Optimizing supply temperatures in district heating grids

Master Thesis

Fianne Stoel

Delft University of Technology

Optimizing supply temperatures in district heating grids

by

Fianne Stoel

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday July 13, 2022 at 10:00 AM.

Student number:	4605519	
Project duration:	November 8, 2021 – July 13, 2022	
Thesis committee:	Dr. M. M. de Weerdt,	TU Delft, supervisor
	Dr. D. M. J. Tax,	TU Delft
	R. Everhardt, MSc.	Flex Technologies, external expert

Preface

This thesis marks the end of my Master in Computer Science. It has been done in in collaboration with Flex Technologies. They have provided the simulator for district heating systems that I have worked with, but also have helped me a lot with my experiments.

All in all, I have really enjoyed working on this thesis. It started off a bit rocky with the literature research on the subject, but once we decided on the optimization approach I would be working on, things got going properly. Although I would have liked to do more, I'm happy with the results that I have gotten.

I would like to thank my thesis advisor, Mathijs de Weerdt, for his guidance during this project. Thank you for thinking along and giving me suggestions for the next steps in my research and for taking the time to give me feedback. Furthermore, I would like to thank Ksenija Stepanovic for meeting with me weekly and answering the sometimes very intricate questions on the subject. Lastly, I would like to thank Rob Everhardt and Jichen Wu from Flex Technologies for helping me understand the workings of district heating systems and for thinking along and checking my reasoning.

Fianne Stoel
Delft, July 2022

Abstract

District heating systems (DHSs) have the potential to play a big part in the energy transition. The efficient operation of DHSs is therefore also an important subject of study. The operation of DHSs where combined heat and power (CHP) plants are used are particularly interesting, because CHPs can operate with high efficiency.

In this work, the operational optimization of DHSs with CHP plants is considered. Determining the optimal heat and electricity production for CHPs for multiple time steps into the future is a complex problem. Because of the heat storage capabilities in the network many solutions are feasible, but determining which solutions are infeasible because of constraint violations in the DHS involves computing time delays that depend on complex network dynamics.

In this work, the possibility of using an input convex neural network (ICNN) to learn the network dynamics of a DHS is explored. ICNNs have limitations on their learning capabilities, but theoretically allow for easier optimization. Experiments on the learning capabilities of ICNNs reveal that caution should be used when they are used to learn non-convex constraints, as the accuracy of the ICNN highly depends on how non-convex the function is. Experiments on the feasible space of supply temperatures to a small district heating network (DHN) reveal that although the network does not provide the same flexibility as heat storage tanks, still some flexibility in the operation can be found. This is due to the fact that water with a higher supply temperature is consumed by consumers at a slower pace and this increases the time delay between production and consumption. Supply temperatures that follow can then be lowered if the increased time delay causes this water to arrive when the heat demands are lower.

In the experiments it was found that this flexibility in operation translates to non-convex areas in the feasible space. When this space would be learned by an ICNN, this space would be made convex. How much of the flexibility would be removed by doing this is yet unknown and could be researched in future work. Other future work can be done on safely learning non-convex constraints with an ICNN.

Contents

1

Introduction

The Netherlands is working towards being climate neutral. In order to reach our goals of reducing the emission of greenhouse gases, plans are being made for the energy transition. In a study on possible future scenarios for climate neutral energy, all scenarios feature the usage of district heating systems (DHSs) [1]. DHSs are systems that can provide heat to residential and other buildings often with less environmental impact than conventional heating systems [2]. In these systems heat is produced in a central location and is then distributed to consumers using heated water that is transported through insulated pipes. Several types of heat producers can be used in DHSs, of which the combined heat and power (CHP) plant is one. CHP plants produce thermal power (heat) and electrical power with higher efficiency than condensing power plants [2].

The optimization of the operation of CHP plants in DHSs has been a subject of study in the past years [3]–[7], because of its complexity. Firstly, in order to reduce the operation costs of the CHP, it is beneficial to produce the least amount of heat that is needed to fulfil the heat demands of the consumers. Determining the minimal heat production that can satisfy the heat demands is already difficult because of the complexity of the network dynamics [8]. This is because the computation of the time delays between the water being heated and being used by consumers is complex, and the delays can sometimes differ multiple hours depending on the network. Secondly, the power production of the CHP can be optimized. This optimization involves producing more electrical energy in time periods when it is most needed or most profitable, and less otherwise. This also influences the heat production that is possible for the CHP, since the amount of heat and electrical power that can be produced depend on each other. Therefore the optimization that is possible for heat production and for electrical power production depend on each other.

What allows for more flexibility in the operational optimization of a DHS is the fact that the pipes in the network are insulated well and can function as heat storage. This network storage effect can be used, for example, to insert more heat into the district heating network (DHN) when it is available, which is stored and can then be discharged at a later time during a peak in heat demand [9]. This method provides flexibility in the planning of the heat production, since there is a possibility to produce heat in advance.

In one of the earlier works on the optimization of the heat production in a DHS, a model of the network is created that accurately computes heat losses and time delays in the network [8]. However, the optimization of the operation of CHPs in a DHS that includes this model becomes a mixed-integer nonlinear programming (MINLP) problem [4]. In this thesis, the usage of input convex neural networks (ICNNs) in this optimization is considered. ICNNs are neural networks and can be trained on the complex network dynamics of a DHS. However, ICNNs differ from regular feedforward neural networks in their architecture, as they are restricted such that they can only learn convex functions. Therefore if the network dynamics learned by an ICNN are used in the optimization problem, it is expected its convexity simplifies the optimization.

In this thesis, the potential of using ICNNs to learn the network dynamics of a DHS with one CHP plant is explored, in order for it to be used in the optimization of the supply temperatures to the DHN. Therefore the research question discussed in this work is: How well can the feasible space of supply temperatures of a district heating system be learned by an input convex neural network?

The difficulty of this question lies in the fact that the feasible space of supply temperatures that needs to be learned is non-convex, but input convex neural networks (ICNNs) can only learn convex functions. It is therefore known that an ICNN will not be able to learn non-convex regions of the feasible space accurately. What is not known, however, is how well an ICNN can estimate non-convex functions and therefore how big the errors in the learned space can be. It is also not known what the feasible space of the supply temperatures at the CHP exactly looks like and therefore also if these non-convexities are in regions that are relevant during optimization or not.

Based on these subproblems, the following subquestions are formed:

- How well can an ICNN learn convex and non-convex functions?
- What does the feasible space of the supply temperatures of a DHN look like?

The rest of this thesis is structured in the following way: first in chapter 2 the background material is covered that is needed to understand the workings of DHSs. Second, the relevant literature on both the optimization DHSs and the usage ICNNs is covered in chapter 3. After that, the subquestions of this research are covered. In chapter 4 the capabilities of ICNNs to learn convex and non-convex functions are tested. Then in chapter 5 the feasible space of the supply temperatures of a DHN is investigated. Lastly, in chapter 6 all research questions are answered and suggestions are done for future work.

2 Background

In this chapter the necessary background material is covered for this thesis.

Generally, the main components of a District Heating System (DHS) are heat producers, consumers and a District Heating Network (DHN). The DHN consists of interconnected pipelines, where the pipes in which the heated water flows towards the consumers are called supply pipes and the pipes in which the water flows back towards the producers are called return pipes. In DHSs often consumers are not directly connected to the same network that producers are connected to. Heat exchangers are then used to transfer heat from the primary network, that producers are connected to, to a secondary or even tertiary network, that the consumer is then connected to. In this work, DHSs with a primary and secondary network are considered. However, the operation of the secondary network is not covered.

A DHS functions in continuous time. However, in order to simulate and optimize the way it functions, it is often easier to divide time into time steps. The system can then be described for every time step, thus in discrete time. The chosen size of the time steps can depend on the precision required to describe the system and the computing power available when making a model of the system.

Furthermore, in order to reason about the behaviour of the DHS, it is often needed to keep track of masses of water that are heated and inserted into the DHN in one time step. Therefore in this work, the mass of water heated and inserted into the network during one time step is given a name: a *plug*. $plug_i$ denotes the plug that is inserted into the network during time step i.

The rest of this chapter is organised in the following way. First the components of a DHS are described in more detail. After that, some of the complex behaviour of DHSs is highlighted.

2.1. Producer

In a DHS different kinds of heat producers can be used. In this work, DHSs with Combined Heat and Power (CHP) plants are considered. In one time step, these plants can produce an amount of thermal power (heat) and electrical power that is within its operating region. Operating regions of specific CHP plants can vary. Figure 2.1 is an example of one such operating region by Stepanovic et al. [7], based on the work of Abdollahi et al. [10] and Bloess et al. [11]. In this image the points within the polygon are valid configurations of heat production H and power production P. Therefore at each time step, a CHP can be configured to produce a variable amount of heat and power.

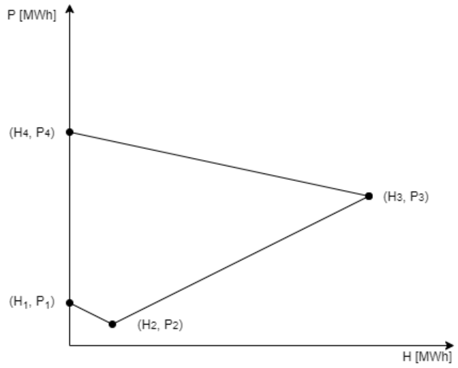


Figure 2.1: An operating region of a CHP [7].

Alternatively, it is possible to operate a CHP by providing the temperature of the plug instead of the heat production. The heat production and temperature setting have a relation according to Equation 2.1, where h_t is the heat produced at time step t, c is the heat capacity of water, m_t is the mass flow at time step t and τ_t^{supply} and τ_t^{return} are the outlet and inlet temperatures of the CHP at time step t respectively. If the return temperature and the mass flow at one time step are known, then the output temperature of a CHP can be converted to the heat production.

$$h_t = c \cdot m_t \cdot (\tau_t^{supply} - \tau_t^{return}), \quad \forall t \tag{2.1}$$

2.2. Network

The network of a DHS consists of a set of interconnected pipes, connecting the producers and the consumers along supply and return pipes. The pipes in this network can branch or merge at locations where it is beneficial, given the locations of the consumers. It is possible for pipes with different diameters to be used at different locations in this network, depending on the consumer load expected to flow through the pipe.

The physical properties of these pipes have an effect on how the DHS runs. For instance, the insulation of the pipes have an effect on the heat loss of the water flowing through and the diameter affects the maximum mass flow for which pipes function properly.

Pumps in the network provide enough differential pressure to ensure a large enough mass flow for all consumers. Pressure changes in the network spread about 1000 times faster than temperature changes [8], which makes the correct modelling of this relatively much less important for the optimization. It can therefore be assumed that pressure and mass flow changes have an instant effect in the DHN.

2.3. Heat exchangers and consumers

The amount of heat that heat exchangers transfer into the secondary network depends on the heat demand of consumers. This heat demand can fluctuate throughout the day and throughout the seasons, and therefore the amount of heat needed from the primary network to fulfil this demand differs.

The way that heat exchangers are able to regulate the amount of heat extracted from the primary network is by operating the valves on the primary side that restrict the mass flow. The extracted heat at time step t also follows the relation of Equation 2.1, where changing the mass flow m_t has an effect on both the return temperature τ_t^{return} and the heat h_t. This effect can be computed [12], but will not be elaborated upon in this work. The supply temperature τ_t^{supply} of the plug at the consumer depends on the supply temperature at the CHP when this plug was heated at some earlier time and the heat loss of this plug during transportation. Therefore, for one supply temperature at the heat exchanger, varying the configuration of the valves allows for different amounts of heat to be extracted from the network. This allows for some flexibility in the operation of the network; the heat demands of consumers can be satisfied with various supply temperatures and therefore multiple configurations of heat production by a CHP.

Although heat exchangers are able to satisfy the heat demands of consumers with flexibility, there are still restrictions on the operation. In order to extract enough heat from the network for lower supply temperatures, the valves of the heat exchanger are opened up, which increases the mass flow. However, as mentioned in section 2.2, there is a maximum mass flow for pipes in the network. Below a certain supply temperature it is possible that in order to fulfill the heat demand a mass flow is required that is above the maximum allowed. Therefore, there is a minimum supply temperature for a heat exchanger to be able to satisfy a given heat demand.

As mentioned before, the way that the secondary network is operated is not described in this work. It is assumed that each time step the heat exchangers need to insert an amount of heat into the secondary network based on the heat demands of the consumers. In this work, the heat exchangers and secondary network are sometimes left out when reasoning about the way that the DHS works. In this scenario, the consumer is used to refer to the heat exchanger that works on behalf of them.

2.4. Complexity in district heating systems

As explained in the previous section, heat exchangers are flexible in their operation; the heat demand of consumers can be satisfied for a range of supply temperatures. This flexibility comes from the ability of heat exchangers to adjust the mass flow using valves in the pipes in order to extract the required amount of heat. In a DHN with heat exchangers at multiple locations, the mass flows in the pipes at these locations can be individually adjusted according to the demands of consumers. The mass flows at producers depend on all mass flows at the heat exchangers.

Although these individual mass flows make the operation of the network more flexible, the impacts of these configurations do complicate the dynamics of the network. Three examples of complexities will be described.

Firstly, because of the varying mass flows over time, there is no fixed amount of time it takes for a plug to arrive at a heat exchanger. This time delay depends on the temperatures of water in the supply pipes at the time that the plug is heated at the producer. So in order to determine the minimal heat production during one time step that is able to satisfy the demands of when it arrives, the mass flows of all time steps before the arrival of this plug have to be determined. Only then is it known during which time step the plug arrives at the heat exchanger and the amount of heat that is lost during transportation.

Secondly, determining the required heat production during one time step also becomes more difficult when considering multiple heat exchangers in the network. The arrival times of the water at these heat exchangers can differ because of different positions in the network and different heat demands of consumers. Even the way that the water is divided when supply pipes branch towards the different heat exchangers can vary over time, as the heat exchanger further downstream in each branch decides on the mass flow for that branch with its primary side valve.

Lastly, the mass flows in the pipes at the heat exchangers together determine the mass flow at the producer, which has an effect on the possible configurations. As can be seen in Equation 2.1, if the supply and return temperatures are kept the same then an increase in mass flow would result in more heat being produced. An increased mass flow with the same supply and return temperatures can also cause the amount of heat produced to increase to a point that it is outside of the operating region of the CHP. In this scenario it would be required to lower the supply temperature in order for the CHP to work correctly. However, this constraint on the supply temperature could also have an effect on the feasibility of some future heat demand.

As illustrated with these examples, the way that a DHS operates over time is complex. Setting the supply temperature of one time step has effects in the network many time steps later and it is difficult to determine for which supply temperatures the grid runs properly. Although these complex dynamics make the optimization of the operation of the DHS difficult, there are approaches that have been developed to tackle this problem.

3 Related Work

In this chapter the relevant literature is discussed for this thesis. Firstly, existing work about the operational optimization of DHSs is discussed in section 3.1. Mostly approaches that include a way to incorporate the network storage capabilities into the optimization are covered here, as this is relevant for the optimization approach that is considered in this thesis. Secondly, the literature introducing ICNNs is covered in section 3.2. Works where ICNNs are used to learn the dynamics of physical systems are also discussed.

3.1. Operational optimization of district heating systems

Benonysson [8] has formulated a mathematical model of the operation of DHSs. Included in this model is a newly introduced method called the node method that is used to model the temperature dynamics of the network. This part of the model for the DHS is said to be the most difficult part of the optimization, because the time delays in the network depend on the supply temperatures. Benonysson also proposes and tests two approaches to the optimization of the model. The objective function of this optimization consists of the operating costs of the producers and the costs of the pumps for all time steps. The approach using an iterative optimization showed promising results. This method performs iterations of first evaluating the time delays in the network for given supply temperatures and then optimizing the supply temperatures for the given time delays.

Although it is concluded from case studies that the iterative method can perform well, there are some drawbacks to using this method. Due to the iterative process, the method can run into convergence problems. Moreover the solution that is a result of the optimization is a local optimum and has no guarantee of being close to the global optimum.

Li et al. [4] have considered an iterative approach to the operational optimization of DHSs with CHPs. In this work, the optimization of the operation of the CHPs includes a term for wind power utilization. This term ensures that the planning of the CHP power production is optimized such that the least amount of wind power goes unused.

In order to minimize the spillage of energy, flexibility in the operation of CHPs is needed. Li et al. [4] attempt to gain flexibility by using the heat storage effect of DHNs. This is an effect where heat is added to the network by CHPs and is stored in the insulated pipes until it arrives at heat exchangers. When storing more heat in the network than strictly needed at that time step, this heat can be used to fulfil the heat demands of consumers of more than one time step. This effect introduces flexibility for the heat production: either the minimum amount of heat that fulfills the heat demand can be produced, or more heat can be produced and stored in the network, which can be used to fulfill the heat demand of multiple time steps. Using the heat storage capabilities of the network can introduce higher operating costs as the heat loss of water increases for higher temperatures and therefore in the end more heat will have to be produced. However, the power consumption of pump will be lower, as the mass flow of the network is lower when higher supply temperatures are consumed. Whether the heat production utilizing the network heat storage is favoured over the heat production that does not utilize the storage depends on the difference in operating cost and which solution allows an electricity production that causes the least spilled wind energy.

Li et al. [4] also formulate a mathematical model of the operation of DHSs, which includes the node method [8]. The operational optimization of the DHS is then considered with an objective function that minimizes the operating costs of the CHPs and the penalties of wind power spillage. The decision variables include heat and power production of the CHPs, mass flow rates and temperature profiles in the supply and return pipes and variables for the pumps and pressure in the network. The resulting formulation is a large-scale mixed-integer nonlinear programming (MINLP) problem. In order to solve this optimization problem, a method similar to the iterative solution of Benonysson [8] is used.

The experiments that are done with the proposed iterative algorithm by Li et al. [4] indicate that it has a good performance. However, since the itererative method is similar to the method used by Benonysson [8], the same drawbacks can be expected.

Giraud et al. [3] have optimized the operation of DHSs with a focus on the usage of network storage. Using the heat storage of the network introduces flexibility, but can introduce higher costs because of the higher heat loss. Using lower supply temperatures reduces the heat loss, but pumping costs increase as the mass flow increases. It is the goal of Giraud et al. to find a compromise between pumping costs and heat losses.

In the problem definition of this paper, a Model Predictive Control (MPC) scheme is used. Using this scheme, for each time step the optimization problem is solved for an amount of time steps called the predictive horizon. From this solution, only the first few output values are applied. The amount of output values applied is called the receding horizon. After the application, the problem is then again solved for the next predictive horizon.

Giraud et al. [3] optimize the DHS using an iterative method. A detailed nonlinear model of the DHS that is partially based on the node method [8] is used to simulate the dynamics of the distribution network for given supply temperatures and differential pressures at production plants. The results from the simulation are used to formulate a linearized model of the distribution network, which is used to make a mixed integer linear program (MILP). The MILP is optimized and the resulting supply temperatures and differential pressures are used to perform the next iteration.

The proposed approach is evaluated with a case study. The results indicate that the supply temperature is often minimized due to low pumping costs in the case study. In scenarios where there is a peak heat demand, the network storage capabilities are used in anticipation. Although the results indicate an improved performance, there could be disadvantages to this approach. Since an iterative approach is used, convergence problems can arise and the algorithm could get stuck in a local optimum. Furthermore, since a linear relaxation of the problem is solved, the optimal solution could be infeasible.

Merkert et al. [5] have solved the operational optimization of DHSs with an approach that also focuses on the heat storage of the DHN. In this approach the optimization is simplified by avoiding a detailed model of the physical grid and instead using a function for the heat energy stored in the grid. This function is based on the past and present supply temperatures and uses time delays to the consumers and their shares of consumption. These time delays and shares of consumption are assumed to be constant. Furthermore, thermal losses in the grid are not considered. An operational optimization problem is formulated using this model of the grid. This problem formulation is a MILP, which is solved to find the optimal supply temperatures.

The results from a case study suggest that the proposed approach is very computationally efficient and is able to reduce the production costs. However, the approach might not perform as well in scenarios that are very different from the case study. The assumed constant time delays performs well when the supply temperature is constant or only has small and short increases. If larger and longer supply temperature increases are used, the model of the network becomes more inaccurate. Furthermore, in the case study a network with only one pipe with a large diameter is used, which makes the heat loss negligible. In larger networks the heat loss becomes more relevant.

Stegman [6] has optimized the supply temperatures of DHSs using population-based metaheuristics. Instead of optimizing a mathematical model of the problem, these methods search for the optimal solution by exploring the solution space. For the current problem, a solution is a list supply temperatures for time steps in the predictive horizon.

In order to guide the search process to good parts of the search space, a function is needed to evaluate solutions. Stegman [6] evaluates solutions by taking the operating costs of the producers and adding penalties for constraints that are violated in the network. The latter is needed in the evaluation, since the whole solution space can also include low cost solutions that violate constraints in the network. A simulator of the DHS is used to determine if a solution violates any of the constraints of the DHS

The results from experiments indicate that significant savings can be made on the operation of the DHS compared to the current method of operation. The network storage effect was also used in the solutions that were found. However it was also stated that the optimization is slow in convergence and that the simulator slowed down significantly for bigger networks. Another concern for using this approach is that constraint violations are included as a penalty term in the solution evaluation and that there is no guarantee that the optimal solution found does not violate the network constraints.

In order to overcome the complexities of the exact mathematical modelling and optimization, Stepanovic et al. [7] have recently proposed and analysed the usage of a reinforcement learning (RL) algorithm for the operational optimization of a CHP in a DHN. The RL environment is formulated as a Markov decision process (MDP) and the RL algorithm is trained by interacting with a simulator of the physics processes in the DHS. The reward function of the RL environment incorporates profit and, if safety constraints are violated, negative rewards. Experiments that compare the RL algorithm to the MILNP optimization by Li et al. [4] are done on a DHN with one producer and one consumer. The RL algorithm has a lower profit gain than the MILNP optimization, but results in fewer constraint violations.

Although the RL algorithm results in fewer constraint violations than the MILNP optimization in the experiments, one of the major concerns when using RL for the operational optimization of DHSs are violations of constraints in the network. The safety constraints in the network are implemented in the RL algorithm as soft constraints in the reward function and it can therefore not be guaranteed that these constraints are not violated. Furthermore, Stephanovic et al. [7] express concern about the scalability of the RL algorithm, as the state space will grow exponentially when the DHN increases in size.

To summarize, detailed mathematical models of the operation of DHS are non-linear because of complicated network dynamics [4], [8]. Therefore the approaches that have been discussed all try to overcome this complexity in the optimization using various methods. Although the case studies and experiments done on these approaches indicate an improvement in performance, these approaches do all have possible disadvantages. Iterative approaches as proposed by Benonysson [8], Li et al. [4] and Giraud et al. [3] can run into convergence problems and will find a locally optimal solution. Approaches that use a MILP to optimize the solution like Giraud et al. [3] and Merkert et al. [5] require assumptions in order to linearize the optimization, which could cause infeasible solutions to be found. Approaches using population based metaheuristics as proposed by Stegman [6] have safety concerns because the constraints are enforced by including negative rewards. RL approaches as proposed by Stepanovic et al. [7] have similar safety concerns and might not be scalable.

In this thesis, a new approach for the operational optimization of DHSs is considered. This approach tries to overcome the complexity of the network dynamics by using ICNNs.

3.2. Input convex neural networks used for physical systems

The concept of ICNNs was first introduced by Amos et al. [13]. In this work constraints on the parameters of a neural network are defined that cause the output of the network to be a convex function of the input. The architecture of a fully input convex neural network (FICNN) is based on a fully connected neural network on input x with k layers with skip connections as displayed in Figure 3.1. In this network the layer activations z_i for $i = 0, ..., k-1$ are defined as:

$$z_{i+1} = g_i(W_i^{(z)} z_i + W_i^{(x)} x + b_i) \tag{3.1}$$

where $W_i^{(z)}$ and $W_i^{(x)}$ are weights for the previous layer and the skip connection respectively, b_i are biases, g_i are activation functions and z_0, $W_0^{(z)} \equiv 0$. $\theta = W_{0:k-1}^{(y)}, W_{1:k-1}^{(z)}, b_{0:k-1}$ are the parameters. The function $f(x;\theta) = z_k$ is convex in input x given that $W_{1:k-1}^{(z)}$ are all non-negative and g_i are all convex and non-decreasing.

The advantage of restricting the output of the neural network to be a convex function is that theoretically, the input can be optimized easier. Amos et al. [13] discuss a few possible approaches for this

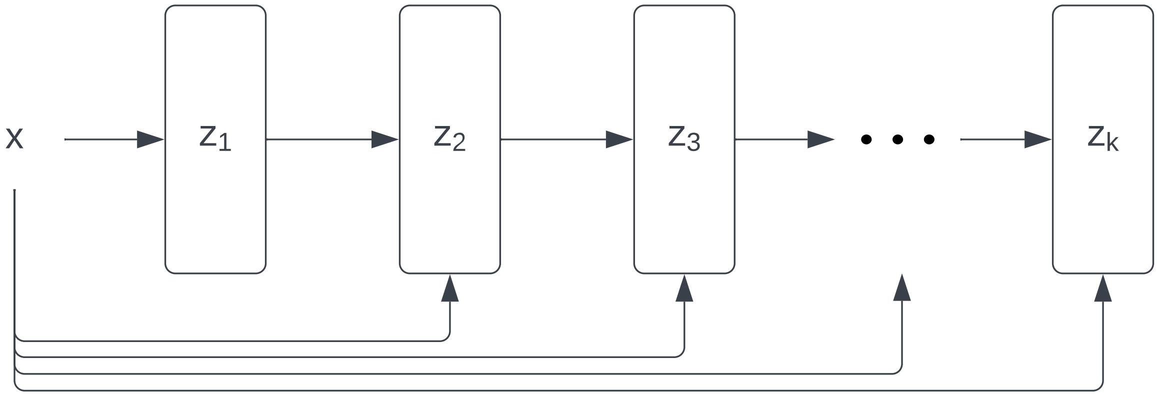


Figure 3.1: The architecture of a fully input convex neural network (FICNN)

optimization. Exact optimization of the input of the ICNN by posing it as a linear program is said not to be a viable option for the optimization of the network. Alternatively, Amos et al. [13] have developed an approximate optimization method called the bundle entropy method. This method is a gradient-based approach and uses backpropagation to compute the gradient of the ICNN with respect to its inputs. With experiments it is shown that the bundle entropy method outperforms gradient descent on the input.

Chen et al. [14] make an adaptation to the work of Amos et al. [13] in order to optimize the control of complex systems. An input convex recurrent neural network (ICRNN) is designed that is able to learn the temporal behaviour of dynamical systems. Gradient-based optimization of the input of the ICRNN can be included in the MPC problem that determines the best control decision. Experiments on the optimization of a building energy management problem are done. For this MPC problem two ICRNNs are used: one for the objective function and one for the constraints on the building states. Results indicate that the optimization using the ICRNNs performs better than the alternative where conventional recurrent neural networks (RNNs) are used.

Bünning et al. [15] also adapt the architecture of an ICNN as defined by Amos et al. [13] in order to use it in building MPC. In this work, however, a feedforward ICNN is used and predictions are made by repeatedly evaluating the network with the output from the previous time step. This makes for a simpler network architecture compared to the ICRNN that is used by Chen et al. [14], but it also limits the ”memory” of the network since only the output of the precious time step is used as an input. In two experiments on building energy management, the controller using the ICNN was able to keep the room temperature within comfort constraints.

Yang and Bequette [16] have used ICRNNs as defined by Chen et al. [14] to solve an optimization-based control problem in a case study. It is shown that even though a normal neural network is able to approximate a model of the physical system better, the controller using the ICRNN is able to achieve a better economic performance in an unconstrained MPC problem. A constrained MPC problem where the constraints are modeled by ICRNNs is also tested. The result was able to remain feasible during operation.

4

Representation power of Input Convex Neural Networks

In this chapter it is investigated how well ICNNs can learn convex and non-convex functions. A wide variety of both convex and non-convex functions exists. In particular, the shapes of non-convex functions can differ substantially and it is therefore expected that the accuracy with which ICNNs can learn these functions can vary greatly. To get insight into the individual functions that can and cannot be learned accurately, ICNNs are trained on simple convex and non-convex functions and the results are analysed. The architecture used to construct the ICNNs in these experiments is the architecture for fully input convex neural networks defined by Amos et al. [13]. The ReLU (displayed in Figure 4.1) is chosen as activation function in the network, because it is commonly used and meets the requirements of being convex and non-decreasing. The implementation is done in Python 3 using the Keras library.

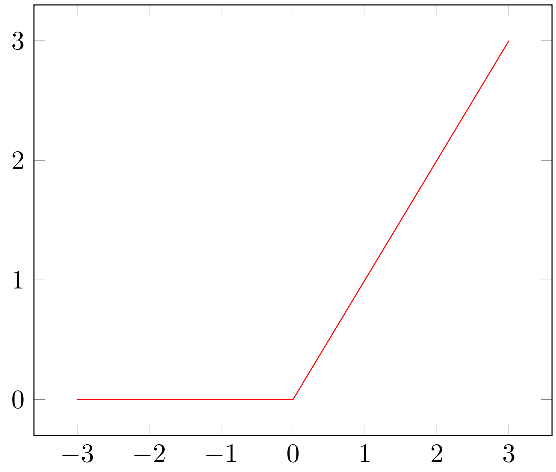


Figure 4.1: Rectified Linear Unit (ReLU) function: $f(x) = max\{0, x\}$

First, in section 4.1 the experiments are done to determine how well ICNNs can learn a simple convex function. Second, in section 4.2 these experiments are done on a simple non-convex function. To then test the performance of ICNNs for different non-convex functions, in section 4.3 randomly generated non-convex functions are used for the experiments. After that, in section 4.4 observations that can be made on the results of the experiments are discussed. Lastly, the implications of the results and observations are discussed in section 4.5.

4.1. A convex function

In order to test the ability of ICNNs to learn convex functions, an experiment is done where an ICNN is trained to learn the simple convex function $f(x) = (x - 0.5)^2 \cdot 4$ with domain $[0, 1]$ as shown in Figure 4.2.

This quadratic function was chosen as it is convex and poses a bigger challenge than a simple linear function.

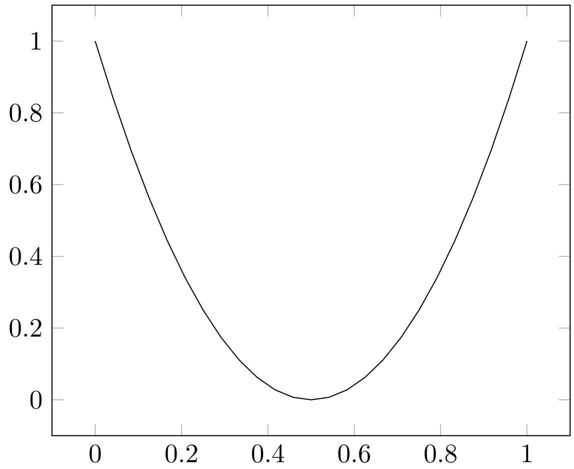


Figure 4.2: A simple convex function.

The fully input convex neural network as defined by Amos et al. [13] has a variable amount of layers. To determine the amount of layers and the layer sizes best used in the ICNN, ICNNs with different configurations are tested. For each configuration ten networks are trained with a training set of 8000 data points and 200 epochs. The mean squared error (MSE) is used as the loss function during training and the Adam optimizer was used. For the ten networks, the MSEs of a testing set of 2000 points are shown in Figure 4.3. Since the figure shows good results for three layers with sizes 20, 5 and 1, these configurations are chosen.

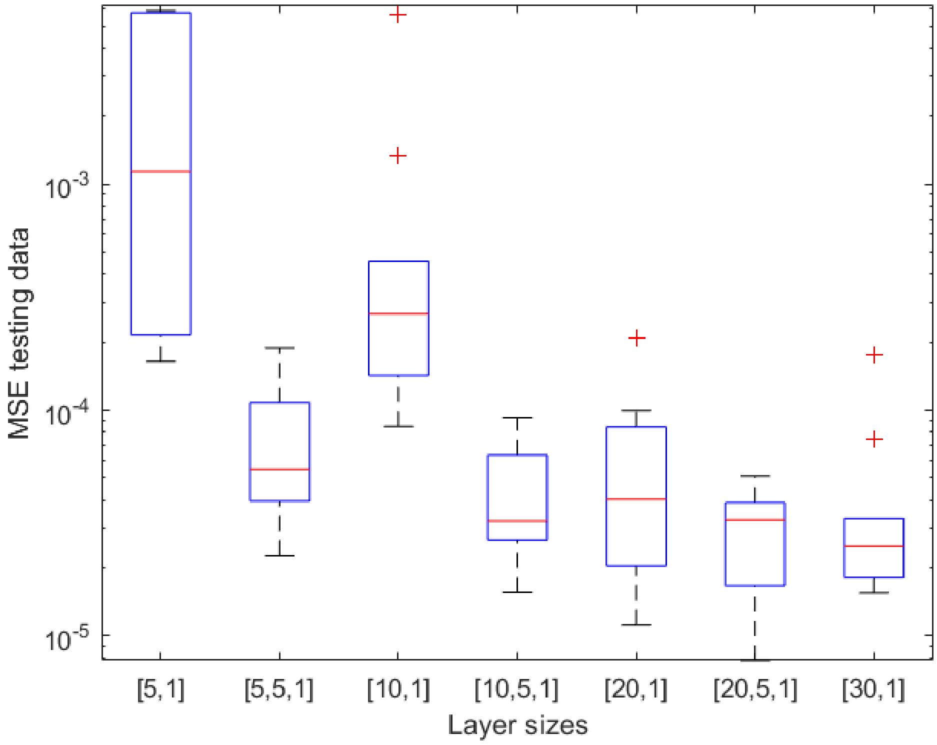
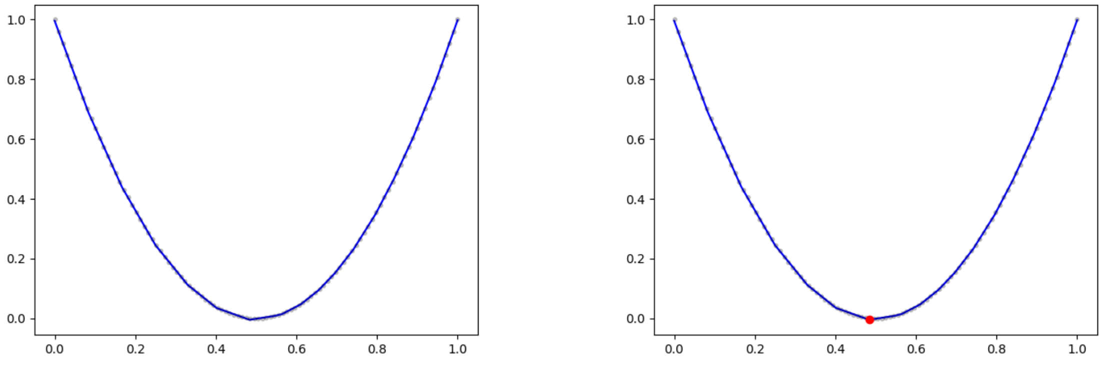


Figure 4.3: Mean squared errors of testing data for different network configurations. The configurations are denotes as lists, where the elements are the sizes of the layers.

Preliminary analysis show that 100 000 data points and 200 epochs can be used to train the network accurately in a reasonable time span. A train-test split of 80:20 is used. An analysis on the batch size has shown that a batch size of 128 performs well in a reasonable amount of time (sizes 64, 256 and 512 were also tried). In Figure 4.4a the function learned by a trained ICNN is shown, by predicting all data points in the data set. Relatively low MSEs can be achieved: an average MSE of the training data of ten trained networks is $4.74 \cdot 10^{-6}$ and the average MSE of the testing data is $4.12 \cdot 10^{-6}$. The training took on average 87.78 seconds.



(a) The predictions by the network are in blue and the points from the function are in gray.

(b) The minimum found with gradient descent, indicated in red.

Figure 4.4: A convex function learned by an input convex neural network.

Next, it is tested how close the optimum found by minimizing the function learned by an ICNN is to the real optimum. This is done using gradient descent on the input of the ICNN. This process is similar to the training of neural networks, except that during training the gradient is calculated for the weights and biases in the network. For the optimization a learning rate of 0.01 is used and 100 iterations are done. Preliminary analysis shows that this learning rate is able to converge to the minimum (0.001 and 0.05 were also tested) and that generally no more improvements are made after 100 iterations.

The resulting minimum that is found for one trained network is displayed in Figure 4.4b. The actual minimum of the function is at $x = 0.5$ and the found minimum is $x = 0.4839$. The average found minimum of ten trained networks is $x = 0.5091$. The average time it took to find the optimum was 0.55 seconds.

4.2. A non-convex function

In this subsection, experiments are done to get an indication of the overall feasibility of learning a non-convex function with an ICNN and the properties of non-convex functions that make them better approximable by an ICNN. It is known from the definition of ICNNs that the output of the network is always a convex function of the input, so it can be expected that a trained ICNN will have inaccuracies.

Firstly, an ICNN is used to learn the non-convex function $f(x) = (x-0.2)^2 \cdot (x-0.6) \cdot (x-1) \cdot 25 + 0.4$ (scaled between 0 and 1). This function is shown in Figure 4.5. This function is chosen as it has clear non-convex properties, but still has a clear global minimum. The same layers and layer sizes are used for the ICNN as in section 4.1, and the same configurations for the training of the network are used. Figure 4.6 shows the function learned by a trained ICNN and the minimum that is found through gradient descent. The average MSEs of the training data and testing data of ten trained networks are both 0.0101, which are more than two orders of magnitude larger than for the convex function. The minimum of the function is at $x = 0.8562$ and the average found minimum of the ten trained networks is at $x = 0.9189$. The average runtimes of the training and optimization (86.98 seconds and 0.56 seconds respectively) are close to the runtimes for the convex function, which is expected as the same configurations are used. The runtime for the rest of the experiments will be omitted because of the same reason.

To test what effects the restrictions on the weights and activation functions in the ICNN have on the learned function, the non-convex function is also learned by neural networks without these restrictions. For a fair comparison, the architecture of these networks and the configurations with which they are run are kept the same. In Figure 4.7a the weights are no longer required to be non-negative, but the activation function is kept the same. In Figure 4.7b the weights are also no longer required to be

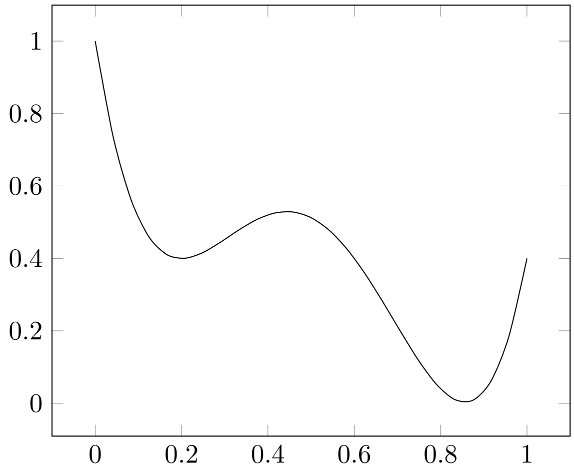
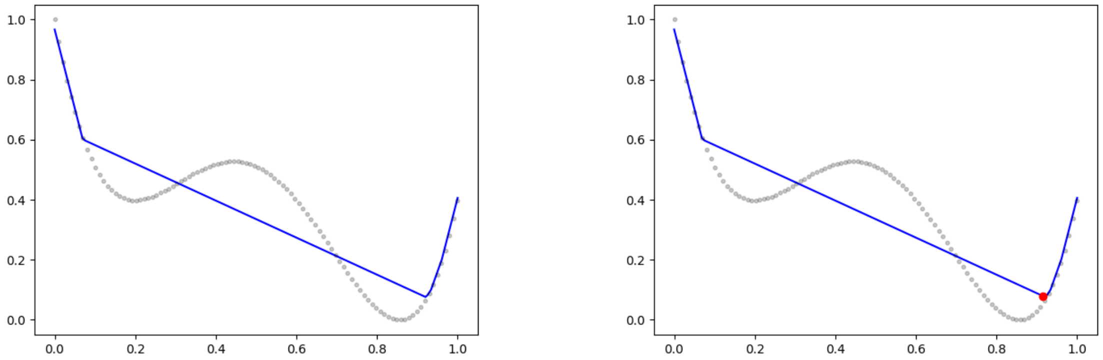


Figure 4.5: A simple non-convex function.



(a) The predictions by the network are in blue and the points from the function are in gray.

(b) The minimum found with gradient descent, indicated in red.

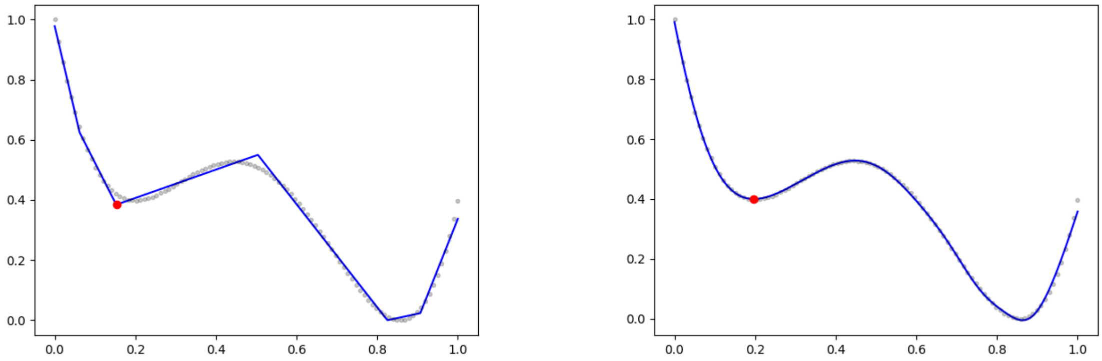
Figure 4.6: The non-convex function learned by an input convex neural network.

non-negative and instead of the ReLu, the sigmoid activation function is used.

Improving the accuracy of the minimum

The results of learning this simple non-convex function demonstrate that both the fit of the function and the optimum of the function are less accurate than for the convex function. However, in the scenario where the learned function is used to find the optimum of the function, then the fit of the whole function is less important than the accuracy of the optimum of the function. It is therefore expected that accuracy of the optimum could be increased by allowing the rest of the function to have a less accurate fit. To test if it is possible to increase the accuracy of the optimum this way, ICNNs are trained on the non-convex function using two different weighted mean squared errors (WMSEs) as loss function instead of the MSE. The weights can be adapted depending on the y coordinate and can therefore make errors in predictions for points with a lower y coordinate more significant. This would mean that errors around local minima are also weighted higher, but errors around the global minimum still have the biggest weight.

The first WMSE that is tested multiplies the squared errors by a linear weight based on the y coordinate. This weight is between 0 and 1 and since the range of the (scaled) function is $[0, 1]$, this weight is taken to be $(y - 1)$. After this weight is applied, the mean is taken. One resulting learned function and minimum found after training an ICNN with this WMSE is displayed in Figure 4.8a. The average found minimum for ten trained ICNNs is at $x = 0.9085$, which is a slight improvement on the results using the MSE.

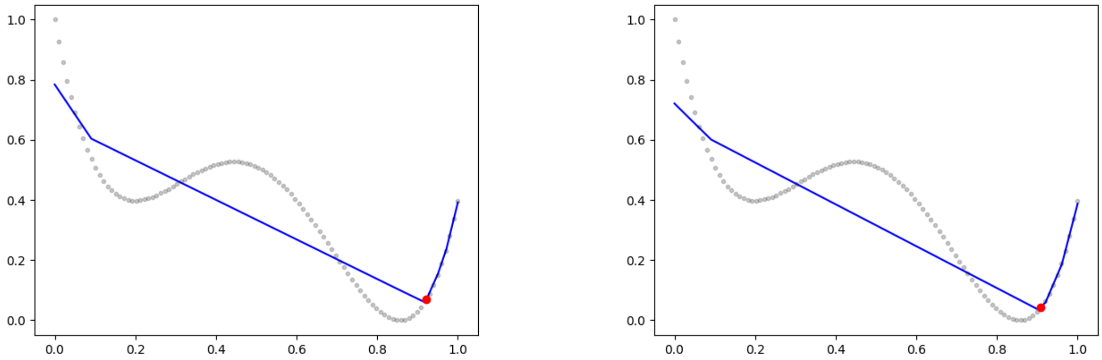


(a) The predictions by the network without weight restrictions with ReLU activation functions.

(b) The predictions by the network without weight restrictions with sigmoid activation functions.

Figure 4.7: The non-convex function learned by neural networks without restrictions.

To try to further increase the accuracy around the optimum, the second WMSE that is tested multiplies the squared errors by a quadratic weight. This weight is taken to be $(y-1)^2$. This results in the same weighted error for $y=0$, but decreases the importance of errors for $y>0$ quicker. One resulting learned function and minimum found after training an ICNN with this WMSE is displayed in Figure 4.8b. The average found minimum for ten trained ICNNs is at $x=0.9020$. This minimum is an improvement as it is closer to the actual minimum at $x=0.8562$.



(a) Results from training with the linear WMSE.

(b) Results from training with the quadratic WMSE.

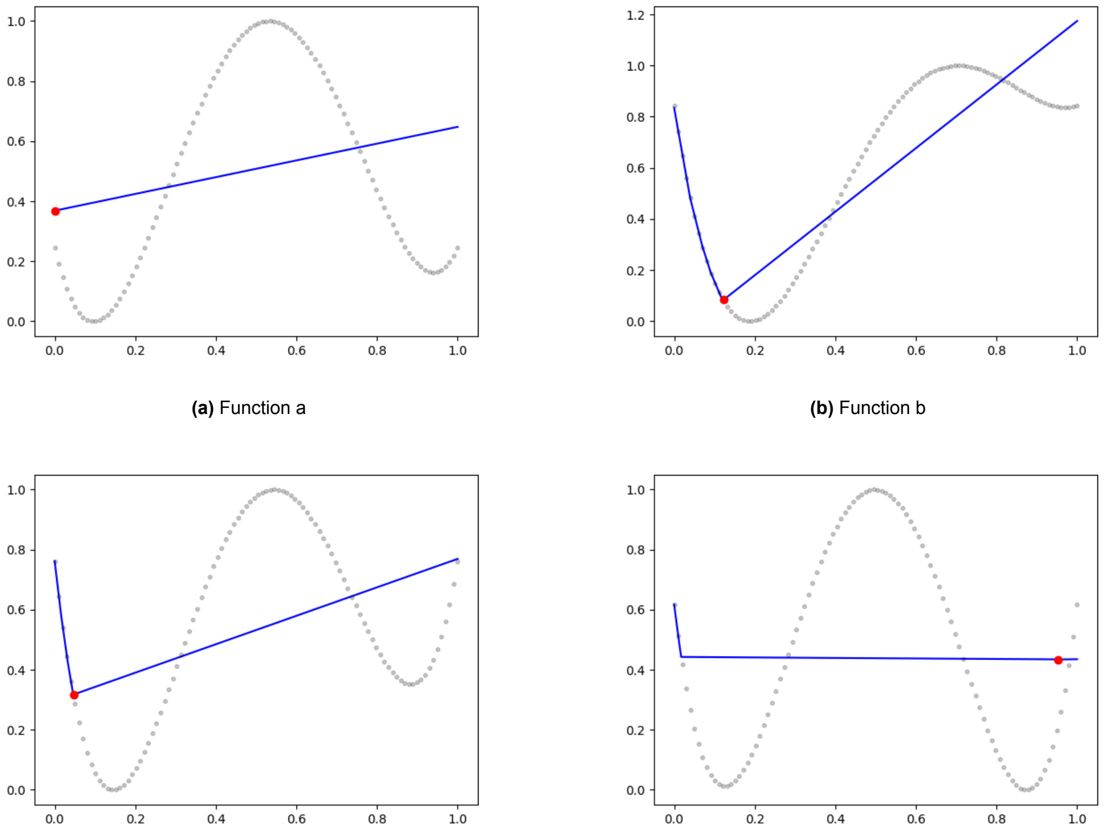
Figure 4.8: A non-convex function learned by an input convex neural network with the WMSE.

4.3. Randomly generated non-convex functions

In this section ICNNs with the same configurations as the last experiments are used to learn randomly generated non-convex functions, in order to test the accuracy of the trained ICNNs on non-convex functions with varying shapes. The non-convex functions used for this are randomly generated quartic functions in the format $f(x)=(x-0)\cdot(x-a)\cdot(x-b)\cdot(x-1)\cdot c$ scaled to $[0,1]$, with a,b and c sampled from a uniform distribution. This produces functions that are non-convex within the domain $[0,1]$ and have one or two minima. This format was chosen as it typically produces functions with significant non-convexities in the domain $[0,1]$. The shapes of these functions do have some restrictions: $f(0)=f(1)$ and there always exists one or two inputs i for which $f(0)=f(i)=f(1)$.

In Figure 4.9, four examples of randomly generated quartic functions are displayed along with the function learned by the ICNN and the found minimum.

These four non-convex functions are also learned by the ICNNs using the linear and squared WM-



(a) Function a

(b) Function b

(c) Function c

(d) Function d

Figure 4.9: Non-convex functions learned by an input convex neural network.

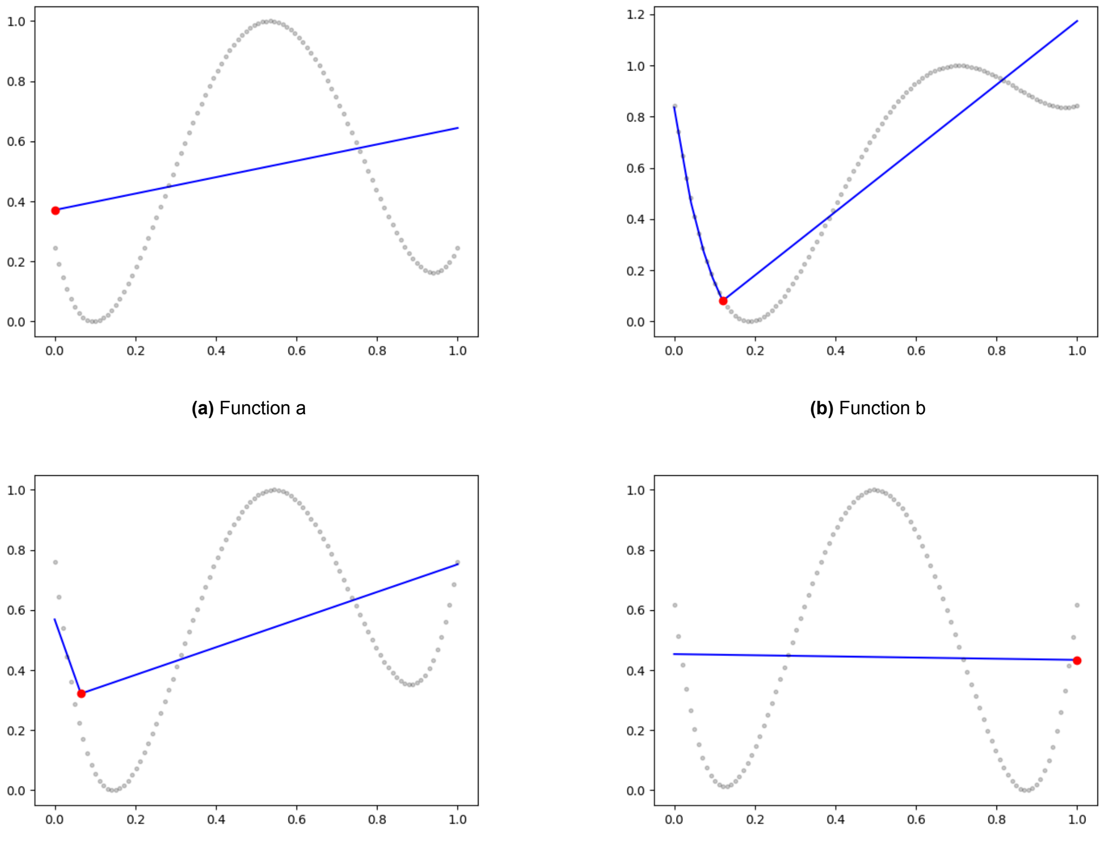
SEs described in section 4.2. The results are displayed in Figure 4.10 and Figure 4.11.

4.4. Observations

In Figure 4.4a the results can be seen of training an ICNN on a simple convex function. When looking closer, it can be seen that the learned function is piecewise linear. This can be explained by the fact that ICNNs can represent a maximum of affine functions [14]. It can be seen in Figure 4.4b that the found minimum is approximately located at the intersection point of two line segments. It can be reasoned that the minimum of the function learned by an ICNN will always be at an intersection point of two linear line segments, as other points can be improved upon by following the gradient of the segment that it is on. Since the amount of affine functions that an ICNN can use to construct the convex piecewise linear function depends on the amount of activation functions in the network [14], it is expected that increasing the network size can increase the accuracy of the minimum of the learned function.

From the results of training an ICNN on a simple non-convex function (Figure 4.6a and Figure 4.6b) it can be seen that the ICNN is unable to learn a convex function that fits well. The left-most and right-most parts of the function (close to 0 and 1) can be fit relatively well because the accuracy in those parts is not restricted by the convexity of the function. The rest of the function, however, cannot be approximated by a convex function. Therefore the ICNN converges to a function that minimizes errors, but is not an accurate representation of the original. It can be seen that the optimum of the function is also inaccurate because of this. In Figure 4.7a it is demonstrated that if the weights in the network are not restricted to be non-negative, then the network is able to learn a piece-wise linear non-convex function. When instead of the ReLU the sigmoid activation function is used, the network is able to learn a more smooth version. However the result of the minimization of both networks are local minima.

The results of learning the same non-convex function with the linear WMSE as loss function is



(a) Function a

(b) Function b

(c) Function c

(d) Function d

Figure 4.10: Non-convex functions learned by an input convex neural network with the linear WMSE.

displayed in Figure 4.8a. It can be seen that the accuracy of the minimum of the learned function has improved slightly, but at a cost of the accuracy of the rest of the function (as can be seen at x close to 0).

In Figure 4.8b it can be seen that the usage of the quadratic WMSE further improves the accuracy of the minimum of the learned function. The shape of the function in the region around the optimum is still not correct though, which is because of the minimization of errors in the rest of the function.

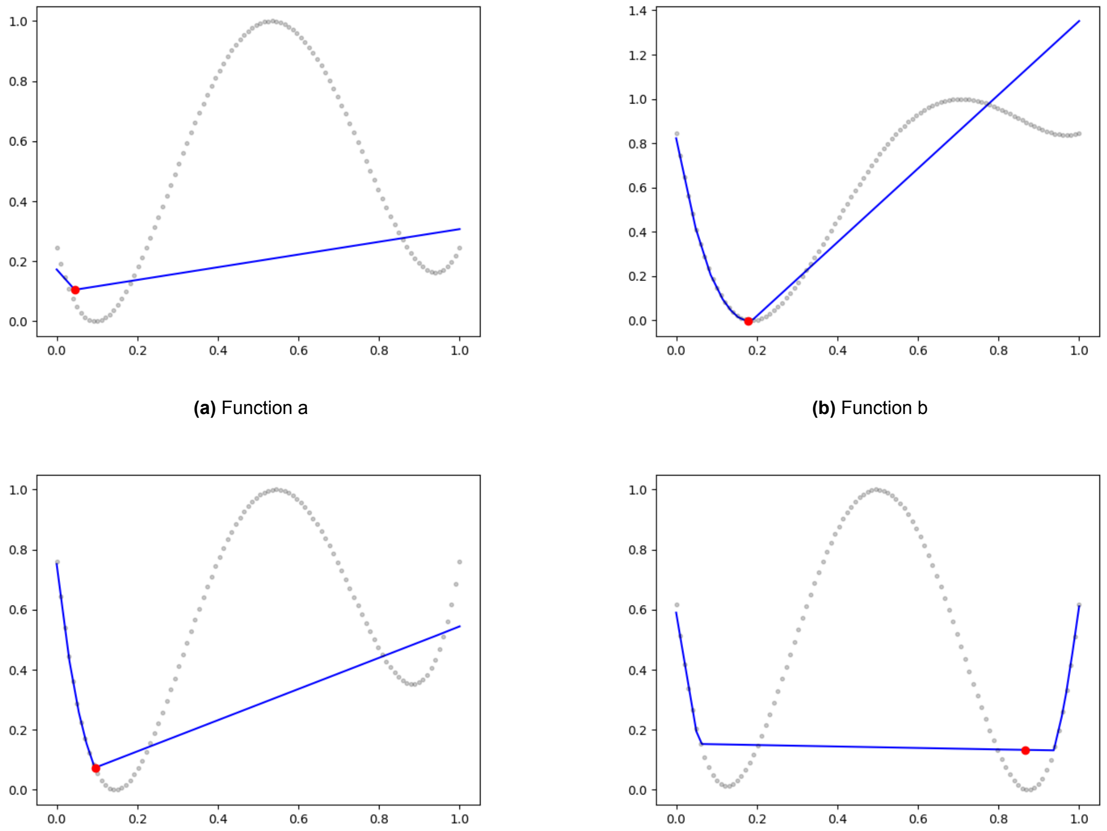
In the results of the four generated non-convex functions in Figure 4.9 it can be seen that the shape of the non-convex function has a large effect on the possible accuracy of the fit of a convex function. In the functions with two more significant minima (Figure 4.9a, Figure 4.9c and Figure 4.9d) the convex fit is less accurate than in the function with one significant minimum (Figure 4.9b).

For these same four functions it can be seen in Figure 4.10 that the linear WMSE does not impact the shape and the minimum of the learned function much. The quadratic WMSE (in Figure 4.11) does visibly impact the shape and minimum. In the case of Figure 4.11b the minimum found by the ICNN is able to get close to the actual minimum.

4.5. Discussion

From the results from section 4.1 it can be concluded that ICNNs can be used to learn the simple convex function $f(x) = (x - 0.5)^2 * 4$ accurately. Since other convex functions are not tested in this work, no general conclusions can be drawn about which convex functions can and cannot be approximated well.

The results from section 4.2 and section 4.3 indicate that how accurately an ICNN can learn a non-convex function depends heavily on the shape of the function. In some cases some parts of the non-convex function can be learned accurately by the ICNN. Other parts of the function that have non-convex properties, however, are impossible to learn accurately with an ICNN. When the MSE is used



(a) Function a

(b) Function b

(c) Function c

(d) Function d

Figure 4.11: Non-convex functions learned by an input convex neural network with the quadratic WMSE.

as the loss function, the priority during the training of the network is to reduce the biggest errors in the training data. For non-convex parts of the function this results in a learned function that ”balances” the errors and does not represent the function accurately in any region.

One property of ICNNs is that the output of the network is a convex function of the input. This convexity guarantees that all local minima of the function are also global minima, which should make optimization of the input easier. To test how well the input of an ICNN can be optimized, gradient descent is used to optimize the input of the network. With a tuned learning rate this method performs well; the optimum can be found within 100 iterations.

Even though the optimum of the function learned by an ICNN can be found efficiently, this optimum is not necessarily accurate if the original function is non-convex. To try to increase the accuracy of the ICNN specifically around the optimum of the function, two different error metrics are tested. The quadratic WMSE improves the accuracy of the optimum of the learned function the most. The shape of the learned function around the optimum, however, is still often inaccurate, as it is affected by the error minimization in other parts of the function.

5 Feasible space of supply temperatures to the District Heating Network

In this chapter the experiments on the feasible solution space of supply temperatures to a DHS are described. These experiments have the goal of gaining insight into what the non-convexities in this space look like and what causes them. In order to find the most simple scenario in which non-convexities in the feasible space occur, a small DHS with one producer and one consumer is used. Furthermore, only the feasible solution space of two time steps in the predictive horizon are considered. This is again done to find simple scenarios in which non-convexities occur and because two-dimensional results can be interpreted better.

In section 5.1 experiments are described on the feasible space for a DHS with a storage tank. In this network the consumer is connected to the producer by a heat storage tank instead of a pipe. This DHS is simpler, as there are no time delays in the network and the heat storage is not influenced by the mass flow in the network. Experiments on the feasible space of this DHS are done first, as the storage tank in this network could have a similar effect on the feasible space as the network storage capacity in a DHS with pipes. In section 5.2 experiments are described on the feasible space for a DHS where the producer and consumer are connected by a supply and return pipe without a storage tank.

5.1. Network with heat storage tank

Firstly, the setup for the experiments are described in subsection 5.1.1. Then in subsection 5.1.2 the results from the experiments are presented and in subsection 5.1.3 observations about these results are discussed. Finally, the implications of these results are discussed in subsection 5.1.4.

5.1.1. Experimental setup

For these experiments a small network is used with one producer, one consumer and a heat storage tank connecting the two. The model for a heat storage tank by Abdollahi et al. [10] is employed here. In this model, the total amount of heat in the storage tank is kept track of, which can increase when charged or decrease when discharged. Since this model works with the heat production from the CHP as input, the experiments on the feasible solution space of this DHS are done with the heat production instead of supply temperatures.

The method used to determine the feasibility of the heat production of two time steps uses the linear programming model by Abdollahi et al. [10] that minimizes the operation costs of a CHP. This model minimizes the costs for heat production and electricity production given the CHP operating region, the heat demands and the maximum storage capacity of the heat storage. The feasibility of the heat production of two time steps is determined by setting the heat production and optimizing the linear model. If the model returns an optimal solution for the electricity production and heat storage usage, then it is known that the heat productions for the two time steps are able to fulfil the heat demand. If the model is infeasible, then it is not possible to fulfil the heat demand with the given heat productions.

The CHP operating region that is used is displayed in Figure 5.1. Furthermore a storage efficiency and discharge efficiency of 0.95 were used.

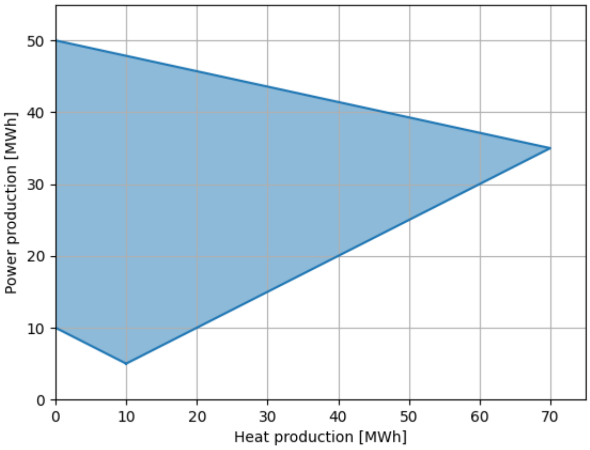


Figure 5.1: The CHP operating region [7].

5.1.2. Feasible space experiments

Firstly, the feasible space is determined for a scenario with a storage tank that has a relatively large storage capacity compared to the heat demands. Because of the large capacity of the tank, it is expected that any excess heat produced during time step 0 (t_0) can be stored and can be used to fulfill the heat demand during time step 1 (t_1).

The maximum heat storage capacity is taken to be 200 MWh. The heat demand of t_0 and t_1 are taken to be 30 MWh and 40 MWh respectively. The feasibility is tested for heat productions h_0 and h_1 (at t_0 and t_1) with values between 0 MWh and 70 MWh, as these values are within the operating region of the CHP (Figure 5.1). The resulting feasible solution space for h_0 and h_1 is displayed in Figure 5.2.

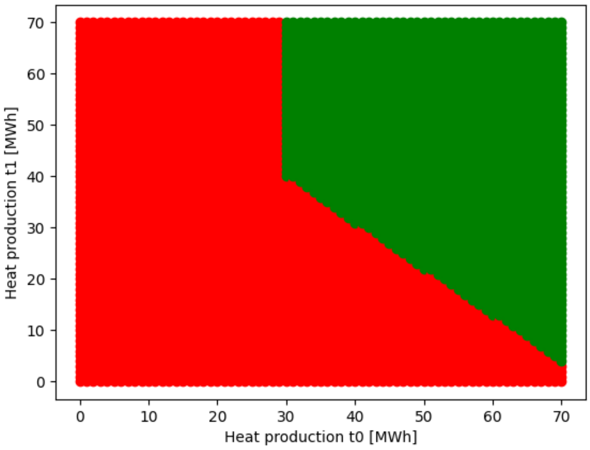


Figure 5.2: The feasible space of heat production of the CHP at time step t_0 and t_1 with a storage capacity of 200 MWh. A green point in the plot means that those heat productions can fulfil the heat demand. A red point means that the heat demand cannot be fulfilled.

The feasible space of a DHS with a storage tank with a smaller capacity is also tested. This smaller capacity limits the amount of excess heat that can be stored, and it is therefore expected that solutions can now be infeasible because too much heat is produced. For this example a maximum capacity of 20 MWh is taken. The same heat demands are used. The resulting feasible space is shown in Figure 5.3.

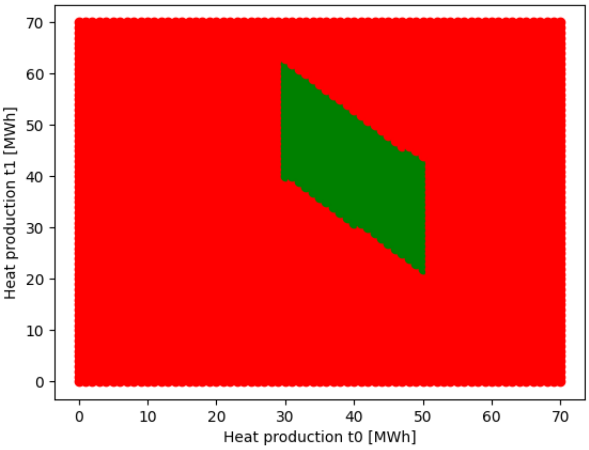


Figure 5.3: The feasible space of heat production of the CHP at time step t_0 and t_1 with a storage capacity of 20 MWh. A green point in the plot means that those heat productions can fulfil the heat demand. A red point means that the heat demand cannot be fulfilled.

5.1.3. Observations

In Figure 5.2 the feasible space is displayed of a DHS with a storage tank with a capacity of 200 MWh and heat demands of 30 MWh and 40 MWh for t_0 and t_1 respectively. In this figure it can be seen that solutions with $h_0 \geq 30$ and $h_1 \geq 40$ are feasible. This is because the enough heat is produced to fulfil the heat demand for both time steps and any excess heat can be stored in the heat storage. It can also be seen that for $h_0 > 30$, increasing h_0 allows for lower h_1 to be feasible. This is possible because the excess heat at t_0 is stored and can be discharged at t_1 to help fulfil the demand.

In Figure 5.3 the feasible space of the same DHS and heat demands are displayed, except the capacity of the heat storage tank is 20 MWh. It can be seen that the solution $h_0 = 30$, $h_1 = 40$ is feasible, but now there is a limit to the additional heat that can be produced in both time steps. For the infeasible solutions with $h_0 > 30$ and $h_1 > 40$ the excess heat cannot be stored because it is more than the maximum capacity of the tank. Furthermore, increasing h_0 allows for excess heat to be stored and to be used in t_1, which makes lower values of h_1 feasible.

5.1.4. Discussion

In these experiments the feasible space of the heat production of a CHP in a DHS with a heat storage tank is studied. For the possible heat productions at t_0 and t_1 within the CHP operating region, it is determined if the heat demand can be satisfied and constraints on the storage tank are not violated. From the figures it can be concluded that the usage of a storage tank allows for more heat to be produced than is needed for the heat demand, since the excess heat can be stored. Storing excess heat produced at t_0 also allows for lower heat production at t_1 to be feasible, since the stored heat can fulfil part of the demand at t_1. It can be said that using heat storage allows for more flexibility in the operation of the DHS, because there are more feasible solutions when heat storage is used. Since a DHS with a network of pipes can also store heat, there could be similarities in the feasible space of the supply temperatures.

5.2. Network without heat storage tank

In this section the experiments are described that are done in order to study the feasible space of supply temperatures for a small DHS without a heat storage tank. This DHS has one producer and one consumer, which are connected by supply and return pipes. As opposed to the model in section 5.1, this model includes time delays in the network.

To begin, the experimental setup is described in subsection 5.2.1. After that, in subsection 5.2.2 and in subsection 5.2.3 the experiments on the feasible space of the supply temperatures are discussed. Then in subsection 5.2.4 observations about these results are discussed and in subsection 5.2.5 the implications of these results are discussed.

5.2.1. Experimental setup

To determine if a set of supply temperatures is a feasible solution for the DHS, a simulator developed by Flex Technologies is used. The report of this simulator is in progress [17]. When this simulator is configured with the physical properties of a DHS, it can determine the statuses of the components of the DHS in the following time steps in the predictive horizon, given the heat demands of consumers and previous and future temperature outputs of the CHP. This simulation of the DHS is based on a detailed model that includes the node method [8]. For every time step in the simulation, it is checked if any of the constraints are violated. The feasibility of supply temperatures to the DHN are determined using these constraints; a solution is marked feasible when no constraints are violated and infeasible when one or more constraints are violated. The following constraints are implemented in the simulator:

- The flow speed constraints. The pipes in the network have a maximum flow speed (or mass flow) for which the pipe functions properly.
- The temperature ramping constraints. Each time step, the amount of degrees that the outlet temperature of the CHP can increase or decrease is limited.
- The operating region constraints. The heat and power production of the CHP should be within its operating region. With a large mass flow, higher supply temperature settings can result in a heat production that is outside of the operating region.
- The supply temperature constraints. When the operation of the CHP is defined with its heat production, this constraint makes sure that the corresponding CHP outlet temperature is within the allowed region.
- The heat delivery constraints. The heat demand of the consumers should be satisfied every time step. If the water temperature is not high enough when it arrives at the heat exchanger, then the amount of heat that can be extracted is not enough to fulfil the demand.

For the experiments the same CHP operating region is used as for the DHS with a heat storage tank (Figure 5.1). The outlet temperature of the CHP can be between 70 and 120 °C. The supply and return network of the DHS both consist of one pipe with a length of 12 km and a diameter of 0.5958 m. These settings are based on the work of Stepanovic et al. [7], where the pipe of 12 km was able to save more money compared to the 4 km pipe because of the increased storage capability. Prior to the simulation, the water in the supply and return pipeline are initially set to 90 °C and 50 °C respectively. The rest of the configurations needed for the simulator were based on the implementation of the work of Stepanovic et al. [18].

The feasible space of the supply temperatures for a DHN depends heavily on the heat demands of the consumers. Therefore how convex or non-convex it is also depends on the heat demands. To look at the feasible space of heat demands in a realistic scenario, the simulator is run with a data set of hourly heat demands of the Netherlands estimated by Ruhnau et al. [19]. The data has been rescaled so the maximum heat demand corresponds with the maximum heat production of the CHP.

5.2.2. Feasible space experiments for time step 0 and 1

For these experiments the feasible space of supply temperatures of time step t_0 and t_1 is considered. Different settings of supply temperatures for these time steps are taken as input for the simulator. The feasibility of these temperatures is partially determined in the time steps that the plug arrives at the consumers, since the temperature of the plug must be high enough to satisfy the heat demand. However, violations in time steps after the consumption can also be indirectly caused by the temperature settings at t_0 and t_1, since the mass flow during the consumption of $plug_0$ and $plug_1$ can limit the supply temperature at the producer. So to determine the full effect that temperatures for $plug_0$ and $plug_1$ have on the feasibility of the problem, the simulator is run for a full predictive horizon of 24 time steps. To try to ensure that the feasibility of the solution is determined by the supply temperatures of t_0 and t_1 and not the rest of the time steps, the supply temperatures in the rest of the predictive horizon are filled in a way that is least likely to cause violations:

- The temperatures of these time steps are set relatively high, in order to ensure that the production in these time steps do not violate the heat delivery constraints if possible. For the heat demands from the data set it was determined that a temperature of 90 °C is high enough.

- In the scenario that the temperature at t_1 is below 90 °C, the temperatures in the following time steps are increased by the maximum that is allowed by the temperature ramping constraints.
- During preliminary analysis it was observed that a low supply temperature can result in a large mass flow when consumed, which can cause the supply temperature of 90 °C to result in a heat production that is outside of the CHP operating region (see Equation 2.1). Therefore after filling the rest of the time steps with supply temperatures up to 90 °C, the simulator is run to check if the operating region constraints are violated. If this is the case, the supply temperature of this time step is lowered until the operating region constraint is no longer violated. After this, the temperatures of surrounding time steps are lowered if the temperature ramping constraints are violated.

The simulator is run with the first 24 heat demands from the data set (displayed in Figure 5.4) for all combinations of temperatures between 70 and 120 °C at t_0 and t_1. The feasibilities of these temperatures are displayed in Figure 5.5. To identify which constraints are violated for the infeasible solutions, the plots in Figure 5.6 depict the temperatures for which the temperature ramping constraints and the heat delivery constraints are violated. The rest of the constraints are not depicted as they were never violated.

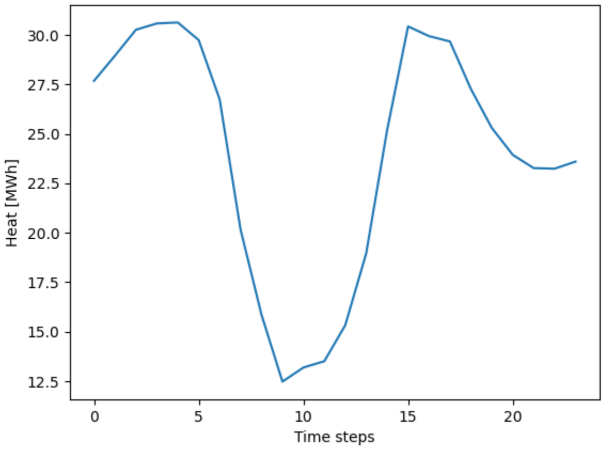


Figure 5.4: The first 24 heat demands from the data set.

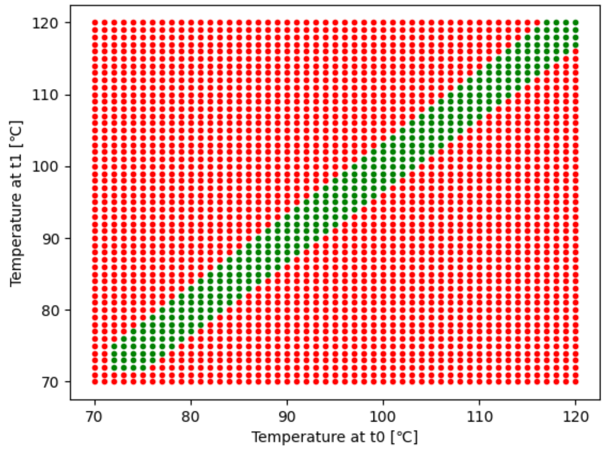
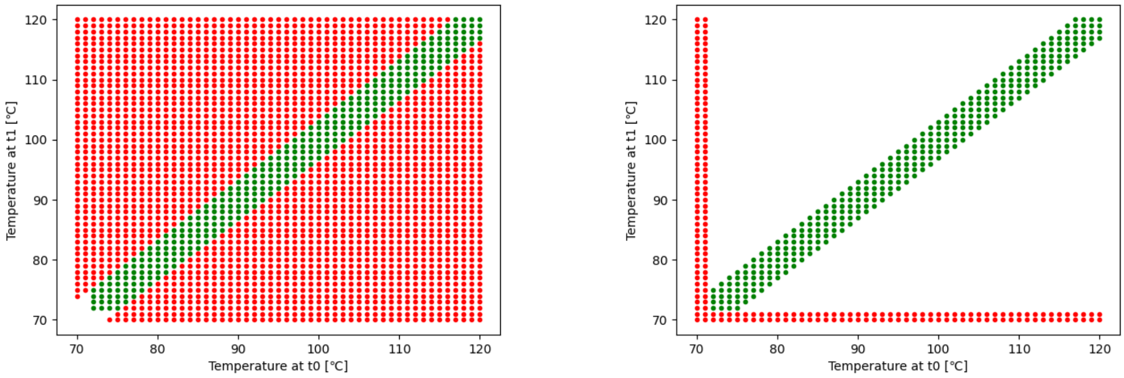


Figure 5.5: The feasible space of supply temperatures at time steps t_0 and t_1. A green point in the plot means that the temperatures can be used for a feasible solution. A red point means that the temperatures cause a violation.

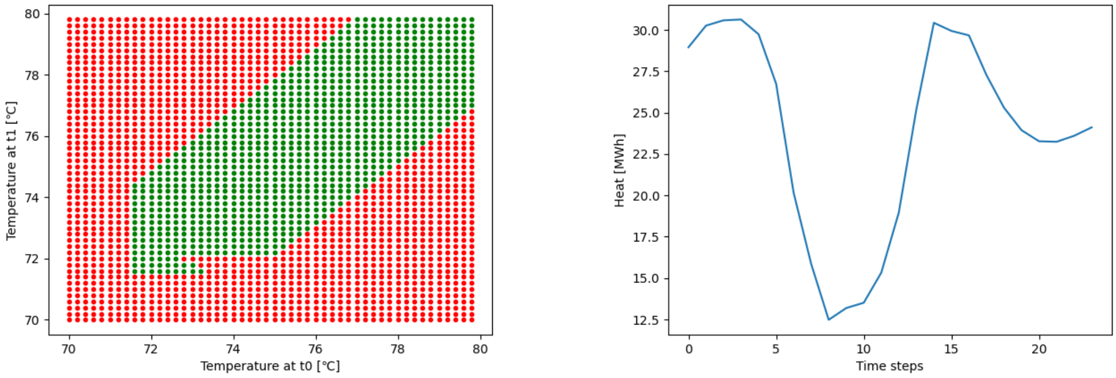


(a) In green are the feasible solutions. In red are solutions that violate the temperature ramping constraints.

(b) In green are the feasible solutions. In red are solutions that violate the heat delivery constraints.

Figure 5.6: The feasible space with infeasible points caused by individual constraint violations.

In order to study the feasible space of scenarios with different heat demands, the simulator is run on 24 time step periods from the data set starting at t_1, starting at t_2, et cetera. For some of the resulting feasible spaces, a non-convex shape can be observed when limiting the supply temperatures between 70 and 80 °C and verifying the feasibility with increments of 0.2 °C. One example of this shape is displayed in Figure 5.7a. This ”hook” in the feasible space is observed in 8 out of 50 total runs. The individual constraint violations (Figure 5.8) reveal that the infeasible solutions around the hook violate the heat delivery constraints.



(a) The feasible space of supply temperatures at time step t_0 and t_1.

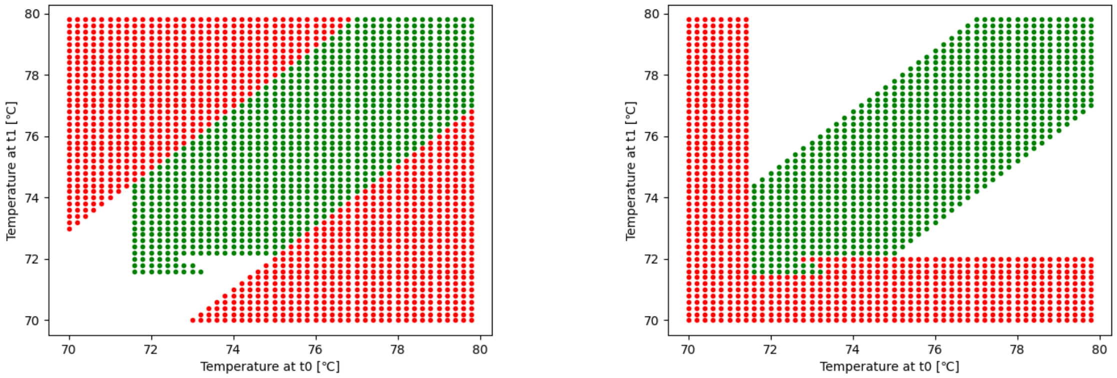
(b) The heat demands.

Figure 5.7: The ”hook” non-convexity

5.2.3. Feasible space experiments for time step 0 and 2

As can be seen in Figure 5.6a, the feasible space of the supply temperatures of t_0 and t_1 is mostly limited by the temperature ramping constraints, and therefore the effect that the temperature of t_0 can have on the feasible temperatures of t_1 is also limited. To study what effect a temperature at one time step can have on the feasible temperature of a later time step, the feasible space of supply temperatures at time step t_0 and t_2 is considered in this subsection. The temperature for t_1 is taken to be the highest temperature that is allowed by the temperature ramping constraints in order to ensure that $plug_1$ does not violate the heat delivery constraints.

Note that there is a possibility that this assignment of t_1 causes the solution to be marked infeasible when it is not. This can happen if t_2 violates the heat delivery constraint when the highest possible t_1 is used, but a lower temperature for t_1 would cause $plug_2$ to be consumed during an earlier time step with a lower demand. In order to be sure that the assignment of temperatures at t_0 and t_2 are infeasible,



(a) In green are the feasible solutions. In red are solutions that violate the temperature ramping constraints.

(b) In green are the feasible solutions. In red are solutions that violate the heat delivery constraints.

Figure 5.8: Feasible space with infeasible points caused by individual constraint violations.

the solution should be infeasible for all temperature settings at t_1. This, however, would increase the computation time significantly. Therefore the temperature at t_1 is taken to be the highest possible, since $plug_1$ is then able to satisfy higher heat demands.

The feasible space of supply temperatures at t_0 and t_2 for the first 24 heat demands from the data set is displayed in Figure 5.9. The individual constraint violations are shown in Figure 5.10.

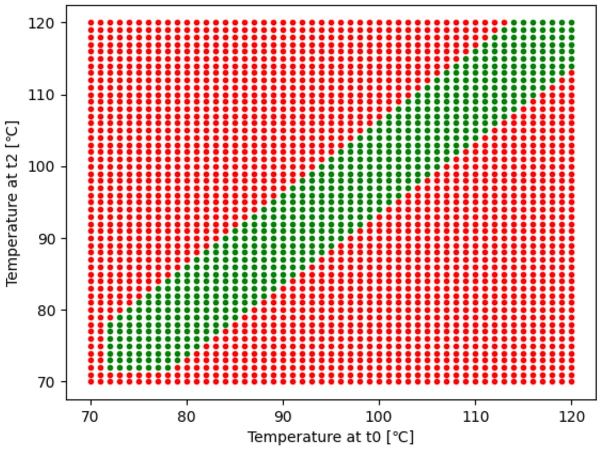
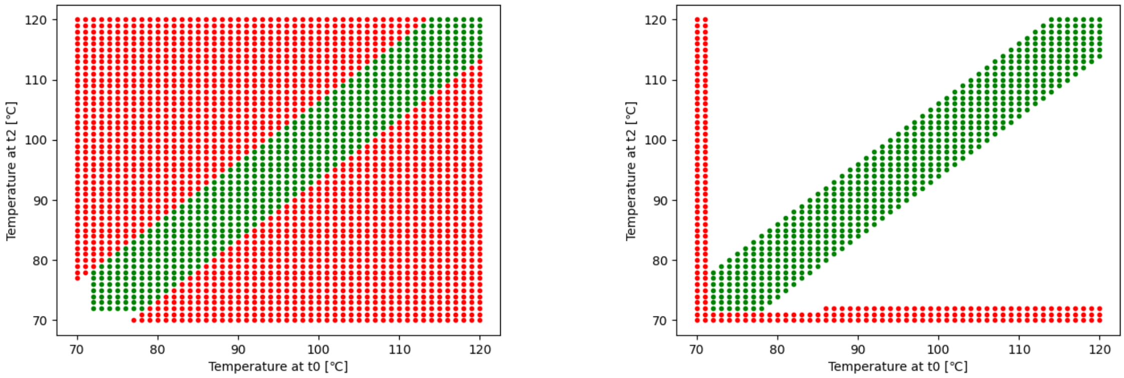


Figure 5.9: The feasible space of supply temperatures at time step t_0 and t_2.



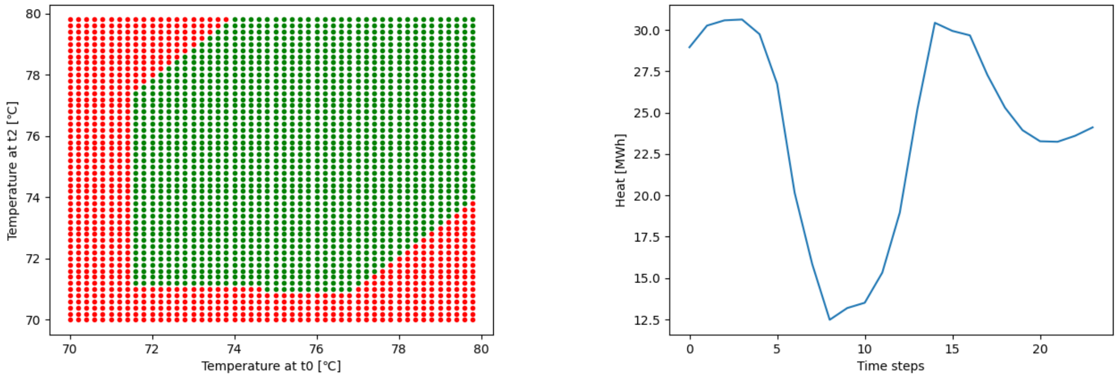
(a) In green are the feasible solutions. In red are solutions that violate the temperature ramping constraints.

(b) In green are the feasible solutions. In red are solutions that violate the heat delivery constraints.

Figure 5.10: Feasible space with infeasible points caused by individual constraint violations.

Similarly to the feasible spaces of supply temperatures at t_0 and t_1 for 24 time step periods from the data set, the feasible spaces of t_0 and t_2 regularly display the "hook" non-convexity. 9 out of 50 runs display this non-convexity.

In addition to the "hook" non-convexity, another non-convexity in the feasible space is found. This non-convexity is displayed in Figure 5.11 and was only found in one out of 50 runs.



(a) The feasible space of supply temperatures at time step t_0 and t_2.

(b) The heat demands.

Figure 5.11: The "extra strip" non-convexity

5.2.4. Observations

In subsection 5.2.2 and subsection 5.2.3 the experiments that are done on the feasible space of the supply temperatures for two time steps of a simple DHS are described. During these experiments, it was observed that all feasible spaces for the 50 tested heat demands have a similar shape to Figure 5.5 and Figure 5.9. From Figure 5.6 and Figure 5.10 it can be seen that this general shape is caused by violations of the temperature ramping constraints and the heat delivery constraints. The configurations with which the simulator is run include a maximum temperature increase or decrease of 3 °C for each time step, and therefore the temperature difference between two time steps is restricted. Additionally, it was observed that the heat delivery constraints are violated when the temperature is below a threshold. The minimum feasible temperatures depend on the heat demands during the time step(s) that the plugs are consumed.

Some of the feasible spaces that were found include an area that is non-convex. Two kinds of non-convexities were found: one shaped like a hook (Figure 5.7a) and one that includes an extra strip of feasible solutions (Figure 5.11a). The origin of both of these non-convexities will now be discussed.

The ”hook” non-convexity

From the status of the network during the predictive horizon provided by the simulator, it was found that the non-convex ”hook” shape of Figure 5.7a has to do with the time step during which $plug_1$ is consumed in combination with the condition under which the simulator flags a violation of the heat delivery constraint.

During simulation, for each time step the heat exchangers extract the right amount of heat from the primary network by adapting the mass flow based on the temperature of the supplied water. Generally, the supply temperature is determined to be high enough to fulfil the heat demands if the required mass flow is below the maximum allowed. There is therefore also a minimum supply temperature which requires the maximum mass flow in order to fulfil the heat demand. In the simulator, a violation is flagged if the inlet temperature to the heat exchanger is below this minimum supply temperature. However, during one time step the supply temperature can change when multiple plugs are consumed. In the simulator, the heat exchangers modify the mass flow for each of the plugs consumed. So in order to ensure that heat is extracted from the network at a constant rate during the time step, the temperature of each consumed plug is verified to be above the minimum supply temperature.

The ”hook” non-convexity in the feasible space can then occur in the scenario where the heat demand increases when $plug_1$ arrives at the consumer. The very point of the hook is the position where the heat demands are satisfied, but increasing either the temperature at t_0 or the temperature at t_1 results in $plug_1$ being (partially) consumed later. When the heat demand at this later time step is higher, then the minimum supply temperature during this time step is higher and a heat delivery violation occurs.

The hooks in feasible spaces of supply temperatures at t_0 and t_2 can be explained similarly. The heat delivery constraint violations in Figure 5.10b also show that this ”hook” non-convexity can also occur for temperatures that are infeasible because of violations of the temperature ramping constraints.

The ”extra strip” non-convexity

The cause of the non-convexity in Figure 5.11a is also determined using the simulator. Similarly to the ”hook” non-convexity, this non-convex shape has to do with to the time step during which the second considered plug (which is $plug_2$, since time steps t_0 and t_2 are considered) is consumed. However, this non-convex shape requires that the heat demands decreases when $plug_2$ arrives at the consumer. The infeasible solution to the left of the ”extra strip” of feasible solutions has a heat delivery violation during time step 5. This violation occurs because $plug_2$ is consumed during time step 5 and has a temperature that is below the minimum supply temperature. However, when the temperature at t_0 is increased, the mass flow during the consumption of $plug_0$ is lowered, which causes $plug_2$ to arrive at the consumer later. When the temperature at t_0 is increased enough, the consumption of $plug_2$ is pushed to time step 6. Time step 6 has a lower heat demand (see Figure 5.11b) and therefore the minimum supply temperature is lower and a lower temperature at t_2 is feasible.

A visualisation of this effect is provided in Figure 5.12. Note that this example shows a different scenario with different temperatures than the circumstances for Figure 5.11a. In scenario 1 $plug_0$ and $plug_1$ are completely consumed during time steps t_{10} and t_{11} respectively. The temperatures of the plugs are the minimum supply temperatures of these time steps. In scenario 2 the temperature of $plug_0$ is increased, which causes a lower mass flow during its consumption. Since the mass of $plug_0$ is the same as scenario 1, the decreased mass flow results in the consumption of the plug being stretched. This delays the consumption of $plug_1$, which is now partially consumed during time step t_{12}. During t_{12} the heat demand is lower and therefore the mass flow is decreased. In scenario 3A the temperature of $plug_0$ is increased enough so that $plug_1$ is no longer partially consumed during t_{11}. The temperature of $plug_1$ is therefore no longer restricted by the minimum supply temperature of t_{11}. Scenario 3B shows that the temperature of $plug_1$ can be decreased to the minimum supply temperature of t_{12}.

5.2.5. Discussion

In the previous subsections, the feasible space of supply temperatures to a DHN with time delays is studied. The feasible space of supply temperatures for two time steps is found to be constrained by temperature ramping constraints of the CHP and heat delivery constraints of the consumer. Non-convexities are found in the feasible space when the simulator is run for some of the heat demands from the data set. These non-convexities were found to be a result of the effect where increasing the temperature of one time step can push back the consumption of a later plug to a later time step. When the demand at this later time step is higher or lower, then the lowest feasible temperature of this plug

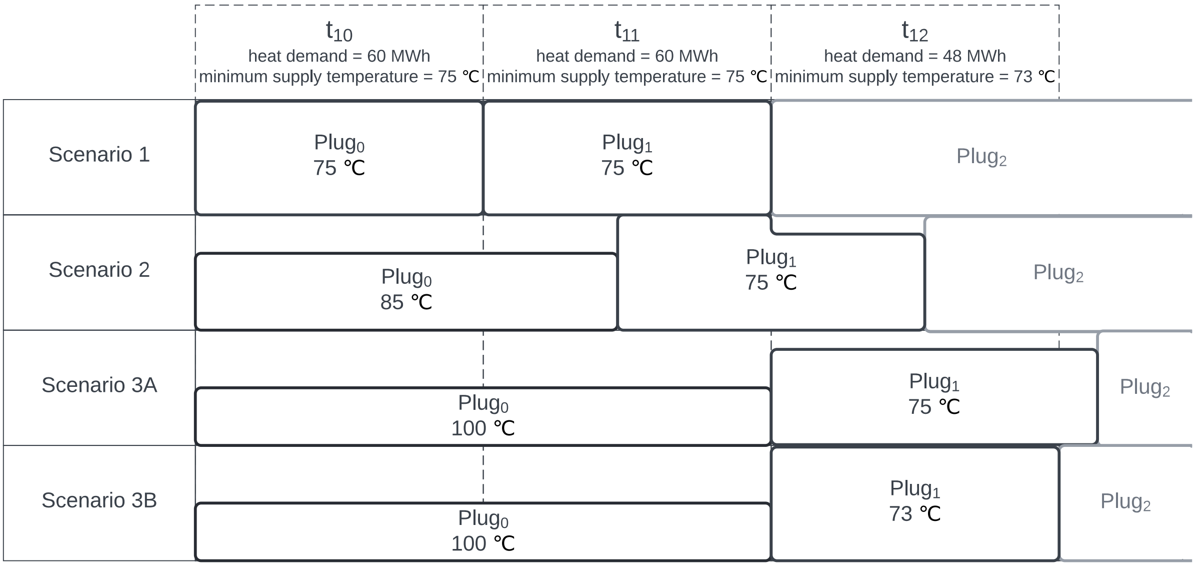


Figure 5.12: Four scenarios of the consumption of $plug_0$ and $plug_1$ with different temperature settings, displaying the effect on the time delay in the network. In the column headers the current time step, the heat demand and the minimum supply temperature can be seen. For all scenarios each column displays which part of which plug is consumed during that time step. Note that this example is not based on a real scenario and only serves as an illustration of the effect. Heat loss is neglected.

will also be higher or lower respectively.

In subsection 5.1.4 it was theorized that the feasible space of the DHS with the heat storage tank and the DHS without the heat storage tank could have similarities because both DHSs have the capacity to store heat. It can be seen that the feasible spaces found in the experiments on the DHS without storage tank do not look very similar to the feasible spaces for the DHS with storage tank subsection 5.1.2. Most notably, the feasible space in this section does not display the same flexibility gained by using the heat storage tank. The effect where increasing the supply temperature in one time step allows for lower feasible temperatures in a later time step is not observed. The absence of this effect can be attributed to the way that the heat delivery violations are determined in the simulator. Increasing the temperature at t_0 will not allow for a lower feasible temperature of a later time step t_x when $plug_x$ is still consumed during the same time step(s), since the heat delivery constraint is violated if at any point during one time step the temperature of the water is below the minimum supply temperature.

Although flexibility similar to that found for the DHS with heat storage tank was not found for the DHS without in this section, it can be reasoned that the non-convex areas in the feasible space can still provide some flexibility. As illustrated by the example in Figure 5.12, increasing the temperature of one plug can cause subsequent plugs to be consumed later. This effect has been observed in the feasible space for heat demands from the data set in Figure 5.11a, but in a small scale. However, when considering the effect that the supply temperatures of multiple time steps can have on the feasible temperature of a subsequent plug, this effect could become significant. The temperature settings of multiple time steps can together influence the mass flow for a longer time period and therefore have a bigger effect on the time delay of a subsequent plug $plug_x$. If the heat demand is fluctuating during the time steps that $plug_x$ can arrive at the consumer depending on these earlier temperatures, then the feasible temperature settings of $plug_x$ also change. It is expected that these changes in feasible temperatures for $plug_x$ will show up in the feasible space as non-convex areas similar to the ”hook” and ”extra strip” non-convexities. It is also expected that these non-convex areas will appear more often in the feasible space, since the variance of the possible time delays of $plug_x$ is now larger. These non-convexities can cause flexibility in the following scenario: when it is beneficial for the desired electricity production of the CHP to produce less heat during time step t_x, then the temperature settings of the earlier plugs can be chosen such that $plug_x$ arrives at the consumer during a time step where the heat demand is low.

Following this reasoning, it is likely that the feasible space of the supply temperatures for all time steps in the predictive horizon will contain more non-convex areas. The size of these non-convex areas and how often they occur, however, are unknown. It is expected that both depend on how much the heat demands of the consumer fluctuates. What also should be taken into account is the heat loss, which increases with longer time delays. This could affect how beneficial it is to increase the supply temperature, since that increases the time delay of subsequent plugs.

Lastly, it is important to note that it is possible that feasible spaces of the supply temperature of two time steps exist that have a different shape than what has been observed in this chapter. One reason for this is that the feasible spaces that were found were all generated using heat demands from the provided data set. It is therefore possible that other heat demands exist that have different or more significant non-convexities in their feasible spaces. However, these non-convexities are only important if they occur for realistic heat demands. If this is not the case, they will never be encountered in real scenarios. Another reason is errors in the feasible spaces due to how difficult it is to prove that two supply temperatures cause the solution to be infeasible. Using the current approach to determine the feasibility, it is possible that two temperature settings are marked infeasible even though different temperature settings for the rest of the predictive horizon would make the solution feasible. Proving that two temperature settings are infeasible therefore in some scenarios requires testing the feasibility of all possible temperature settings for the rest of the predictive horizon. Since this requires checking exponentially many solutions, this has not been done in the current work. However, assuming that this scenario does not happen frequently, the feasible spaces that were found were correct.

6 Discussion

In this chapter the results from the previous chapters will be used to answer the research questions of this work (see chapter 1). First the results from chapter 4 will be discussed in section 6.1, followed up by the discussion of chapter 5 in section 6.2. Then in section 6.3 the main research question of this work is covered. Finally in section 6.4, the possibilities of future work are discussed.

6.1. Representation power of input convex neural networks

In chapter 4 the representation power of an ICNN is tested in order to answer the subquestion of this thesis: How well can an ICNN learn convex and non-convex functions?

From the definition of the ICNN it is expected that ICNNs should be able to learn convex functions with a good accuracy. Therefore the ability of ICNNs to learn convex functions is only confirmed by training on one instance of a convex function.

To test how well ICNNs are able to learn non-convex functions, ICNNs have been trained on multiple instances. The results indicate that the performance of the ICNN is highly dependent on the exact shape of the non-convex function. Functions with one significant minimum can be learned more accurately than functions with two significant minima. However it is also reasoned that there will always be errors in the function learned by the ICNN, since at best errors can be ”balanced” for non-convex areas.

When an ICNN trained on a non-convex function is used in an optimization problem, the errors could have different consequences depending on how it is used. If an ICNN is used to learn a non-convex objective function, it is expected that the optimum of the learned function has an error. This error could potentially be reduced by using a loss function similar to the WMSEs defined in section 4.2, but it is likely that this optimum will stay inaccurate because of error minimization elsewhere in the function. Depending on the problem, this error in the optimum could be acceptable or not. If an ICNN is used to learn a non-convex constraint, however, the consequences of errors could be more dangerous. For example, if the non-convex function in Figure 4.5 would have a safety constraint $f(x) \leq 0.5$, then the learned function in Figure 4.6a would give an incorrectly infeasible solution for $x = 0.2$ and an incorrectly feasible solution for $x = 0.4$.

6.2. Feasible space of supply temperatures of the district heating network

In chapter 5 the feasible space of supply temperatures to DHN is generated for scenarios with different heat demands, in order to answer the subquestion: What does the feasible space of the supply temperatures of a DHN look like?

In this chapter the feasible solution space for two time steps is studied of a small DHS with one producer and one consumer. Initially a DHS with a heat storage tank without time delays is considered. The shape of the feasible space of the heat production in this DHS is convex and shows the effects of being able to use heat storage.

Then the feasible space of the supply temperatures of two time steps is considered for a small DHS with time delays. The feasible space is determined for multiple heat demands from a data set using a simulator of the DHS. The shape of the space was mostly found to be constrained by violations of the temperature ramping constraints. Additionally, the constraints that ensure that the heat demands of the consumer are fulfilled constrain the supply temperatures to be above a minimum.

For most of the heat demands a feasible space was found that looks to be convex. However, for some heat demands the space contains a non-convex area. The circumstances that cause these non-convexities seem simple and could occur regularly in real life scenarios. How significant it is for these non-convexities to be included in the feasible space for the optimization is unknown. The fact that the non-convex areas found in the feasible space often only occur once temperature increases of 0.2 °C were considered suggests that the non-convex area might not be relevant when the minimum temperature increase or decrease used for operation is 1 °C. Furthermore, the sizes of the non-convex areas that were found were small and could therefore potentially be left out of the feasible space without major consequences during optimization. Alternatively, it can be reasoned for these small non-convex areas that a few infeasible solutions could be included in the feasible space in order to make it convex. These small-scale non-convex areas are caused by very small violations of the heat delivery constraints. If the heat demands of the consumers are fulfilled the rest of the time, such smaller violations might be acceptable.

However, when considering the feasible space for 24 time steps for a larger DHS, more and bigger non-convexities might appear for the following reasons:

Firstly, increasing the amount of time steps that is considered could result in more non-convex areas in the feasible space. As discussed in subsection 5.2.5, the non-convexities that were found for the feasible space for two time steps are a result of the effect that supply temperatures have on the time delays of subsequent plugs. For 24 time steps, the time delays of later plugs are affected by the cumulative effect that earlier temperature settings have on the delay in the network. If heat demands fluctuate during the time steps that these plugs can be consumed, it is expected that multiple non-convexities similar to the ones discussed in subsection 5.2.4 could occur in the feasible space.

Secondly, when a larger network with more consumers is considered, the fact that one plug is now partially consumed by multiple consumers could cause an increase in non-convex areas in the feasible space. In larger networks it is likely that plugs are consumed by consumers during different time steps, since the distances to the producer can be different and the mass flows at the consumers can be modified individually. A change in the time delay of a plug then also affects when the plug is consumed by all consumers. So because of the likely difference in arrival times, it is expected that changes in the time delay of a plug (caused by changes of earlier temperature settings) can cause feasible temperature settings to change more often. This could potentially result in more non-convex areas in the feasible space.

Lastly, in a larger network with more pipes it is possible that mass flow violations could influence the feasible space of the supply temperatures. During the experiments on the feasible space of the small DHSs in subsection 5.2.2 and subsection 5.2.3, no violations of the mass flow constraints have occurred. Due to the relatively large diameter of the supply and return pipe, the heat delivery constraints have always been violated before the mass flow in the pipes could become too large. In a larger network with pipes of different diameters, mass flow violations could occur when the temperatures of plugs that are consumed simultaneously require a mass flow that is too large for one of the pipes in the network. However, in what way these violations restrict the shape of the feasible space is unknown.

6.3. Learning the feasible space of supply temperatures of a district heating system with an input convex neural network

In previous sections, the subquestions of this research have been answered in order to address the subproblems of the main research question. In this section, the main research question of this work will be answered: How well can the feasible space of supply temperatures of a district heating system be learned by an input convex neural network?

During the experiments on the representation power of ICNNs it was observed that ICNNs are better able to approximate functions that are not highly non-convex. Furthermore, the experiments on a small DHS with two time steps have resulted in feasible spaces of which the majority was completely convex, with some exceptions where relatively small non-convex areas were found. Therefore, it is expected that an ICNN should be able to learn these feasible spaces or the constraints which they are based on with relatively good accuracy. It is possible, however, that this ICNN then classifies some solutions as feasible when in fact they are not. In order to ensure that no constraints are violated, it would be safer for the ICNN to be trained on the convex subset of feasible solutions. This would mean that some feasible solutions are excluded from the feasible space. The method of training an ICNN on the convex subset of the feasible solutions is not covered in this work.

6.4. Future work

In section 6.2 it was reasoned that more and bigger non-convexities might appear in the feasible space for 24 time steps for a larger DHS. This theory has not been tested, however, so it is unknown how significant these non-convexities will be. This makes it hard to reason about how well the optimization using the constraints learned by an ICNN would perform. It could therefore be useful to do experiments on the feasible space of larger DHSs. Experiments on the feasible space for more than two time steps might not be practical, since it would take much more time to determine the feasibility of all temperature combinations and since the resulting feasible space would be much harder to interpret due to a higher dimensionality.

Furthermore, in section 6.3 it was reasoned that it would be the safest to train an ICNN on the convex subset of feasible solutions. How this could be achieved, was not discussed. Therefore it could be useful to do experiments to determine how an ICNN can be trained so that it is as accurate as possible, while never classifying a solution to be feasible when in fact it is not. This point of concern was also mentioned by Yang and Bequette [16]. In their case study inaccuracies in their constraints learned by the ICNN happened to lie on the safe side, but they warn that care should be taken when training an ICNN on constraints.

The next step could be to train an ICNN on the network dynamics and to use this in the operational optimization. Similar to Chen et al. [14], an ICRNN could be used to learn states in the DHS that depend on the network dynamics during one time step, given the previous supply temperatures to the DHN. The constraints defined on these states can be used in the MPC problem. Since the pumping costs depend on the mass flow in the network, the objective function would also depend on the states in the DHS. However, there are some aspects of this approach that may prove difficult:

Firstly, it has to be determined which states in the DHS have to be included in the ICRNN. The mass flows in all pipelines in the DHN and the supply temperatures at all consumers all depend on the network dynamics and it would therefore make sense to have these included in the DHS states. However, the amount of states that this produces might be hard to learn, especially in larger networks. And considering that the discretization of the supply temperatures at the consumers for each time step causes the effect that temperatures of individual plugs have to be lost, these states might not be detailed enough.

Secondly, it is expected that a very large data set is needed for the ICRNN to be trained properly. The relationship between the input and output of the ICRNN is complicated; the states in the network depend on time delays caused by earlier supply temperatures to the DHN and on the behaviour of all consumers, which also depends on their heat demands. If not enough training data is used, the ICRNN might be overfitted and will not learn the relationship correctly.

Lastly, it has to be considered which method can be used to optimize the MPC problem with the ICRNN. Amos et al. [13] argued against posing the the optimization of their ICNN as a linear program because of the large number of variables, but have not performed experiments on it. Alternatively, a gradient-based optimization method could be used for the MPC problem [16]. These methods have the advantage that the gradient of the input of the network can easily be computed using backpropagation [13].

Bibliography

[1] B. den Ouden, J. Kerkhoven, J. Warnaars, *et al.*, "Klimaatneutrale energiescenario's 2050," *Berenschot & Kalavasta*, pp. 1–146, 2020.

[2] K. Difs, M. Danestig, and L. Trygg, "Increased use of district heating in industrial processes–impacts on heat load duration," *Applied Energy*, vol. 86, no. 11, pp. 2327–2334, 2009.

[3] L. Giraud, M. Merabet, R. Baviere, and M. Vallée, "Optimal control of district heating systems using dynamic simulation and mixed integer linear programming," in *Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017*, Linköping University Electronic Press, 2017, pp. 141–150.

[4] Z. Li, W. Wu, M. Shahidehpour, J. Wang, and B. Zhang, "Combined heat and power dispatch considering pipeline energy storage of district heating network," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 1, pp. 12–22, 2015.

[5] L. Merkert, A. A. Haime, and S. Hohmann, "Optimal scheduling of combined heat and power generation units using the thermal inertia of the connected district heating grid as energy storage," *Energies*, vol. 12, no. 2, p. 266, 2019.

[6] L. Stegman, "Optimising district heating operations," M.S. thesis, 2019.

[7] K. Stepanovic, J. Wu, R. Everhardt, and M. de Weerdt, "Unlocking the flexibility of district heating pipeline energy storage with reinforcement learning," *Energies*, vol. 15, no. 9, p. 3290, 2022.

[8] A. Benonysson, "Dynamic modelling and operational optimization of district heating systems," 1991.

[9] E. Guelpa and V. Verda, "Thermal energy storage in district heating and cooling systems: A review," *Applied Energy*, vol. 252, p. 113 474, 2019.

[10] E. Abdollahi, H. Wang, S. Rinne, and R. Lahdelma, "Optimization of energy production of a chp plant with heat storage," in *2014 IEEE Green Energy and Systems Conference (IGESC)*, IEEE, 2014, pp. 30–34.

[11] A. Bloess, "Modeling of combined heat and power generation in the context of increasing renewable energy penetration," *Applied Energy*, vol. 267, p. 114 727, 2020.

[12] H. Pálsson, H. V. Larsen, B. Bøhm, H. F. Ravn, and J. Zhou, "Equivalent models of district heating systems for on-line minimization of operational costs of the complete district heating system," 1999.

[13] B. Amos, L. Xu, and J. Z. Kolter, "Input Convex Neural Networks," en, in *Proceedings of the 34th International Conference on Machine Learning*, ISSN: 2640-3498, PMLR, Jul. 2017, pp. 146–155. [Online]. Available: `https://proceedings.mlr.press/v70/amos17b.html` (visited on 01/03/2022).

[14] Y. Chen, Y. Shi, and B. Zhang, "Optimal control via neural networks: A convex approach," *arXiv:1805.11835 [math]*, Feb. 2019, arXiv: 1805.11835. [Online]. Available: `http://arxiv.org/abs/1805.11835`.

[15] F. Bünning, A. Schalbetter, A. Aboudonia, M. H. de Badyn, P. Heer, and J. Lygeros, "Input convex neural networks for building mpc," in *Learning for Dynamics and Control*, PMLR, 2021, pp. 251–262.

[16] S. Yang and B. W. Bequette, "Optimization-based control using input convex neural networks," *Computers & Chemical Engineering*, vol. 144, p. 107 143, 2021.

[17] J. Wu, R. Everhardt, K. Stepanovic, and M. de Weerdt, "Simulation of the district heating network with a computer program," Flex-Technologies, Utrecht, The Netherlands. 2022, (manuscript in preparation; to be submitted), 2022.

[18] K. Stepanovic and J. Wu, *Flex_heat*, Available online: `https://github.com/AlgTUDelft/flex_heat`, version: 0.1.0., Mar. 2022.

[19] O. Ruhnau, L. Hirth, and A. Praktiknjo, "Time series of heat demand and heat pump efficiency for energy system modeling," *Scientific data*, vol. 6, no. 1, pp. 1–10, 2019.