                 = ones(1,n).*HVK D mp;
                                                  %Upper bound excess steam
    lb_HVK F
                 = zeros(1,n);
                                                  %Lower bound solid fuel
    ub_HVK F
                 = ones(1,n).*HVK F mp;
                                                  &Upper bound solid fuel
    lb HVK E
                  = zeros(1,n);
                                                  %Lower bound electric
boiler
    ub HVK E
                  = ones(1,n).*HVK E mp;
                                                  %Upper bound electric
boiler
```

%Merge all lower and upper bounds

lb=[lb_start_s,lb_s,lb_c,lb_d,lb_J_DK_RGK,lb_J_TK_RGK,lb_J_DK,lb_J_min,... lb_sv,lb_ind_1,lb_ind_2,lb_HVP,lb_Carls,lb_Ersbo,lb_RGK,... lb_HVK_D,lb_HVK_F,lb_HVK_E];

ub=[ub_start_s,ub_s,ub_c,ub_d,ub_J_DK_RGK,ub_J_TK_RGK,ub_J_DK,ub_J_min,... ub_sv,ub_ind_1,ub_ind_2,ub_HVP,ub_Carls,ub_Ersbo,ub_RGK,... ub_HVK_D,ub_HVK_F,ub_HVK_E]; %% Call the solver [x,fval]=linprog(f,A,B,Aeq,Beq,lb,ub);

APPENDIX 2. Mixed integer nonlinear programming

```
%% Retrieve all input data from the Excel spreadsheet
    DH demand
                     = input.demand;
                                                            %Heating demand
                      = Johannes output.DK mp; %Johannes DK available prod.
     J DK mp
     J DK RGK mp
                      = Johannes output.DK RGK mp; %Johannes DK + RGK available
prod.
                      = Johannes output.TK RGK mp; %Johannes TK + RGk available
    J TK RGK mp
prod.
    J_Min_mp = Johannes_output.Min_mp; %Johannes min load

SV_mp = SV_output.SV_mp; %Waste heat available prod.

IND_1_mp = IND_output.IND_1_mp; %Evap 1 available prod.

IND_2_mp = IND_output.IND_2_mp; %Evap 2 available prod.

HVP_mp = HVP_output.HVP_mp; %HVP available prod.

HVP_min = HVP_output.HVP_min; %HVP min load
     Carlsborg mp = Carlsborg_output.Carlsborg_mp;%Carlsborg available
prod.
    Carlsborg_min = Carlsborg_output.Carlsborg_min;%Carlsborg min load
     Ersbo_mp = Ersbo_output.Ersbo_mp; %Ersbo available prod.
    Ersbo_min= Ersbo_output.Ersbo_min;%Ersbo min loadRGK_mp= RGK_output.RGK_mp;%BEAB RGK availa
                                                                %BEAB RGK available
prod.
    HVK_D_mp= HVK_output.HVK_D_mp;%Excess steam available prod.HVK_F_mp= HVK_output.HVK_F_mp;%Solid fuel available prod.HVK_E_mp= HVK_output.HVK_E_mp;%Electric boilers available
prod.
    Name='Storage nonlinear';
     IntVars values = zeros(1, n*17+1); %For production units
     IntVars binary = ones(1,n*11); %For binary variables
     %Add regular variables and integers together
     IntVars = logical([IntVars_values,IntVars_binary]);
VarWeight = []; %No set priorities
    VarWeight
     88
     %Lower and upper bounds
    Storage_start = input.Ack_start; %Set storage initial value
lb_start_s = Storage_start; %Force initial storage value
ub start_s = Storage_start; %Force initial storage value
                     = Storage_start;
     lb s
                      = ones(1,n)*20;
                                                 %Lower bound for storage
                     = ones(1,n)*350; %Upper bound for storage
     ub s
     %Decide the end level for the storage unit, depending on the demand
     if mean(DH demand(1, n-6:n)) > 200
          lb s(1,n) = 200;
     elseif mean(DH demand(1, n-6:n)) > 150
         lb s(1,n) = 170;
     else
         lb s(1,n) = Storage start;
     end
                    = ones(1,n)*-40;
= ones(1,n)*40;
     lb_c
                                                             %Lower bound charging
                                                             %Upper bound charging
     ub c
    lb_d= DH_demand;ub_d= DH_demand.
                                                             %Force demand
     ub_d = DH_demand; %Force demand
lb_J_DK_RGK = zeros(1,n); %Lower bound Johannes DK + RGK
     ub J DK RGK = ones(1,n).*J DK RGK mp(:)'; %Upper bound Johannes DK +
RGK
     lb J TK RGK = zeros(1,n); %Lower bound Johannes TK + RGK
     ub_J_TK_RGK = ones(1,n).*J_TK_RGK_mp(:)'; %Upper bound Johannes TK +
RGK
```

lb J DK	= zeros(1,n);		%Lower bound Johannes DK		
np 'T DK	= ones(1,n) *	J DK mp(:)':	&Upper bound Johannes DK		
lb Tmin	= ones(1,n) *	T Min mp(·) ':	& Forces minimum load		
ub T min	$= \cos(1 n) *$		& Forces minimum load		
	= 200000(1, 11)	0_hiin_mp(.) ,	Slower bound waste best		
ub su	= 2005(1,11),		Subwer bound waste heat		
ub_sv	- ones(1,11).~	sv_mp;	Supper bound waste neat		
	= zeros(1,n);	1	%Lower bound evap 1		
ub_ind_1	= ones(1,n).*	IND_1_mp;	%Upper bound evap 1		
lb_ind_2	= zeros(1,n);		%Lower bound evap 2		
ub_ind_2	= ones(1,n).*	IND_2_mp;	&Upper bound evap 2		
lb_HVP	= ones(1,n).*	HVP_min;	%Lower bound HVP		
ub_HVP	= ones(1,n).*	HVP_mp;	%Upper bound HVP		
lb_Carls	= ones(1,n).*	Carlsborg_min;	%Lower bound Carlsborg		
ub_Carls	= ones(1,n).*	Carlsborg_mp;	%Upper bound Carlsborg		
lb_Ersbo	= ones(1,n).*	Ersbo_min;	%Lower bound Ersbo		
ub_Ersbo	= ones(1,n).*	Ersbo_mp;	%Upper bound Ersbo		
lb RGK	= zeros(1,n);	_	%Lower bound BEAB RGK		
ub_RGK	= ones(1,n).*	RGK mp;	%Upper bound BEAB RGK		
lb HVK D	= zeros(1,n);		%Lower bound excess steam		
ub_HVK_D	= ones(1,n).*	HVK D mp;	%Upper bound excess steam		
lb_HVK_F	= zeros(1,n);		%Lower bound solid fuel		
ub_HVK_F	= ones(1,n).*	HVK F mp;	&Upper bound solid fuel		
lb_HVK_E	= zeros(1,n);	+	%Lower bound electric boiler		
ub HVK E	= ones(1,n)	HVK E mp;	^{&} Upper bound electric boiler		
lb binarv	= zeros(1.n*1)	1):	&Lower bound binary variables		
ub binary	= ones(1, n*11));	⁸ Upper bound binary variables		
%Merge all	lower and upper	bounds			
x L=[]b start	s.lb s.lb c.lb d.	lb J DK RGK.lb	J TK RGK.lb J DK.lb J min		
lb sv	$_{2}, \underline{2}, \underline{2}$	2.1b HVP.1b Ca	urls.lb Ersbo.lb RGK.		
lb HV	//// /K D.lb HVK F.lb H	VK E.lb binary	/: /:		
. II- (b. ataxt		what Dr Dor wh	T MY DOW the I DY the I min		
x_0=[ub_start_	_s,up_s,up_c,up_d,	ub_J_DK_RGK, ub	_J_TK_RGK, UD_J_DK, UD_J_MIN,		
ub_sv		Z,UD_HVF,UD_Cd			
ин_аи	K_D, UD_HVK_F, UD_H	IVK_E, up_pinary	/];		
%% Linear	constraints				
∛storage c	constraint				
S_start	= zeros(n, 1);	%Initial	value for storage		
S_storage	= -eye(n,n);	%Storage			
S_charge	= eye(n,n);	%Charge			
S_zeros	= zeros(n,n);%Ot	her variables	not included in the constraint		
%Minimal u	%Minimal uptime/downtime for backup units				
Up_start	= zeros(n,1);	%Initial	value for storage		
Up_zero	= zeros(n,n);	%Variable	s not included in constraint		
Up_1	= zeros(n,n);	%Constrai	nt 1 for uptime		
Up_2	= zeros(n,n);	%Constrai	nt 2 for uptime		
Up 3	= zeros(n,n);	%Constrai	nt for first time step		
Down 1	= zeros(n,n);	%Constrai	nt 1 for downtime		
Down ²	= zeros(n,n);	%Constrai	nt 2 for downtime		
for $\overline{i} = 2$:	1:3				
Up 3(i	(i-1) = 1;	%Cc	onstraint for first time step		
up 3(i	(1,1) = -1;	%Cc	nstraint for first time step		
end - ·	, , , , , , , , , , , , , , , , , , ,		L		
%Ramp func	ctions				
R start	= $zeros(n,1);$	%Initial	value for storage		
R zero	= zeros(n.n);	%Variable	s not included in constraint		
R one	= eve(n,n);	%Constrai	nt for rampfunction		
R one (1.1)	- 0.	& Removes	constraint for first time sten		
	- U;	01/61/07/63	CONSCRATHE TOT TITSE FINE SIED		
	- 0;	%so it ca	n be any value at start		
for $i = 1$.	- 0; 1:n	%so it ca	in be any value at start.		

```
if i==1
                              S start(i, i)=1;
                       else
                              S storage(i,i-1)=1;
                       end
          end
          for i = 4:1:n
                              Up 1(i,i-3)=-1;
                               Up 1(i, i-2) = 1;
                              Up 1(i,i)=-1;
                               Down 1(i,i-3)=1;
                               Down_1(i,i-2)=-1;
                               Down 1(i,i)=1;
          end
          for i = 3:1:n
                              Up_2(i,i-2)=-1;
                              Up_2(i,i-1)=1;
                              Up 2(i,i)=-1;
                              Down 2(i, i-2) = 1;
                               Down 2(i, i-1) = -1;
                               Down 2(i,i)=1;
          end
          for i = 2:1:n
                    R one(i,i-1)=-1;
          end
          A = [S start, S storage, S charge, S zeros, S zeros, S zeros, S zeros, ...
%Storage
                                                S zeros, S zeros, S zeros, S zeros, S zeros, S zeros, ...
S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_zeros, S_z
S_zeros,S_zeros,S_zeros,S_zeros,S_zeros,S_zeros,S_zeros;...
Up start,Up zero,Up zero,Up zero,Up zero,Up zero,Up zero,Up zero,Up zero,...
. %HVP up 1
Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, ...
Up zero, Up zero, Up zero, Up zero, Up zero, Up 1, Up zero, Up zero,...
                                                Up zero, Up zero, Up zero, Up zero; ...
Up start, Up zero, ...
. %HVP up 2
Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, ...
Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, ...
                                                Up zero, Up zero, Up zero; ...
Up_start,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,...
. %HVP first time step uptime
Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, ...
Up zero, Up zero, Up zero, Up zero, Up zero, Up 3, Up zero, Up zero,...
                                                Up zero, Up zero, Up zero, Up zero;...
```

Up start, Up zero, %Carlsborg up 1 Up zero, ... Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, Up 1, Up zero,... Up zero, Up zero, Up zero; ... Up start, Up zero, %Carlsborg up 2 Up zero, ... Up zero, ... Up zero, Up zero, Up zero, Up zero; ... Up_start,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,... . %Carlsborg first time step uptime Up zero, ... Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, Up 3, Up zero,... Up zero, Up zero, Up zero, Up zero; ... Up start, Up zero, %Ersbo up 1 Up zero, ... Up zero, Up 1,... Up zero, Up zero, Up zero, Up zero;... Up start, Up zero, %Ersbo up 2 Up zero, ... Up zero, Up 2,... Up zero, Up zero, Up zero; ... Up start,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,.. . %Ersbo first time step uptime Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,... Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_3,... Up zero, Up zero, Up zero, Up zero;... Up start, Up zero, %HVP down 1 Up zero, ... Up zero, Up zero, Up zero, Up zero, Up zero, Down 1, Up zero, Up zero,... Up zero, Up zero, Up zero, Up zero;... Up start, Up zero, %HVP down 2

Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, ... Up zero, Up zero, Up zero, Up zero, Up zero, Down 2, Up zero, Up zero,... Up zero, Up zero, Up zero; ... Up start, Up zero, %Carlsborg down 1 Up zero, ... Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, Down 1, Up zero,... Up zero, Up zero, Up zero, Up zero;... Up start, Up zero, %Carlsborg down 2 Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,... Up zero, Up zero, Up zero, Up zero, Up zero, Up zero, Down 2, Up zero, ... Up zero, Up zero, Up zero, Up zero;... Up start, Up zero, %Ersbo down 1 Up zero, ... Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Down_1,... Up zero, Up zero, Up zero, Up zero;... Up start, Up zero, %Ersbo down 2 Up zero, ... Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Up_zero,Down_2,... Up zero, Up zero, Up zero; ... R start, R zero, ... %HVP +/-R_zero, R_zero, R_zero, R_zero, R_zero, R_zero, R_zero, R_zero, ... R zero, R_zero, R_zero, R_zero, R_zero, R_zero, R_zero, ... R_zero,R_zero,R_zero;... R start, R zero, ... %Carlsborg +/-R zero, R zero, R zero, R one, R zero, R zero, R zero, R zero, ... R zero,R zero,R zero,R zero,R zero,R zero,R zero,R zero,... R_zero,R_zero,R_zero;... R_start,R_zero,R_zero,R_zero,R_zero,R_zero,R_zero,R_zero,... %Ersbo +/-R zero, R zero, R zero, R zero, R one, R zero, R zero, R zero, ... R zero,R zero,R zero,R zero,R zero,R zero,R zero,R zero,...

```
%Upper and lower bounds for the linear constraints
    b L storage
                      = zeros(1,n);
                                                  %Storage
    b L up down start = ones(1, n*4*3+n*3)*-inf; %Backups
(HVP, Carls., Ersbo) uptime/downtime/start
    b L hvp
                      = ones(1,n)*-40;
                                                  %HVP +/-
    b L carls
                      = ones(1, n) * - 40;
                                                  %Carlsborg +/-
    b_L_ersbo
                      = ones(1, n) * - 40;
                                                  %Ersbo +/-
                     = b L storage; %Storage, '=' Enforces equality
    b U storage
                      = zeros(1,n*3*3); %Backups (HVP,Carls.,Ersbo)
    b U up start
uptime/start
    b U down
                     = ones(1, n*2*3); %Backups (HVP, Carls., Ersbo)
downtime
   b U hvp
                      = ones(1,n)*40;
                                        %HVP +/-
    b_U_carls
                      = ones(1,n)*40;
                                         %Carlsborg +/-
    b_U_ersbo
                      = ones(1,n)*40;
                                        %Ersbo +/-
    %Merge all lower and upper bounds
    b L = [b L storage,b L up down start,b L hvp,b L carls,b L ersbo];
    b U = [b U storage, b U up start, b U down, b U hvp, b U carls, b U ersbo];
    88
    %Nonlinear constraints
    c L demand = zeros(1,n);
                                   %Demand
    c L charge = ones(1,n)*-inf;
                                   %Charging
    c U demand = c L demand;
                                   %Demand
    c U charge = zeros(1,n);
                                   %Charging
    %Merge all lower and upper bounds
    c L = [c L demand, c L charge];
    c U = [c U demand, c U charge];
    88
    %Create initial guess from the lp optimization, setting binary variable
    %accordingly to optimization from lp.
    for i = 1:1:n
        %Johannes
        if Prognos(n*6+1+i) > 0
            Prognos(n*17+1+i) = 1;
        else
            Prognos(n*17+1+i) = 0;
        end
        %SV
        if Prognos(n*7+1+i) > 0
            Prognos(n*18+1+i) = 1;
        else
            Prognos(n*18+1+i) = 0;
        end
        %IND 1
        if Prognos(n*8+1+i) > 0
            Prognos(n*19+1+i) = 1;
        else
            Prognos(n*19+1+i) = 0;
        end
        SIND 2
        if Prognos(n*9+1+i) > 0
            Prognos(n*20+1+i) = 1;
        else
            Prognos(n*20+1+i) = 0;
        end
        %HVP
        if Prognos(n*10+1+i) > 0
```

R zero,R zero,R zero,R zero];

```
Prognos(n*21+1+i) = 1;
        else
            Prognos(n*21+1+i) = 0;
        end
        %Carlsborg
        if Prognos(n*11+1+i) > 0
            Prognos(n*22+1+i) = 1;
        else
            Prognos(n*22+1+i) = 0;
        end
        %Ersbo
        if Prognos(n*12+1+i) > 0
            Prognos(n*23+1+i) = 1;
        else
            Prognos(n*23+1+i) = 0;
        end
        %RGK BEAB
        if Prognos(n*13+1+i) > 0
            Prognos(n*24+1+i) = 1;
        else
            Prognos(n*24+1+i) = 0;
        end
        %HVK D
        if Prognos(n*14+1+i)>0
            Prognos(n*25+1+i) = 1;
        else
            Prognos(n*25+1+i) = 0;
        end
        %HVK F
        if Prognos(n*15+1+i) > 0
            Prognos(n*26+1+i) = 1;
        else
            Prognos(n*26+1+i) = 0;
        end
        %HVK E
        if Prognos(n*16+1+i) > 0
            Prognos(n*27+1+i) = 1;
        else
           Prognos(n*27+1+i) = 0;
        end
    end
    응응
    %Set intial guess to the solution from lp
        = Prognos;
    x_0
                        % One optimum known
    x opt = []';
    f opt = [];
                        % Value f(x opt), optimal cost is 0.
                        % Used for plotting, lower bounds
    x \min = [];
                        % Used for plotting, upper bounds
    x max = [];
    응응
   HessPattern = spalloc(n*28+1, n*28+1, 0); %All elements in Hessian are
zero.
    %Create the nonlinear constraint pattern matrix
    %Demand constraint
                                                    %Storage initial value
    D start = zeros(n, 1);
    D_storage = -eye(n,n);
                                                    %Storage
    D_charge = -eye(n, n);
                                                    %Charge
```

```
61
```

```
D \text{ demand} = -eye(n, n);
                                                     %Demand
    D prod
            = eye(n,n);
                                        %Production Units
    D x
             = eye(n,n);
                                        %Same for all binary variables
    %Complete the matrix for the storage unit
    for i = 1:1:n
         if i == 1
            D  start(i,i) = 1;
         else
            D storage(i, i-1) = 1;
         end
    end
    %Charge constraint
    C start = zeros(n, 1);
                                        %Storage initial value
    C storage = zeros(n,n);
                                        %Storage
    C charge = eye(n,n);
                                         %Charge
    C demand = zeros(n, n);
                                         %Demand
    C J prod = -eye(n,n);
                                         %Johannes production
    C \text{ prod} = \operatorname{zeros}(n, n);
                                         %Other units (not Johannes)
    C x J
             = eye(n,n);
                                         %Binary variable for Johannes
    C x = zeros(n, n);
                                         %Binary variables for other units
    %Create the nonlinear constraint pattern.
    ConsPattern =
[D start,D storage,D charge,D demand,D prod,D prod,D prod,...
                    D_prod, D_prod, D_prod, D_prod, D_prod, ...
                    D_prod, D_prod, D_prod, D_prod, D_x, D_x, D_x, ...
                    D_x, D_x, D_x, D_x, D_x, D_x, D_x; \dots
C_start,C_storage,C_charge,C_demand,C_J_prod,C_J_prod,C_J_prod,C_J_prod,C_p
rod,...
                    C_prod,C_prod,C_prod,C_prod,C_prod,C_prod,...
C_prod,C_prod,C_x_J,C_x,C_x,C_x,C_x,C_x,C_x,...
                    C \times, C \times, C \times, C \times];
    응응
    fIP = []; % An upper bound on the IP value wanted. Makes it possible
    xIP = []; % to cut branches. xIP: the x value giving fIP
    88
    %Setup the problem according to TOMLAB syntax
    Prob = minlpAssign('minlpQG_f', 'minlpQG_g', 'minlpQG_H', HessPattern,
. . .
                       x L, x U, Name, x 0, ...
                       IntVars, VarWeight, fIP, xIP, ...
                       A, b L, b U, 'minlpQG c', 'minlpQG dc',
'minlpQG d2c', ...
                       ConsPattern, c_L, c_U, ...
                       x_min, x_max, f_opt, x_opt);
    Prob.DUNDEE.optPar(20) = 1;
    Prob.P = 1; % Needed in minlpQG xxx files
    %Call the function, MINLP-BB
    Result = tomRun('MINLPbb', Prob, 2);
```

Appendix 3. Nonlinear constraints (MINLP)

```
%Non-linear constraints
function c = minlpQG c(x, Prob)
     %Get number of time steps, n.
     n = evalin('base', 'n');
     %Define matrixes
     A1 = zeros(n, 1);
     A2 = zeros(n, 1);
 for i = 1:1:n
        %Demand constraint
        A1(i,1) = -x(n*1+1+i) -
x(n*2+i+1)+(x(n*3+i+1)+x(n*4+i+1)+x(n*5+i+1)+...
x(n*6+i+1))*x(n*17+i+1)+x(n*7+i+1)*x(n*18+i+1)+x(n*8+i+1)*x(n*19+i+1)+...
x(n*9+i+1)*x(n*20+i+1)+x(n*10+i+1)*x(n*21+i+1)+x(n*11+i+1)*x(n*22+i+1)+...
x (n*12+i+1) * x (n*23+i+1) + x (n*13+i+1) * x (n*24+i+1) + x (n*14+i+1) * x (n*25+i+1) + ...
                  x(n*15+i+1)*x(n*26+i+1)+x(n*16+i+1)*x(n*27+i+1);
        %Charge constraint
        A2(i,1) = x(n*1+1+i) - (x(n*3+i+1)+x(n*4+i+1)+x(n*5+i+1)+...
                  x(n*6+i+1))*x(n*17+i+1);
 end
c = [A1; A2];
```

Appendix 4. Objective function (MINLP)

```
%Objective function
function f = minlpQG f(x, Prob)
     %% Retrieve needed data
                       = evalin('base', 'n');
     n
     Priser output = evalin('base', 'Priser output');
     J DK pris = Priser output.J DK;
     J DK RGK pris = Priser output.J DK RGK;
     J_TK_RGK_pris = Priser_output.J_TK_RGK;
     J_Min_pris=Priser_output.J_Min;SV_pris=Priser_output.SV;IND_1_pris=Priser_output.IND_1;IND_2_pris=Priser_output.IND_2;HVP_pris=Priser_output.HVP;
     Carlsborg pris = Priser output.Carlsborg;
     Ersbo_pris = Priser_output.Ersbo;
RGK_pris = Priser_output.RGK;
     RGK_pris = Priser_output.RGK;
HVK_D_pris = Priser_output.HVK_D_pris;
HVK_F_pris = Priser_output.HVK_F_pris;
HVK_E_pris = Priser_output.HVK_E_pris;
     %% Actual cost, dependent if the plant is operating or not
     T_start = 0;

Tank = zeros(1,n);

Charge = zeros(1,n);

D = zeros(1,n);
     J DK
                  = J DK pris.*x(n*17+2:n*18+1,1);
     J DK RGK = J DK RGK pris.*x(n*17+2:n*18+1,1);
     J_TK_RGK = J_TK_RGK_pris.*x(n*17+2:n*18+1,1);
     J_min = J_Min_pris.*x(n*17+2:n*18+1,1);
     SV
                  = SV pris.*x(n*18+2:n*19+1,1);
     IND_1 = IND_1_pris.*x(n*19+2:n*20+1,1);
IND_2 = IND_2_pris.*x(n*20+2:n*21+1,1);
HVP = HVP_pris.*x(n*21+2:n*22+1,1);
     Carlsborg = Carlsborg pris.*x(n*22+2:n*23+1,1);
     Ersbo = Ersbo pris.*x(n*23+2:n*24+1,1);
                 = RGK_pris.*x(n*24+2:n*25+1,1);
     RGK
     HVK_D = HVK_D_pris.*x(n*25+2:n*26+1,1);
HVK_F = HVK_F_pris.*x(n*26+2:n*27+1,1);
HVK_E = HVK_E_pris.*x(n*27+2:n*28+1,1);
     x binary = zeros(1,n*11);
%% Merge the objective function
f=[T start,Tank,Charge,D,J DK RGK',J TK RGK',J DK',J min',SV',IND 1',...
```

IT_start,Tank,Charge,D,J_DK_RGK',J_TK_RGK',J_DK',J_min',SV',IND_1',... IND_2',HVP',Carlsborg',Ersbo',RGK',HVK_D',HVK_F',HVK_E',x_binary]*x;

Box 883, 721 23 Västerås **Tfn:** 021-10 13 00 Box 325, 631 05 Eskilstuna **Tfn:** 016-15 36 00 **E-post:** info@mdh.se **Webb:** www.mdh.se