Master's thesis

2021

Master's thesis

Ask Norheim Morken

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

Ask Norheim Morken

Using Machine Learning for Improved Production Planning in the Concrete Supplier Business

July 2021

NTNU
Norwegian University of Science and Technology

NTNU
Norwegian University of Science and Technology

NTNU
Norwegian University of
Science and Technology

Using Machine Learning for Improved Production Planning in the Concrete Supplier Business

Ask Norheim Morken

Engineering and ICT
Submission date: July 2021
Supervisor: Marco Semini

Norwegian University of Science and Technology
Department of Mechanical and Industrial Engineering

Acknowledgments

This master's thesis is submitted as the final work of the Master's Degree Programme Engineering and ICT with the main profile of ICT & Operation Management at the Department of Mechanical and Industrial Engineering at NTNU.

I would like to thank my supervisor, Marco Semini at NTNU, for his excellent consultation and feedback throughout the project.

I am grateful for the frequent feedback I was able to get from Overhalla Betongbygg. Many thanks to Didrik Ytterdahl Evensen, controller at Overhalla Betongbygg, for his very useful comments throughout the development process.

I would like to thank Ola Lauve, project leader of Skogmo Industry Park AS, for coordinating the trip to Skogmo and Overhalla Betongbygg from which I obtained a lot of useful information.

Lastly, I would like to thank Overhalla Betongbygg, Skogmo Industry Park, the Department of Mechanical and Industrial Engineering at NTNU and Marco Semini for providing me with this great project.

Ask Northeim Morken

July 2021

Oslo

Table of contents

List of figures

List of tables

1 Introduction

1.1 Purpose and objective

This project is a continuation of my specialisation project performed during the autumn semester of 2020. In the thesis of the specialisation project, I explored the opportunities artificial intelligence could provide within operations management in several businesses within Skogmo Industry Park. Of these businesses, Overhalla Betongbygg was the one where the most promising and relevant opportunities were identified. Therefore, exploring the opportunities artificial intelligence could provide within operations management at Overhalla Betongbygg in more depth was a natural continuation of my specialisation project. At Overhalla Betongbygg, the most promising opportunities for artificial intelligence were found to be within production planning. Therefore, the aim of this project was set to further research how application of artificial intelligence can be beneficial within the production planning at Overhalla Betongbygg.

Through research, I have been able to find some literature on the topic regarding the use of artificial intelligence in production scheduling. A lot of the literature were somewhat aged however, with mainly old methods from artificial intelligence researched. I was however able to find some literature researching more recent methods from artificial intelligence and machine learning. In a field with frequent improvements and scientific breakthroughs like artificial intelligence, aged literature might be missing key opportunities discovered later than the time of writing and therefore not be as relevant as of today.

As stated by Mula et al. (2006), for complex processes, analytical approaches to production planning is typically replaced by methodologies based on artificial intelligence and simulation. The production planning at Overhalla Betongbygg has been found to have multiple complex processes. These are the processes I have been looking to apply artificial intelligence to aid in.

Artificial intelligence is defined as “a field of science and engineering concerned with the computational understanding of what is commonly called intelligent behaviour, and with the creation of artefacts that exhibit such behaviour" (Ramesh et al., 2004). That is, with artificial intelligence, the goal is to make machines act intelligently, as though it was thinking. Almost like a human. With artificial intelligence, a machine can train and learn to act increasingly intelligently over time. When combining such intelligent behaviour with the computational power of today's computers, many possibilities open up. A machine can consider way more data than a human could do within a reasonable amount of time, often giving it a better basis for making a good decision.

According to Van Dierdonck and Miller (1980), one of the primary functions of production planning is to match the market demand with supplies from production and external vendors.

In make-to-stock manufacturing, there will be an uncertainty connected to this marked demand at the time of production planning. The production planning will therefore be based on forecasting of marked demand.

This uncertainty is not present in make-to-order and engineer-to-order manufacturing. The demand to match with supplies form production and external vendors will simply be the demand of the orders they have received.

However, especially engineer-to-order manufacturing, will generally have other, more complex processes within production planning. This is mainly due to the complex products, the uncertainty and the variation in production processes engineer-to-order manufacturing often have (Rauch et al., 2018). In make-to-stock manufacturing, the products to be produced are known, while in engineer-to-order manufacturing, a new order can introduce all new products and components to be produced, often requiring more preparation of the production processes. This increased uncertainty and complexity of engineer-to-order production planning might often cause traditional approaches to production planning insufficient and might make approaches like artificial intelligence methods more applicable.

Buer et al. (2016) proposed the following definition for the production planning environment:

"The production planning environment is the sum of internal and external variables that influence the production planning and control process."

Naturally, these internal and external variables will differ from business to business. Therefore, production planning environment is business specific.

1.2 Research questions

RQ1: What are the characteristics of the production planning environment and how is production planned at Overhalla Betongbygg?

RQ2: What are the challenges within production planning at Overhalla Betongbygg?

By answering these questions, I aim to provide an overview of the current state of the production planning at Overhalla Betongbygg. I want to discover which processes are executed in the production planning, and the general workflow for these. I also want to identify processes within the production planning where improvements can be made and what challenges the production planners face both in production planning and in attempts to improve efficiency of production planning.

RQ3: How can artificial intelligence be used to improve production planning?

By answering this question, I want to describe which approaches utilising artificial intelligence I have found to be useful to improve production planning. This will include both approaches found by studying relevant literature and what I have learned in my attempts to use artificial intelligence to improve production planning myself.

1.3 Research scope

The scope of this master thesis project is narrowed down from the scope of my specialisation project that this project is based on.

From focusing on all of operations management in my specialisation project, this project will rather have its focus restricted to production planning, a topic under operations management.

The research done in my specialisation project included multiple businesses in Skogmo Industry Park. This project will only include one of these businesses, more specifically, Overhalla Betongbygg.

1.4 Structure

The following chapter will describe the methodology used in my research. I will present the methods that was used and a more detailed description of my usage of them.

Next follows a chapter presenting relevant theory discovered through the literature study conducted as part of this project.

In the chapter following, the state of the production planning at Overhalla Betongbygg will be described.

Next follows a chapter describing the tool development work done as part of this project.

In the next chapter, the results of the tool development work are presented.

Then a chapter with discussion on the findings of my project work is presented.

Last is a chapter with conclusions of my research. Here the contribution and limitations of the research is presented as well as opportunities for further work.

2 Methodology

In this project, both quantitative and qualitative research methods will be used.

Firstly, a literature review will be performed to identify how artificial intelligence has been used to aid in similar challenges to those in the scope of this project.

The project will also include a case study of the business that I will work with during this project in order to get a more detailed understanding of the procedures and challenges in the production planning in the business.

Furthermore, a goal of my case study was to identify the challenges where application of artificial intelligence could be beneficial.

A significant part of the work in this project will be attempting to develop a tool that can, with use of artificial intelligence, be used to aid in production planning in the case business. A part of developing a tool like this will be testing out multiple artificial intelligence methods on use-cases within production planning and examining the results they provide. These results will be compared to existing solutions in this report to present how much of a benefit the different artificial intelligence methods can provide.

2.1 Literature review

Early on in my literature review, I mainly used the search words listed under “Main search words” in Table 1. These search words are relatively broad.

I used these to get an initial overview of the literature available regarding these broad topics and to find more specific terms relevant for my project that I could use to make more specific searches for literature. The search words found this way is listed under “Additional search words” in Table 1.

Conducting the literature review in this fashion helped me get an overview of the broad topics while later finding more information on more specific topics I found fitting for this research project.

Besides using search words found to created new literature searches, the literature found by using the initial broader search words would often briefly mention several more detailed pieces of literature on more specialised topics. This way, these papers providing an overview over these broader topics pointed me towards literature even more relevant for my research. While the papers providing overviews were often largely based on literature reviews, the more specialised papers discussed in these overviews were more often practical applications of the technology discussed.

Main search words	Additional search words
Production planning Artificial intelligence	Production scheduling Supervised learning Reinforcement learning Engineer to order Production planning environment Smart planning and control Production planning and control (PPC)

Table 1: Literature review search words

2.2 Case study

2.2.1 Identifying possible applications of artificial intelligence

In my case study conducted in this project, as well as the case studies conducted as part of my specialisation project, the basis for this master project, I was always looking for challenges where artificial intelligence seemed to be promising as a potential solution.

One important thing I tried to look for when identifying suitable applications for artificial intelligence was the complexity of the problem. Are there many different factors to consider when trying to solve the problem? If there are not, then artificial intelligence would often be "overkill" for the problem, and a simpler solution might be more suitable. If a simpler, more traditional approach is able to find an optimal solution to a problem, then there would not be much of a reason to use artificial intelligence.

However, a decision to be made being complex is not in itself a very good argument for applying artificial intelligence. Such problems do not necessarily require artificial intelligence to get the best result. Multi-objective optimisation is especially suitable for problems like these.

When there is also a level of uncertainty involved, then matching the results methods within artificial intelligence can provide becomes way more difficult using more traditional mathematical methods.

Methods within artificial intelligence will have different requirements for being applicable. However, common for all applications of artificial intelligence is that having large amounts of data available is important for having a good starting point.

Alternatively, one should look for the opportunity to gather such data. For my project, it was important to consider how time consuming such data gathering would be.

With supervised learning, labels are required for training. That means that we must already know the best solution to the problem we try to solve for our training data. If we try to train a model to be able to predict prices of houses, then we must have the actual prices of the houses our example data consists of in order to be able to train a model.

With reinforcement learning, a label that is used in supervised learning is not needed. However, there needs to be a way to calculate how good an output made by the machine learning model is. Instead of training the model based on how close the model output is to a label, the model will be trained using the reward calculated in another way.

Because of the requirements for supervised and reinforcement learning stated above, when looking for possible opportunities for artificial intelligence application, I analysed whether the best solution to the problem would already be available or whether I could calculate how suitable a model output would be in another way.

3 Theory

3.1 Artificial intelligence

Within artificial intelligence there are multiple branches. Perhaps the most significant one of these branches, especially in later years, is machine learning. An algorithm using machine learning have the ability to, with the use of input data, achieve a desired task without having how to complete the task defined by programming (El Naqa and Murphy, 2015). With traditional programming, the developer will code how the program is going to transform the input to produce the desired output.

So how can a program using machine learning do such transformations from input to output in a way that is useful without having the transformation process defined by code? The answer is that it learns how to do it. By providing both the input and the corresponding desired output, a machine learning algorithm will be able to use these to train and by training with these examples, estimating the transformation process.

This way, machine learning algorithms are able to perform predictions based on a data set.

3.1.1 Supervised learning

Supervised learning is probably the most simple, common, and well-known form of machine learning. With supervised learning, the algorithm aims to learn a target function which transforms the input into the desired output (Muhammad and Yan, 2015). A model is trained to learn such functions with example data with corresponding labels. A label is the correct output for a given input. In the training process, the model will transform the input into an output and compare this output to the provided label. The model will then adjust depending on how close the output was to the label. Over time, the model will be able to improve its performance, that is calculate outputs increasingly similar to the label, given there is dependency between the input and the output. With enough training data, a model can often achieve a good estimation of the actual function from the input to the output if such a function exists.

3.1.1.1 Regression and classification

As shown in Figure 1, we separate between two kinds of problems in supervised learning. These are regression and classification.

Regression is used when trying to predict a continuous numerical value. An example of a regression problem is predicting the value of a house based on its features. Since a house can have any numerical value, we want a model that can transform the input properties into such a value in a continuous space.

On the other hand, with classification, we want to predict which of some predefined classes a given instance belongs to. An example of a classification problem is to predict the language of a sentence.

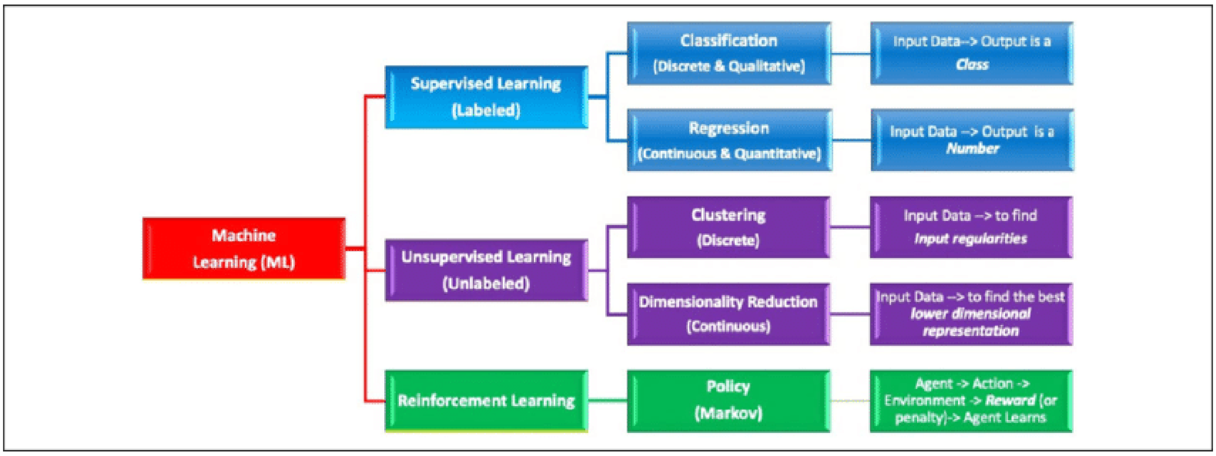

Figure 1: Overview diagram of machine learning algorithms (Rashidi et al., 2019).

3.1.1.2 Loss functions

In supervised learning, the loss function is of high importance. The loss function is the function used to calculate how bad the outputs from the model was given the labels. The larger the loss, the larger the adjustments made on the model will be. Different loss functions are used for regression and classification.

3.1.2 Reinforcement learning

Reinforcement learning is another type of machine learning. It can be defined as a learning paradigm concerned with learning to control a system so as to maximise a numerical performance measure that expresses a long-term objective (Szepesvári, 2010). It differs from supervised learning by not requiring examples of the desired output, or as it is often referred to as in machine learning, labels. Instead of labels, reinforcement learning algorithms learns by receiving a reward from the environment it is interacting with. The reinforcement learning algorithm will use this reward to determine how good its output, or action as it is more commonly called in reinforcement learning, was.

The program using reinforcement learning in such a way is often referred to as an agent. The agent will receive input, or a state and transform this into an output, the action to be applied on the environment. Thereafter, the agent will receive a new state, reflecting the impact the action had on the environment as well as a reward, signalling how successful the action was. The reinforcement learning algorithm will use these rewards to learn to make increasingly good actions given the state of the environment. This process is shown in Figure 2.

A complete experiment with a reinforcement learning environment is often referred to as an episode. Each time the environment receives an action from the agent and returns a state and a reward, we call it a step. For each step, the environment will also inform the agent whether the end of the episode has been reached. If it has, a new episode can be started with the environment reset to its initial state.

The characteristics of reinforcement learning makes it suitable to learn machines how to play games.

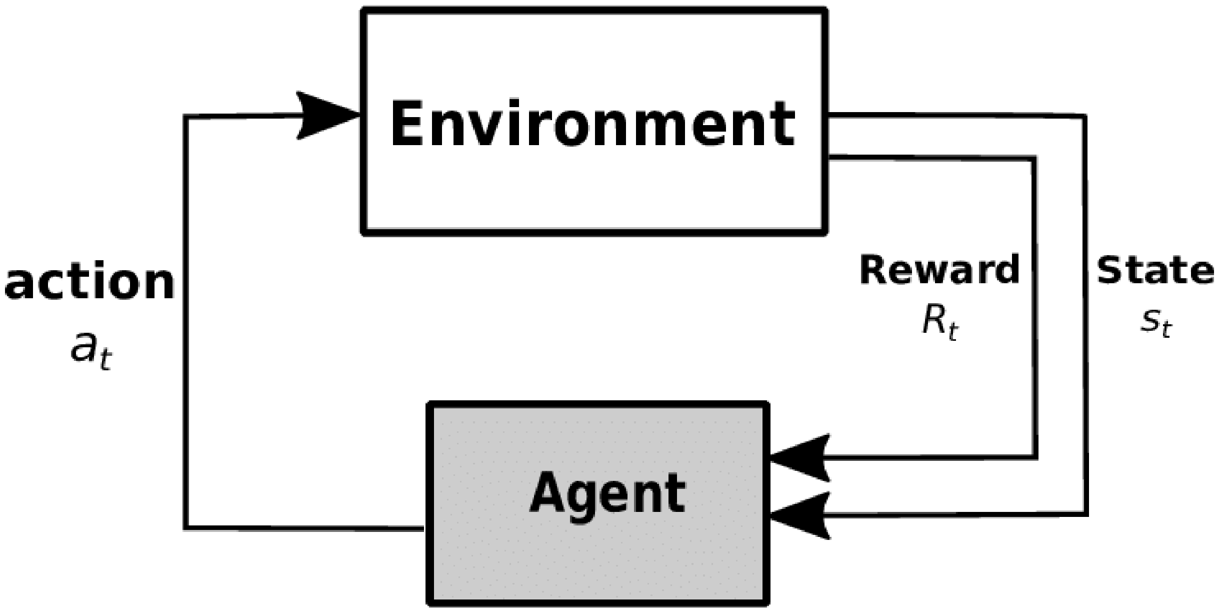

Figure 2: Reinforcement learning: Agent and Environment (Amiri et al., 2018).

3.1.2.1 Policy-based vs Value-based approaches

A policy in reinforcement learning is a mapping from a state or observation to an action. With a traditional policy-based approach to reinforcement learning, such a mapping will be built during training and stored in memory (Guan et al., 2019).

On the other hand, we have value-based approaches which differs from policy-based approaches. With value-based approaches, a function estimating the returned reward for taking an action in a given state is being fitted during training (Guan et al., 2019). The output of this function is referred to as Q-values. Instead of outputting an action given a state like with policy-based approaches, value-based approaches will derive an action by optimising the Q-values.

Both policy-based and value-based reinforcement learning can be implemented via many different algorithms that have been developed. The most known algorithm from policy-based reinforcement learning is policy gradient (PG) (Sutton et al., 1999), while the most known value-based algorithm is deep Q-learning (DQN) (Mnih et al., 2013). Figure 3 display a collection of algorithms from both these branches of reinforcement learning. In addition, several algorithms attempting to combine policy-based and value-based reinforcement learning has been developed (Nachum et al., 2017, Lillicrap et al., 2015, Christodoulou, 2019, Haarnoja et al., 2018a, Haarnoja et al., 2018b). Some of these are also shown in Figure 3. The most notable of these is Deep Deterministic Policy Gradient (DDPG).

As mentioned, many different algorithms have been developed for the different approaches to reinforcement learning. In many cases, new algorithms that are presented will use one or more existing algorithms as foundation for their algorithm. This way, reinforcement learning has been able to experience relatively frequent incremental improvements in recent years as well as the more ground-breaking new approaches or ideas that do not come around as often, but still not too rarely in the fast-evolving field of machine learning that reinforcement learning is.

One notable algorithm presented relatively recently is Proximal Policy Optimization (PPO). Rather than just an algorithm, the original paper presents PPO as "a new family of policy gradient methods for reinforcement learning" (Schulman et al., 2017). As this can imply, PPO builds upon and provides improvements on policy gradient, an algorithm presented all the way back in 1999 (Sutton et al., 1999).

3.1.2.2 Model-based vs Model-free reinforcement learning

An important distinction to make in reinforcement learning is whether the agent has access to a model of the environment or not. With a model of the environment, we refer a function that given the current state can predict the next state and the reward for each possible action (OpenAI, 2018).

If a model of the environment is available, the agent will be able to plan ahead and explore multiple possible actions and considering the states and rewards they would result in before deciding on an action.

However, the outcome of an action in a state will not always be known in advance. This does however not mean that using a model of the environment is not possible. There are also algorithms used to learn a model of the environment from training. The biggest challenge with this approach is when the model of the environment is imperfect, the agent might learn to take advantage of this. The agent might learn to perform well when interacting with the learned model of the environment, but then perform worse when interacting with the real environment as it has learned to take advantage of characteristics of the model that is not present in the real environment.

Reinforcement learning where a model of the environment, whether defined in advance or learned from experience, is used, is called model-based reinforcement learning while when such a model is not used, it is called model-free reinforcement learning. Model-free reinforcement learning is the easiest to implement and the most used (OpenAI, 2018). As shown in Figure 3, both policy-based (policy optimization) and value-based (Q-learning) reinforcement learning is branches of model-free reinforcement learning.

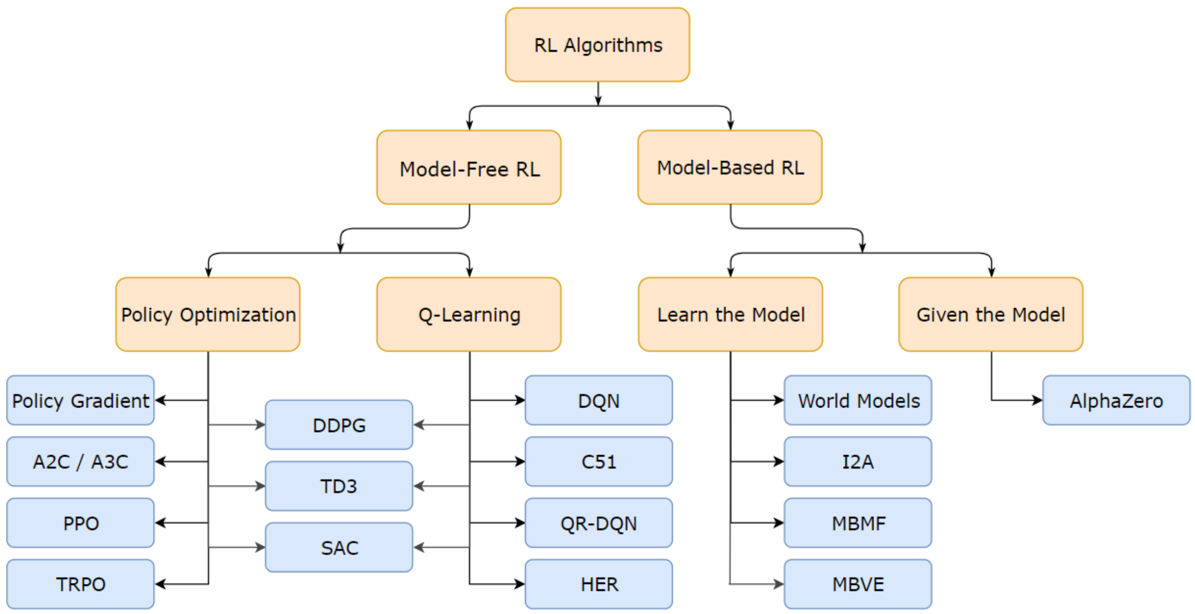

Figure 3: A Taxonomy of RL Algorithms (OpenAI, 2018).

3.2 Artificial intelligence in production planning

Enterprise resource planning systems (ERPs) has been implemented in an ever-increasing number of businesses in recent years. However, the vast majority of these systems has no support for production scheduling functions. Many different methods within artificial intelligence have been researched and used for production scheduling. The first of these was expert system (Metaxiotis et al., 2003).

With expert systems, the idea is to transfer expertise from a human to a machine. This expertise knowledge is then stored in machines to be used later to offer advice on decisions and even provide the logic behind the advice if necessary (Liao, 2005).

Following the use of expert systems in production scheduling was the use of hybrid expert systems. With hybrid expert systems, expert systems were combined with neural networks (Nikolopoulos and Fellrath, 1994). One way to benefit from combining these is to have the neural network be used for classification before an expert system is used for drawing a conclusion.

The use of neural networks has also been proposed as a method of solving scheduling problems on its own.

Furthermore, the applications of genetic algorithms in scheduling were studied by a handful of researchers during the 1990s (Mula et al., 2006). Several potential uses of genetics algorithms were proposed.

Mula et al. (2006) provides a review of models for production planning under uncertainty. Literature from 1983 to 2004 was used in the compilation of this review. The review contains numerous citations to artificial intelligence models applied to the research topics aggregate planning, material requirement planning, manufacturing resource planning, inventory management and supply chain planning. Only analytical models were presented for hierarchical production planning and only analytical models, as well as one simulation model, for capacity planning.

The paper shows a clear upwards trend for applications of artificial intelligence in production planning under uncertainty. Among the 34 citations from the period 1990-1999, 8 of these were references concerning models based on artificial intelligence. In the period 2000-2004, this number had grown to 10 of a total of 19 references. This is an increase from about 24 percent of the citations in 1990-1999 to about 53 percent of the citations in 2000-2004 (Mula et al., 2006).

Within aggregate planning, the application of fuzzy modelling has been research and described in a series of publications (Rinks and DB, 1981, Rinks, 1982, Turksen, 1988a, Turksen, 1988b, Ward et al., 1992, Gen et al., 1992, Wang and Fang, 2001). The reason for the research done on this was questioning of the probabilistic approach. With this approach, no distinction between randomness and imprecision was made. With fuzzy modelling, on the other hand, this distinction is made.

Also, within material requirement planning, fuzzy logic is the most significant artificial intelligence method applied from early on. As early as in 1987, Lehtimaki (1987) studied the master production schedule in material requirement planning where a fuzzy level of customer satisfaction was maximised. (Chih-Ting Du and Wolfe, 2000) presents an active, bucket-less, real-time material requirement planning system using object-oriented databases together with fuzzy logic controllers and neural networks. Fuzzy logic controllers are combined with object-oriented databases for an integration of dynamic and static knowledge. Artificial neural networks are then used to simulate fuzzy membership functions by learning if-then rules. Neural networks were also used together with fuzzy logic controllers for inventory classification.

Manufacturing resource planning has also seen models based on fuzzy logic in research from the 1980s, 1990s and early 2000s (Sommer, 1981, Miller et al., 1997, Hsu and Wang, 2001, Reynoso et al., 2002, Mula, 2004). These references do apply fuzzy logic in different ways to handle different types of uncertainties. However, since these does not examine any artificial intelligence methods besides fuzzy logic, they will not be discussed further in this thesis.

Inventory management is another area of production planning where models based on fuzzy logic has been applied from relatively early on (Kacprzyk and Stanieski, 1982, Park, 1987). However, within inventory management, we also see artificial intelligence methods besides fuzzy logic applied. (Porter et al., 1995) describes a solution using a genetic algorithm to determine optimal stock levels, production quantities, and transportation quantities in an attempt to minimise cost in an inventory-production-distribution problem.

Besides some literature on application of fuzzy logic, within supply chain planning there are a couple of papers presenting usage of software agents (Chu et al., 1998, Fox et al., 2001). These solutions are built up of multiple agents acting within different areas with the ability to communicate with each other.

In a newer piece of literature, (Yuldoshev et al., 2018) presents an analyse of methods of artificial intelligence applied in production management systems. In this paper, an application of genetic algorithms to schedule production is described. Furthermore, the paper discusses usage of neural networks. It states that neural networks are a very powerful and flexible mechanism for planning and forecasting. The paper stresses the importance of choosing the right data for analysation and prediction and the right number of variables. In other words, it stresses the importance of data preparation and feature engineering. Secondly, the paper stresses the importance of three variables in a neural network planning system: The planning period, the planning horizon, and the planning interval. These are described as the basic unit of time for which a forecast is made, number of periods in the future that covers the forecast and the frequency with which a new forecast is made.

Machine learning does together with other digital technologies such as internet of things (IoT) and big data analytics present opportunities in production management by enabling more frequent replanning of production to reflect the real-time situation changes within factories and supply chains (Oluyisola, 2021).

In their paper on smart production planning and control, Oluyisola et al. (2020) concludes that which smart industry 4.0 technology that is the most useful and beneficial for a business will depend on the production characteristics of the business. They argue that business producing products in lower volumes and with a wide range of variants in their products, maybe even one-of-a-kind products, and a job shop process structure should pursue smart product strategy. Meanwhile, businesses producing products in high volumes of highly standardised products with a continuous flow process structure should pursue smart process strategy according to this paper.

Bueno et al. (2020) explored the impact of smart and digital capabilities in production planning and control in the Industry 4.0 context. They argue that PPC is influenced by smart capabilities from five different base technologies: Internet of Things, Cyber-Physical Systems, Big Data and Analytics/Artificial Intelligence, Cloud Manufacturing and Additive Manufacturing. For this thesis, the fourth technology, Analytics/Artificial Intelligence, is naturally the interesting one.

Within demand forecasting, the leading smart capability is Internet of Things, but the forecast processing and data analytics are strongly focussed on Big Data and Analytics with Artificial Intelligence (Bueno et al., 2020). Big Data and Analytics with Artificial Intelligence are used to improve predictability of resources, improve the accuracy and performance of the forecasts, do analytics with machine learning methods, and monitor and diagnose with data analytics tools to mention some.

Interestingly, Bueno et al. (2020) found that within Production Scheduling and Shop Floor Control all the five base technologies the study considered, mentioned above, are explored in greater detail.

Zhang et al. (2017) presents the use of a dynamic game theory model to reduce the complexity of multi-objective optimisation for a production scheduling problem. This way they managed to reduce the makespan, the total workload of machines and the energy consumption with 4.5%, 8.75% and 9.3% respectively.

Goodall et al. (2019) proposes use of a simulation of the manufacturing environment that is automatically updated using data from the digital manufacturing systems to reflect the real world. This simulation is proposed used to support remanufacturing operations. By the use of data from the digital manufacturing systems, predictions are made on material flow behaviour which is utilised by the simulation. This enables the simulation to adapt and be maintained simply by feeding it with data from the real world without the need of hard-coded logic being written and maintained manually.

Bueno et al. (2020) argues that the core smart capabilities based on Big Data and Analytics with Artificial Intelligence real-time manufacturing execution systems, real-time object traceability for adaptive and optimized production scheduling, smart scheduling based on big data, analysis and optimisation, predictability, event-driven scheduling, and automated data analytics enabling self-learning by artificial intelligence. Furthermore, they argue that these technologies make new organisational philosophies possible within production planning and control. They highlight how it enables anarchic manufacturing, meaning very distributed PPC systems with the ability to adapt and self-optimise.

4 Case study

4.1 Background from specialisation project

In the specialisation project, I identified two promising use-cases of artificial intelligence within the production planning at Overhalla Betongbygg. The first was the process of dividing a project, or a building, into concrete elements in the best possible way. The second was production scheduling. Both processes are currently done by humans which have a lot to consider when making decisions. This is one important factor to why I considered artificial intelligence to be promising in aiding in these processes.

4.2 Overview of Overhalla Betongbygg

Overhalla Betongbygg is a manufacturer of concrete elements for use in construction. They will typically receive orders for entire buildings. The buildings are then divided into concrete elements at Overhalla Betongbygg. The concrete elements produced includes pieces of foundation, walls, roof, columns, and beams. In addition, they produce elements specifically designed for agriculture.

Overhalla Betongbygg was founded in 1946 under the name Overhalla Cementvarefabrikk. For more than 70 years they have supplied the Norwegian market with concrete products. They experienced an especially large growth during the 1990s. In this decade, they went from being a relatively local supplier to being among the large Norwegian actors in the concrete element industry through development and expansion. From a turnover of 1 million NOK in 1990, they have gown to have a turnover of 350 million NOK as of 2015. A important reason for this success has been their adaptability according to their website (Betongbygg). According to (proff.no), their turnover as of 2019 was just over 361 million NOK.

In June of 2013, they partnered with three other businesses and created Overhalla Gruppen AS (Betongbygg). Besides Overhalla Betongbygg, the business included in Overhalla Gruppen is Overhalla Mekaniske, Overhalla Hus and Overhalla Transport. Together, Overhalla Gruppen are suppliers of elements in concrete, wood, and steel.

4.3 Supply chain management at Overhalla Betongbygg

Overhalla Betongbygg has been very successful in their implementation of lean. In 2019, they won the award for the best lean business in Norway that year.

In their supply chain management, they use an arrow, as shown in Figure 4, to visualise the workflow all the way from when a contract for a project is signed until assembly of elements at the construction site. This arrow shows how many days each process along the way is expected to take. From this, a series of deadlines are calculated.

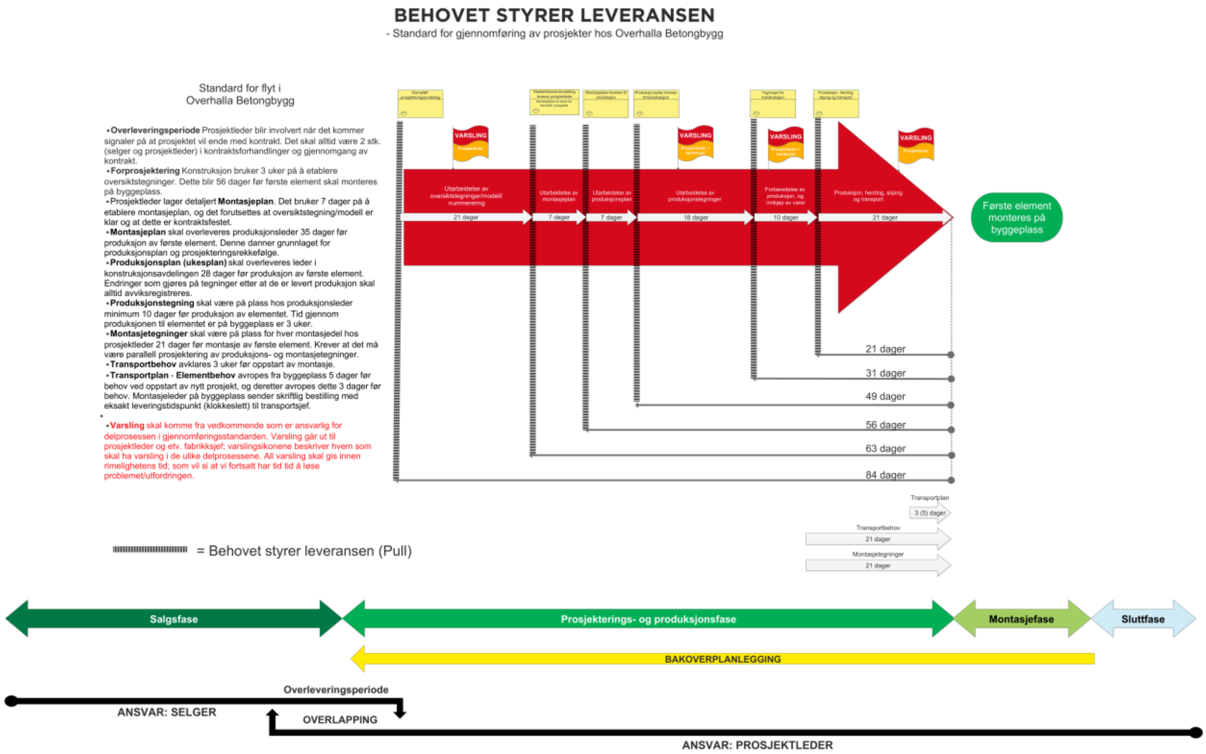

Figure 4: The arrow visualising the workflow at Overhalla Betongbygg.

4.4 Production planning practices at Overhalla Betongbygg

The aim of the production planning at Overhalla Betongbygg is ensuring that projects are executed as efficient in regard to time and resources as possible. With their production planning they attempt to find a way to use their available resources to make this happen.

At Overhalla Betongbygg, a project typically begins with an external architect attempting to draw up a building to meet the requirements of the customer. The drawing is then sent to the constructors at Overhalla Betongbygg and received as a PDF document. The first thing the constructors need to do after receiving the drawing is to go over it thoroughly and make sure constructing the construction drawn up is doable and that it will not collapse. If there are any issues in these regards, the constructors send the drawings back to the architect with feedback on the issues so that the architect can work up a new revision. This is done back and forth between the architect and the constructors at Overhalla Betongbygg until the constructors find the drawings acceptable.

Before production can begin at Overhalla Betongbygg's facility, there is significant amounts of planning to be done. The first step will be for the constructors to attempt to divide the building into concrete elements. From the drawings received from the architect, they create a bird's-eye view drawing with the construction divided into elements. This drawing is then imported into LOS, the software used for a lot of the production planning at Overhalla Betongbygg. Furthermore, as this bird's-eye view drawing as a basis, the constructors will create a list of elements in LOS.

After the building has been divided into elements, the project is handed over to the project leader. The project leader will then assign a date of assembly for each individual element. This is done in LOS. The date of assembly is the day the element is to be assembled on the construction site. The plan created here is referred to as the assembly plan. The assembly plan is an important basis to be used in following processes within production planning at Overhalla Betongbygg.

With the assembly plan for the project completed, the project is sent over to the production planners. The production planners at Overhalla Betongbygg then needs create a production plan. In the production plan, each individual element is assigned a production date. This is done by dragging the individual elements onto workdays in LOS.

Thereafter, the project is handed back to the constructors at Overhalla Betongbygg. Their job is now to create detailed drawings of each element to be used in production.

These detailed drawing are then used by the people at Betongbygg in charge of procurement as theses drawing are used to determine what needs to be ordered. This includes keeping a stock of ingredients for creating concrete as well as metal rods for the machine they have at their facility which bends metal rods to create armouring for the most part. In addition, more complicated metal components are often required. These are ordered from Overhalla Mekaniske, also located at Skogmo, in the industry park.

The detailed drawings from the constructors are also sent back to the production planners. The next step for them is to delegate the production of each element to workers and areas of the production facility and make sure the restrictions are complied with in these regards.

Lastly, the production planners create a transportation plan. That is, a plan for when and with which vehicle each element should be transported to the construction site.

Figure 5 shows a visualisation of this workflow.

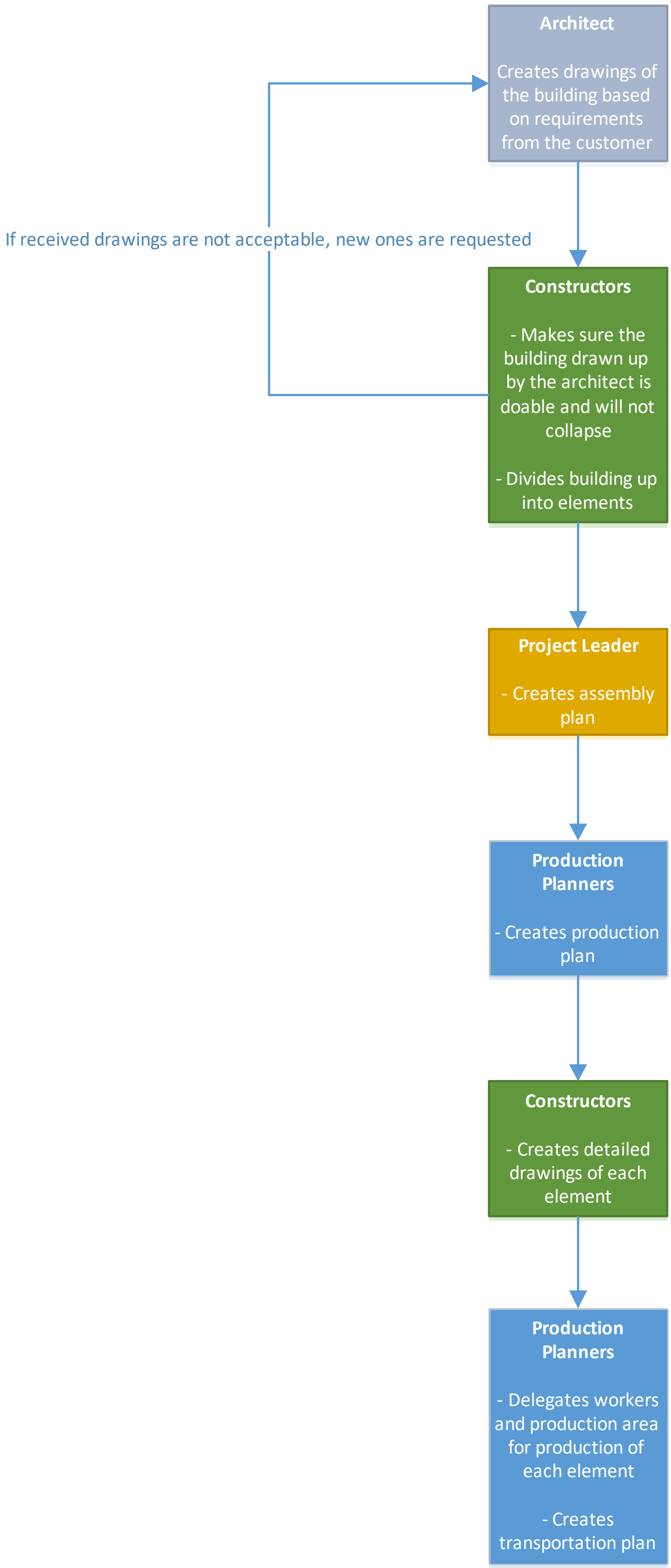

Figure 5: Workflow in production planning at Overhalla Betongbygg

4.5 Challenges within production planning at Overhalla Betongbygg

Several of the processes done in the production planning at Overhalla Betongbygg comes with significant challenges.

When the constructors divide buildings into concrete elements, there are a lot of considerations to be taken when trying to find the optimal solution to the problem. Especially how efficiently the elements can be produced and transported is of high importance. There are also limitations regarding production and transportation capacity to be considered. The constructors do spend some time trying to make sure the buildings are divided into elements in a way that ensures production and transportation of these elements can be conducted relatively efficiently. However, with these considerations being complex, taking all considerations into account to find an absolute optimal solution would be an impossible job for a human to complete in a reasonable amount of time. Therefore, the decisions made are often largely based on experience.

Creating a production plan is also a complex problem with many considerations to be taken. In addition to considering the dates set for assembly, the planners need to make sure all the workers at the factory have a somewhat equal amount of work each day. They also need to consider what elements can be most efficiently produced on what day. For example, if two identical elements are to be produced, it is often best to produce these on consecutive days so that the frame used for casting can be reused and therefore reducing the work to be done for the carpenters.

Additionally, there are several constraints to production that must be overheld. There are several different types of concrete elements that are being produced at Overhalla Betongbygg with different requirements to production resources. All areas of production will have its limitations that must be considered. For example, some roof elements are produced using dedicated, stationary frames as these will have a common shape. The production plan must make sure the number of such roof elements to be produced on a day never exceeds the number of these frames they have available at their facility.

Also creating a plan for transportation comes with its complex limitations need to be considered. Besides making sure each element arrives at the construction site in time for assembly, the production planners need to take into account the limitations of the available transportation means to attempt to figure out which elements can most effectively be transported together. Therefore, the dimensions and weight of the elements must be considered as well as the limitations of these on the transportation vehicle. Often, assembly will happen directly from the transportation vehicle, meaning the vehicle must remain at the production site until assembly of the elements it carried has been completed. The assembly plan is used as a base when trying to solve the problem of transportation, but the order of elements to be transported will often differ from the order of elements to be assembled to optimise transportation loads.

As the creation of both production plan and transportation plan should preferably be considered when dividing building projects into concrete elements, the complexity, and challenges of creating these two plans would be transferred to the process of dividing buildings into elements.

As mentioned, the dates set for assembly for the individual elements are important to consider when creating plans for production and transportation. Therefore, considering production and transportation is important also when creating the assembly plan.

Lastly, transportation will be dependent on the production dates of elements. Therefore, also the processes of creating a production plan and creating a transportation plan is connected in this way and considering transportation when creating a production plan can also be beneficial.

Figure 6 illustrates the dependencies between production planning processes at Overhalla Betongbygg.

4.6 Opportunities for artificial intelligence

In the previous section I described the challenges Overhalla Betongbygg faces within their production planning. In this section I will attempt to connect these challenges to usage of artificial intelligence. I will discuss whether the challenges are fitting for application of artificial intelligence and how artificial intelligence might aid.

Common for the challenges presented in the previous section is their complexity. There are more traditional mathematical approaches, like multi objective optimisation, to solving such problems, but these do come with some limitations (Ngatchou et al., 2005). Artificial intelligence does handle complex scenarios well and is often the preferred solution for complex problems within production planning as stated by (Mula et al., 2006).

Additionally, AI provides good handling of uncertainty. Since the manufacturing at Overhalla Betongbygg is engineer-to-order manufacturing, what products are needed to be produced are known at the time of planning production as they will be based on orders from customers. However, since many projects are large and production for each project might be done over a quite long period of time, production of projects will overlap. Therefore, when scheduling production for a project, other projects that have already been scheduled to have production in upcoming days will affect how the production of the project at hand will be scheduled. Therefore, ideally a project should have its production scheduled in a way that besides making the production of its own elements as efficient as possible, also makes scheduling the production of upcoming projects in a good way as easy as possible. However, when scheduling production for a project, upcoming projects will not be known. There are methods within artificial intelligence that have the ability to learn to take such uncertainties into account. From learning using historical data, an artificial intelligence model can learn to take actions that will make taking good actions for the tasks that are likely to come next easier.

4.7 Discussion

The most promising applications of artificial intelligence within operations management at Overhalla Betongbygg were found to be within production planning. Based on my previous knowledge of artificial intelligence and applications of artificial intelligence I have found through literature review, both in this research project and in my specialisation project, there seemed to be multiple processes within the production planning where artificial intelligence seemed to be able to potentially provide significant benefits if used as a tool. In this master project, I have aimed to develop such a tool that can be of aid in the production planning at Overhalla Betongbygg.

The different processes within production planning are quiet dependant of each other. With this I mean that the way one process is executed will affect how other processes should be done to achieve the best result. For this reason, creating a tool to suggest suitable actions for more than one process would most likely give synergy. If the tool is able to consider how an action in the current process would affect the following processes and take this into account, it might be very beneficial.

However, due to time restrictions, I was not able to extend the scope of my project past a single process within production planning. Researching the benefits from doing this could be an interesting continuation of this research project.

The choice for the process to attempt to apply artificial intelligence in fell on the production scheduling process. The reason for this choice was mainly the fact that from my conversations with representatives of Overhalla Betongbygg, I was given the impression that this was the process where the most historical data was available, providing a good starting point for developing and training a solution with artificial intelligence.

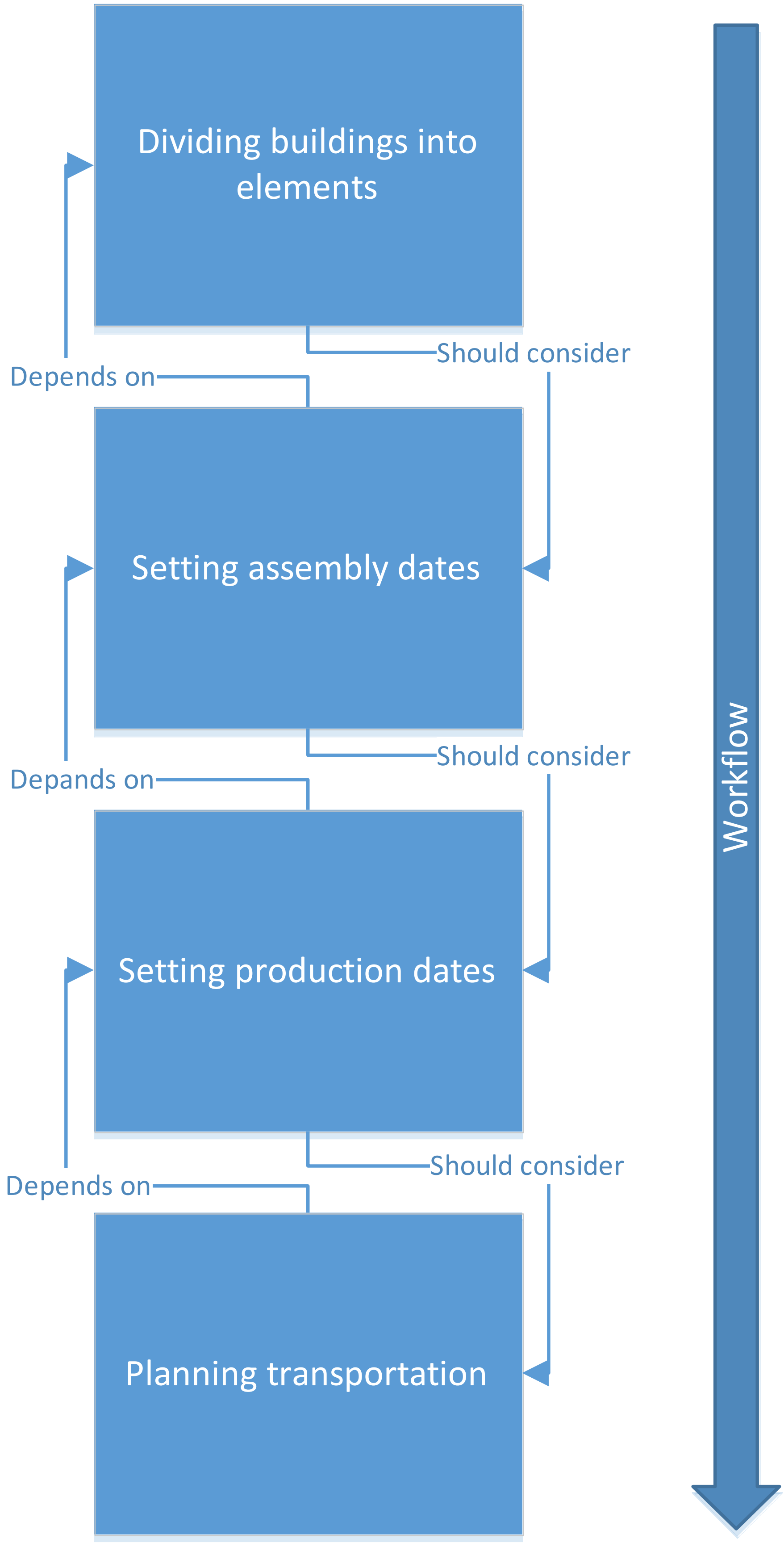

Figure 6: Dependencies between processes in production planning at Overhalla Betongbygg

5 Development

5.1 Data collection

With assistance from Overhalla Betongbygg, I was able to fetch data on several thousands of concrete elements from their database into an Excel spreadsheet. I could then read the data present in this spreadsheet from the code I wrote.

5.2 Artificial intelligence method selection

The initial plan was to use the historical data from Overhalla Betongbygg to look at how production dates has been set in the past as well whether the set production date was postponed or not to learn a model to set an appropriate production date given the concrete element and possibly other relevant input data. However, some limitations in the data provided from Betongbygg provided some challenges. Whenever a production date is postponed at Overhalla Betongbygg, no records of the original production date is kept, so there is no way to know which elements have had their production dates postponed just from the provided data. The data only tells you what the final actual production date was.

With no way of knowing whether a production date was postponed or not, I had to rethink how I could train my model. I attempted to simply use the final production date as the label when training an artificial neural network with the input data as one should be able to consider the final set production date as a suitable production date to set in the first place in most cases. However, this might not always hold true.

In my attempts to trains such an ANN model, I struggled to get a model to give any significant performance increase over simply setting the production date a constant number of days prior to the assembly date of the element. In addition, there are many things to consider when determining how good a suggested production date is. It is not only how close it is to the actual production date. The production date of other elements is also very important to consider for this for two reasons. Firstly, the amount of production to be carried out each day should preferably be rather equal to give the workers a steady amount of work each day and to make sure the amount of work on a day is never more than they have capacity to complete. Secondly, if two identical elements are to be produced on consecutive days, the frames used for casting can be reused which is beneficial.

These factors proved difficult to consider with a custom loss function. Therefore, I started looking at whether reinforcement learning could be a more suitable solution for the problem at hand. Reinforcement learning provides more flexibility in terms of providing feedback on how suitable a suggested production date is. With reinforcement learning, no label is needed. Instead, the model receives feedback in the form of a reward score which can be calculated however the developer find suitable. This allowed me to make the model take into account things that I was not able to make the ANN model do. I could have included the label used in the ANN, the production date set in reality, when calculating a reward for a suggested production date on an element, but since this production date might be suboptimal in the first place, I decided to instead calculate the reward based on other measures developed in cooperation with Overhalla Betongbygg.

5.3 Reinforcement learning tools

The tool I developed was written in the Python programming language (Van Rossum and Drake, 2009). Python was a natural choice for the task at hand as it is widely used for artificial intelligence and machine learning. I addition, Python is the programming language I personally have experience with for programming artificial intelligence applications.

My previous experience with development with artificial intelligence included several different methods, including machine learning methods, but for the most part methods within supervised learning. I did not have any prior experience with developing anything using reinforcement learning. I did however have prior theoretical knowledge of reinforcement learning which helped me consider it as an approach for the problem I attempted to solve in this project.

In my research on how to best develop a reinforcement learning application in Python, I found Ray RLlib, an open source library for reinforcement learning (Liang et al., 2018). I found Ray RLlib suitable as it provided great flexibility with different kinds of environments and importantly implementations of a large collection of reinforcement learning algorithms which in many cases could easily be applied on the same environments without any significant work.

With the use of Ray RLlib, I was able to focus the large majority of my development efforts into the development of the reinforcement learning environment. The agent side of the reinforcement learning solution was in large parts handled by the Ray RLlib library. I had to define the format of the data the agent would be fed with from the environment, the input data described in 5.4.1, or the observation space which it is called in my environment. Furthermore, I had to define the action space, the format of the data output from the agent. Then, after selecting the reinforcement learning algorithm I wanted to apply and setting some optional parameters, everything was set for Ray RLlib to generate an agent that was ready for training and use.

Figure 7 shows an overview of how my finished reinforcement learning solution works. Furthermore, Figure 8 displays more detailed how the trainer samples batches of rewards and corresponding actions and states from the workers to train a model and provide the workers with updated models throughout the training process. The workers use these models to transform states into actions.

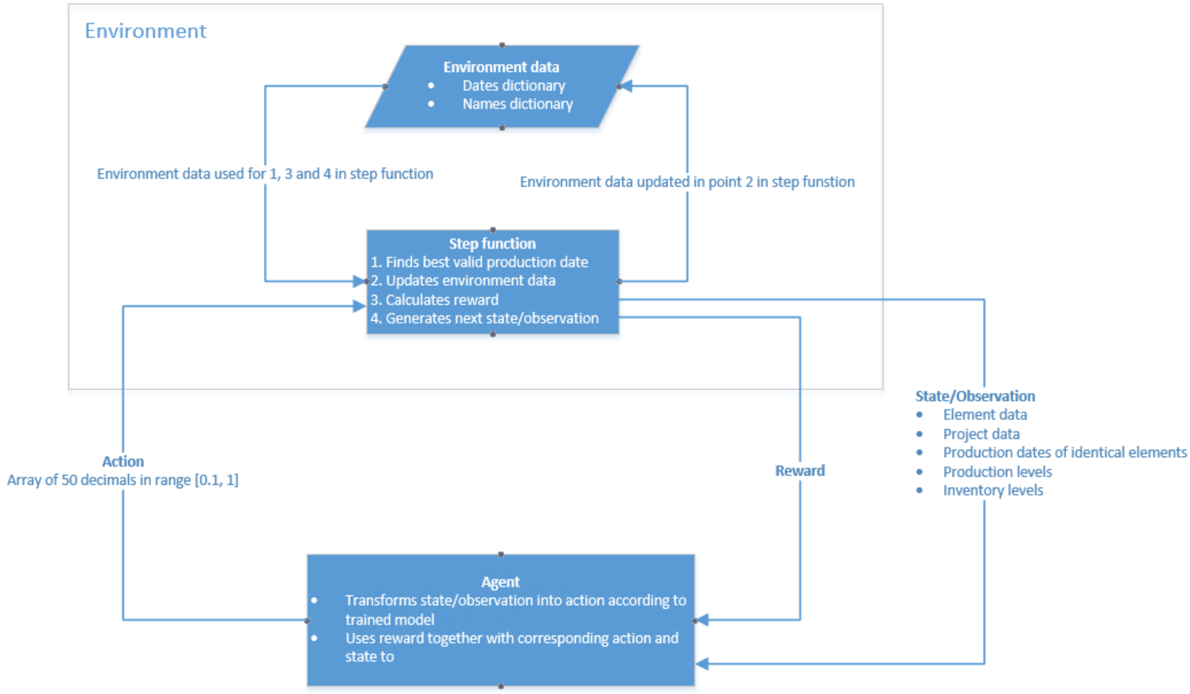

Figure 7: My reinforcement learning solution.

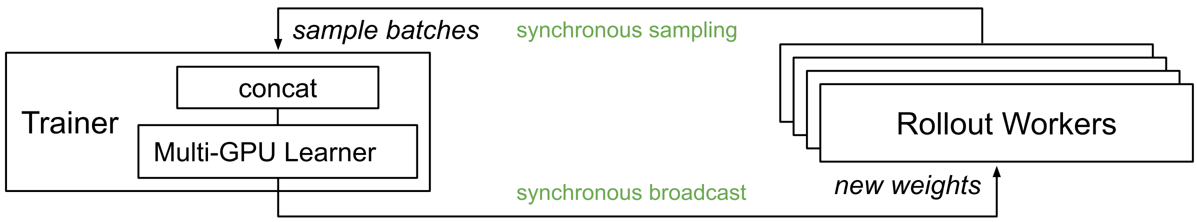

Figure 8: Ray RLlib PPO architecture (Ray, 2021).

5.4 Reinforcement learning environment development

In order to develop a machine learning model capable of suggesting reasonable production dates for the concrete elements at Overhalla Betongbygg, I need to develop a suitable reinforcement learning environment that can be used to train the model.

As described in 3.1.2, the environment in reinforcement learning is responsible for feeding the agent with input data, the state of the environment, as well as giving the agent numerical rewards for its actions. The model to be trained can be said to be part of the agent.

5.4.1 Input data

Selecting appropriate data as input for your model is always important in machine learning.

As the most suitable production date of an element would largely depend on production dates set for previous elements, it was important to include this in the model input in some way.

At the same time, I needed to decide what input data from the current element itself, like the element name and production deadline, to include.

I needed to decide on the data to be included in the state that is sent over to the agent as well as the structure of this, possible actions for the agent to make and how to calculate the reward for the action.

A lot of the data available for each concrete element in the data set I was provided by Overhalla Betongbygg I considered irrelevant for the agent when trying to set an appropriate production date. Data like name of the project the element was a part of, or name of the project leader would ultimately not help the agent make a better decision in this regard, and including data like this in the input data would actually hurt the performance of the agent as it would have a harder time figuring out what data actually matters for the decision making. In the end, the element data included in the input data for each element included the following:

- Element name
- Whether the element belongs to one or more specified categories of elements
- Deadline for production

The data available for each element was a mix of numerical and categorical data. Categorical variables are variables that can be divided into groups (Yale, 1997). The concrete elements in my data set had numerical variables like weight, height and project number which was ultimately not included in the input data. It is however worth mentioning that what project the element was a part of was included by the inclusion of a separate array of project data in the final environment.

Categorical data needs to be handled in a different way than numerical data in preparation for using it in machine learning. Since a machine can only do the mathematical operations, done on the input data of a machine learning model, on numbers, categorical data must be converted to numbers in one way or another. One widely used method to do this conversion is one-hot encoding (Brownlee, 2017). One-hot encoding was used to transform the categorical data into number in this project.

I decided to create an environment that would feed the agent with the relevant data of one element at the time. However, I made the environment append an array of already selected production dates for identical elements as these are of importance when selecting a production date. This early version of my environment did not consider inventory levels in its reward calculation. Neither did it have constraints on how many of certain types of concrete elements could be produced each day.

With further development of the environment, where abovementioned considerations were made, the input data was also extended. If one is to hope for the agent to be able to learn to take these considerations, the agent must be given the information to make it possible to do so. However, since the environment will always be able to filter out production dates that are in violation with any of the constraints defined and to keep the size of the input data reasonable, data that was relevant to the reward calculation was prioritised over data relevant to the constraints.

Therefore, the input data was extended to include production levels and inventory levels for all the possible production dates for each element. Furthermore, since, in reality, production dates will be set for an entire project at the time, data for the entire project the element was a part of was also included in the input data. Including the data for each element in the project in the project data would make the size of the project data way to large. I wanted to keep the proportion of the input data made up by the project data reflecting the importance of the project data in the decision-making of the agent. Therefore, I had to get the size of the project data down a significant amount. This was done by the means of dimensionality reduction. More specifically, I used Principal Component Analysis (PCA) (Wold et al., 1987). Using PCA allowed me to reduce the dimensional complexity of the project data while maximising the data variance between the projects in my data set.

I ended up with my input data having the following structure:

- Element data
    - Array of 785 decimals
- Project data
    - Array of 50 decimals
- Production dates of identical elements
    - Array of 148 decimals
- Production levels
    - Array of 50 decimals
- Inventory levels
    - Array of 50 decimals

5.4.2 Action space

The environment also defines the format of the action to be decided on by the agent. This is defined by an action space.

In my environment, I initially decided on a discrete action space with a size of 100, meaning the agent is to return an integer between 0 and 99 inclusive. The idea was to interpret this value as a percentage of working days to pass from the date of the deadline for the drawing to be completed to the date of assembly before producing. For example, if the date of assembly for an element is 50 days after the date of the deadline for the drawing to be completed, an action of 50 made by the agent would be interpreted as selecting a production date 25 days after the date of the deadline for the drawing to be completed.

From analysing the data provided by Overhalla Betongbygg, I noticed that for some elements, their deadline for drawing were actually only few days prior to their date of assembly. Through further discussions with Overhalla Betongbygg, I was informed that the deadline for drawing is actually being set after the date of production has been set. Therefore, I had to find another approach besides using the drawing deadline.

In cooperation with Overhalla Betongbygg, I decided on giving each element a total of 50 possible production dates. The latest possible production date, the production deadline, would be one week prior to assembly and the earliest production date would then be 49 workdays prior to the production deadline.

The action space of the environment was defined as an array of 50 floating point decimals, one floating point decimal for each of the 50 possible production dates for an element. The floating points decimals were bounded to be between 0.01 and 1.0. A larger value would be interpreted as a more suitable production date. The reason for defining the lower bound of these values to 0.01 rather than 0 was that 0 would be used to mark a production date as invalid as described in 5.4.4.

The idea behind having the model output a decimal for each possible production date rather than simply outputting a single integer from 0 to 49 to be interpreted as the most suitable production date was to have the opportunity to fall back on another production date if the production date considered by the model to be the most suitable turned out to not uphold the constraints of the production. How these constraints are handled in the environment developed is described in more detail in 5.4.4.

5.4.3 Step function

For each step, my reinforcement learning agent would provide an output interpreted as the action of the agent. My reinforcement learning environment would then have to handle this action appropriately and give feedback to the agent on how good the last action was in form of a reward score.

To make this possible, my environment stored a series of data to keep track of the status of the production schedule. To make this data easily accessible when checking whether constraints are upheld, rewards are calculated and input data for the agent is compiled, most of this data was stored in dictionaries. A dictionary consists of a series of key and value pairs. The key can be used to efficiently extract the wanted value from the dictionary.

The dictionary containing the most information is the dates dictionary. In the dates dictionary, every key corresponds to one possible production date, that is workdays. Every workday in the dictionary will then another dictionary with information about this date. This dictionary contains the total number of elements scheduled for production on this workday, the number of elements in inventory on this workday, the average number of elements scheduled for production per day the previous week as well as the number of elements of some special element categories with special considerations to be made on that are scheduled for production this date. The structure of this dictionary is also visualised in Figure 9.

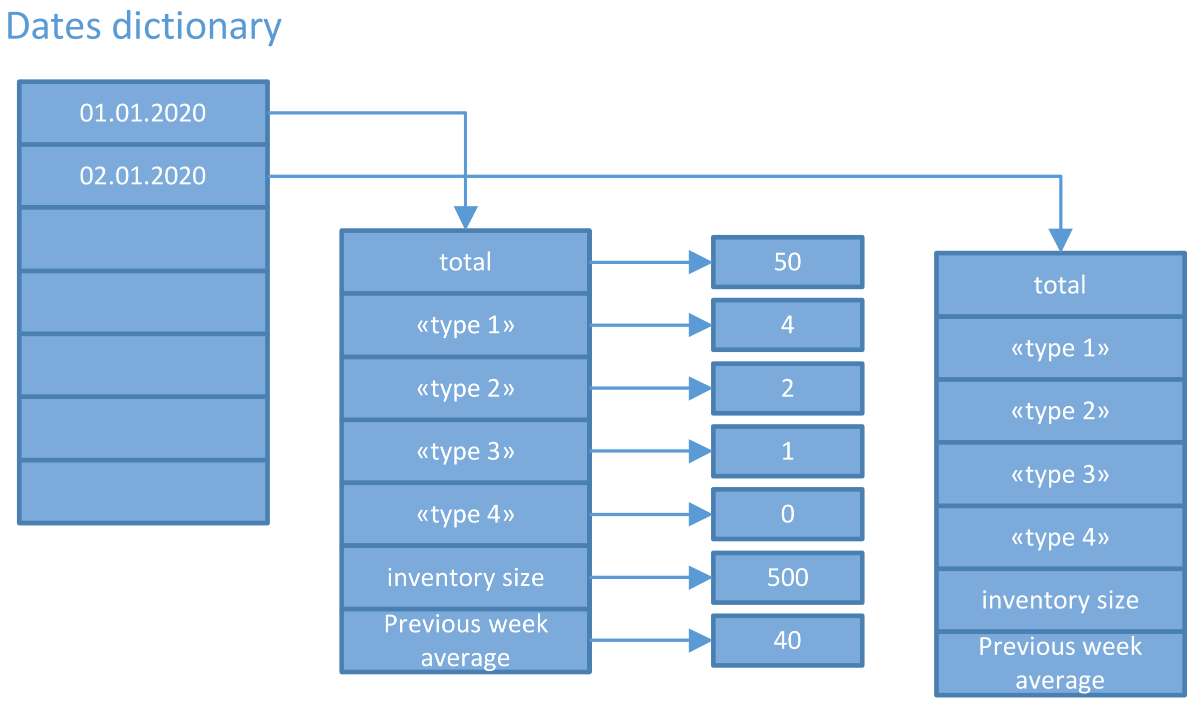

Figure 9: Dates dictionary structure.

Secondly, the environment holds a name dictionary. This dictionary has a key corresponding to every element name of elements that has been scheduled for production. Every key in the dictionary holds an array containing scheduled production dates of elements with element name equal to the key. The structure of this dictionary is visualised in Figure 10.

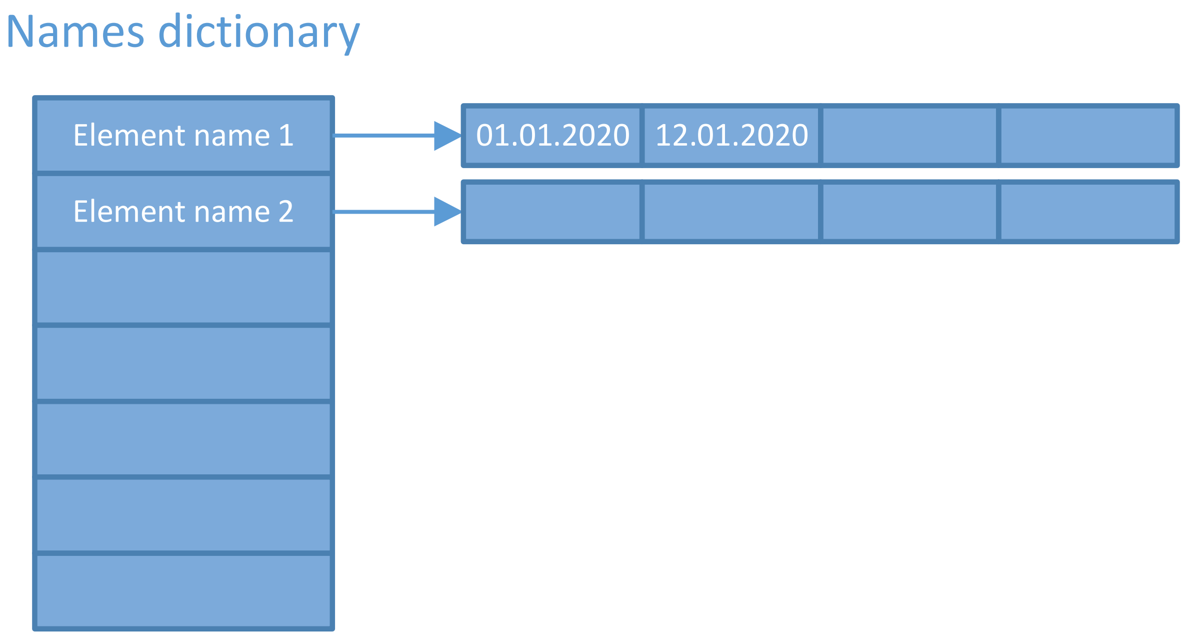

Figure 10: Names dictionary structure.

These dictionaries play a vital role in handling actions from the agent appropriately and efficiently.

When the environment receives an action array from the agent, “Arg max” will be applied to the array. This will return the index in the action array with the highest value. This will be the chosen action.

This single integer chosen as the action would then be interpreted as the production date with 0 being the earliest possible production date for the element and 49 being the latest possible production date.

Before proceeding with this production date, I the environment would control that the constraints are upheld if this production date is set for the concrete element.

5.4.4 Constraints

The constraints that every action would have to upheld were developed in cooperation with Overhalla Betongbygg to make sure the model developed would not suggest production plans that Overhalla Betongbygg would not consider doable.

These constraints are briefly mentioned in 4.5. In this section, I will describe how my environment has been programmed to handle these constraints.

As described in 5.4.2, the output of the model that is received by the environment is in the form of an array of decimals. After applying argmax to get the index in the array with the highest value, the production date that the model considers the most suitable, this production date is taken through some test to make sure no constraints are broken if this production date is used. If the production date breaks any of the constraints, it cannot be used. This is handled by setting the corresponding index of the action array to 0 to mark it as unusable. Then argmax is applied on the array again and will now return a new index, the production date the model considered as the second best option. Like the first production date, this production date will be tested to check if constraints are broken. If it is, this index will also be set to 0 in the array. This process will continue until a production date that does not break any constraints are found. If all the 50 possible production date break one or more constraints, the attempt at creating a production schedule is stopped and a large negative reward is returned to the agent. This process is illustrated in Figure 11. Furthermore, Figure 12 displays pseudocode of how this is done in the solution developed.

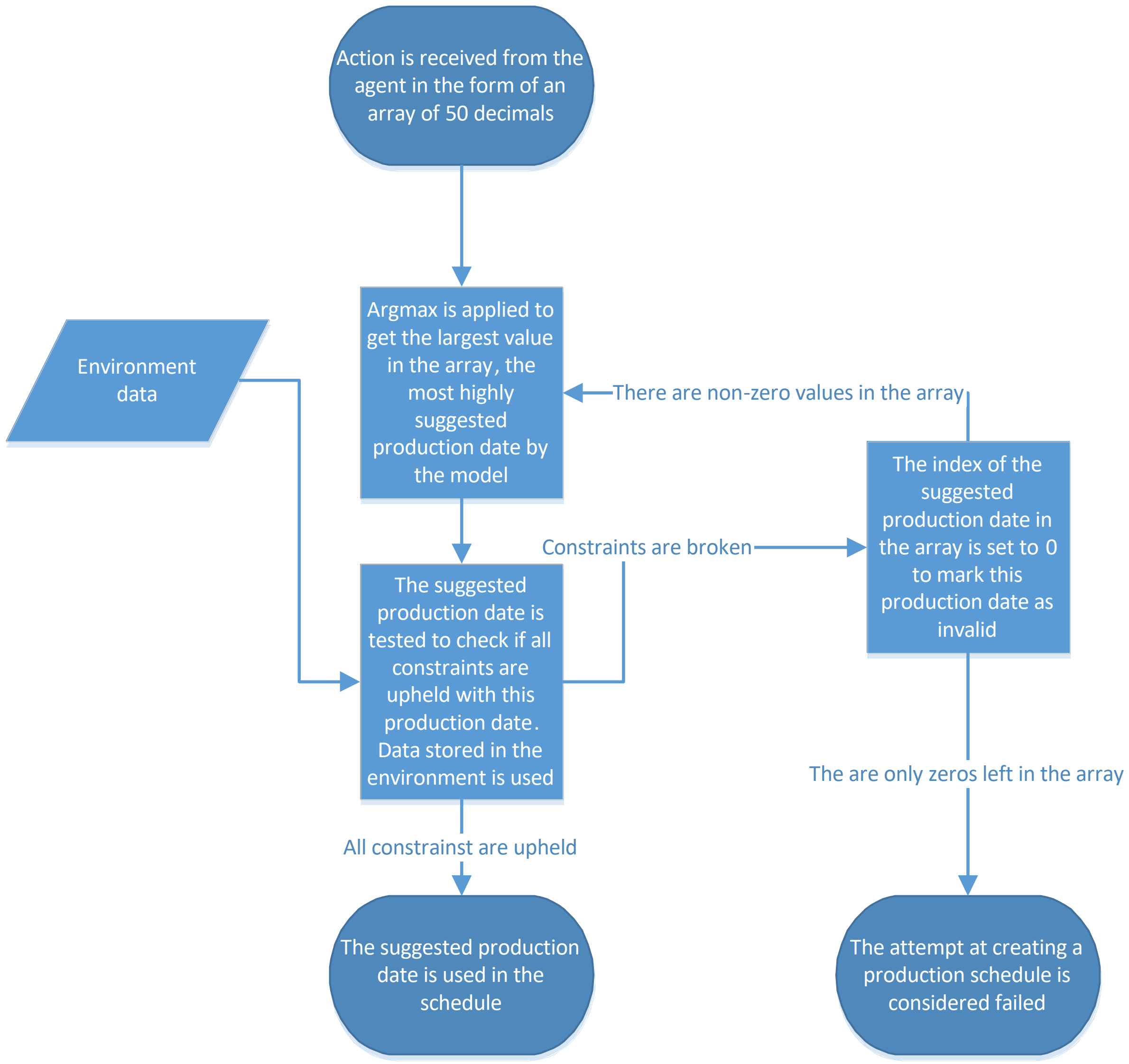

Figure 11: How the environment selects and validates production dates for each element in the developed solution

```
accepted = False
while not accepted:
    action = argmax(action_array)
    accepted = True

    // The count of identical elements to be produced at the same date should
    // never exceed 4 and if there are no more than 10 of this element, they
    // should all be produced on seperate days
    if count_of_current_element <= 10:
        accepted = count_of_same_element_at_same_date == 0
    accepted = accepted and count_of_same_element_at_same_date < 4

    if current_element_is_of_type_1:
        accepted = accepted and count_of_type_1_elements_at_same_date < 10
    if current_element_is_of_type_2:
        accepted = accepted and count_of_type_2_elements_at_same_date < 2
    if current_element_is_of_type_3:
        accepted = accepted and count_of_type_3_elements_at_same_date < 2
    if current_element_is_of_type_4:
        accepted = accepted and count_of_type_4_elements_at_same_date < 2

    for day in days_element_will_be_in_inventory:
        accepted = accepted and inventory_at(day) < inventory_limit

    if not accepted:
        action_array[action] = 0
        if is_all_zeros(action_arr):
            // Attempt at creating production schedule failed
```

Figure 12: Pseudocode describing how production dates are selected and validated for each element.

5.4.5 Reward calculation

After a production date that does not break any constraints has been chosen, the reward to return for this production date will be calculated. In this section I will describe how this reward value is calculated.

Since producing two identical elements on consecutive days will give Overhalla Betongbygg the opportunity to reuse the frame used for casting, which is beneficial, this will be rewarded by the environment.

Firstly, producing identical elements on the same day can be considered a missed opportunity to reuse a frame. Therefore, a production date suggested by the model will receive a negative reward for each identical element already scheduled to be produced on this date.

Secondly, a positive reward will be given for each identical element scheduled to be produced on a date close to the date suggested for the current element. A quite large reward will be given for elements scheduled for production the workday before or the workday after the suggested date, with the reward quickly shrinking with larger gaps between the production dates as they do not want to be storing many frames for later reuse over longer times at Overhalla Betongbygg.

Since Overhalla Betongbygg wants to have a somewhat equal amount of work to do each day at their production facility, the environment will give a larger reward for creating a production schedule that does just this. This is done using the dictionary keeping track of the average number of elements produced each day the last week for every workday as well as number of elements scheduled for production at each workday. When a production date is suggested, the number of elements to be produced on this date, with the current element includes, is being compared to the average for the previous week. A negative reward is given according to the percentwise difference between these two values to encourage the model to keep these values close, resulting in rewarding a steady number of elements to be produced each workday.

Lastly, keeping a low inventory level is preferred. There is only space enough for a limited number of elements in the inventory of Overhalla Betongbygg's production facility. In addition, carrying inventory is expensive. Inventory is held up resources. In a business with engineer-to-order manufacturing like Overhalla Betongbygg, there is no real advantage of keeping products in inventory as these are never used to fulfil upcoming orders like in make-to-stock manufacturing. Rather, the products are produced to help fulfil one specific project order and in the case of Overhalla Betongbygg, in time for the date set for the element to be assembled at the construction site. Therefore, for Overhalla Betongbygg, producing the elements just in time for it to be transported to the construction site in time for assembly could be seen as the most cost-efficient approach. However, limitations in production capacity will force them to produce many elements earlier than this to be stored in inventory to closer to the assembly date. Still, we want our machine learning model to keep inventory levels low.

In an attempt to encourage the model to keep inventory levels low, inventory levels will also have an affect one the reward returned for a suggested production date.

Firstly, a reward, which will be larger the fewer days before the production deadline the suggested production date is, is given since this will lead to the element being stored in the finished goods inventory for a shorter time.

Secondly, a reward, which will be larger the smaller the summed inventory over all the days the element is planned to be stored in finished goods inventory is, is given. This is included to prioritise keeping finished goods inventory low in periods with already many elements in this inventory.

How the reward is calculated in the developed solution is shown in Figure 13.

```
reward = 0

// Gives negative reward for producing multiple of the same element at the same day
reward -= count_of_same_element_at_same_date

// Rewards for keeping production dates of identical elements close
for element in same_elements_produced_same_date:
    reward += squared(1 / days_between_the_production_dates)

// Subtracts from the reward according to the percentwise different between
// elements to be produced at the suggested date and the previous week average
reward -= abs((count_of_elements_to_be_produced_at_suggested_date - week_average) /
    max(count_of_elements_to_be_produced_at_suggested_date, week_average))

// Rewards for not having to keep the element in inventory for long
reward += 100 / days_before_production_deadline_production_is_suggested

reward += 10000 / inventory_level_summed_over_all_days_the_element_will_be_in_inventory
```

Figure 13: Pseudocode describing how the reward is calculated for each element.

As seen in Figure 13, in the last version of the developed solution prior to thesis delivery, a lot of weight was put on the reward for keeping inventory levels low. The weights of the rewards were developed and updated throughout the development process using feedback from Overhalla Betongbygg. These weights, and the reward calculation as a whole, might easily be updated further to meet the priorities at Overhalla Betongbygg, which might change over time.

5.4.6 Environment development process

In order to make sure the reinforcement learning environment I developed would have constraints and give rewards in a way suitable for the production at Overhalla Betongbygg, I was dependent on some feedback from someone knowing more about the production at their facility than I did. I was lucky enough to get this. Over multiple iterations, I developed the environment, trained a model with reinforcement learning, made the model generate a production schedule and then request feedback on this schedule while explaining the constraints and reward calculations used in the environment to develop the model and the schedule. I then received feedback on how the production schedule generated by the model could be improved. This feedback was given both in the form of limitations in the production at Overhalla Betongbygg that the production schedule did not consider and perhaps broke, which I could use to define the constraints in the reinforcement learning environment, and in the form of explanation of what is the most important considerations to make to create a schedule considered good at Betongbygg, which I could use to further develop and adjust the reward calculation.

6 Results

In this chapter, the results from the tool development work will be presented. Firstly, the results of the training process will be shown to display the solution's ability to learn to perform better over time of training. Then, a production schedule generated by the trained machine learning model will be compared to the production schedule that was used at Overhalla Betongbygg.

6.1 The training process

As mentioned several times throughout 5.4, my reinforcement learning environment went through multiple versions during the development process. In general, earlier versions of the environment were simpler, with less considerations made, and lower complexity of the data handled.

The perhaps most significant change in the environment came with the inclusion of considerations of inventory levels. This change had big impacts on the data the agent needed to be fed with, constraints, reward calculation and initially, the agent's ability to learn by training. Prior to this change, I was able to get an agent learning to perform better and better on the environment with not a lot of work. Figure 14 shows rewards over time of training prior to considering inventory levels in my environment. Proximal Policy Optimization (PPO) was used as the algorithm to achieve this learning. As the figure displays, the agent was, with the help of PPO, able to learn to interact with the environment in a way that resulted in an increasingly high reward.

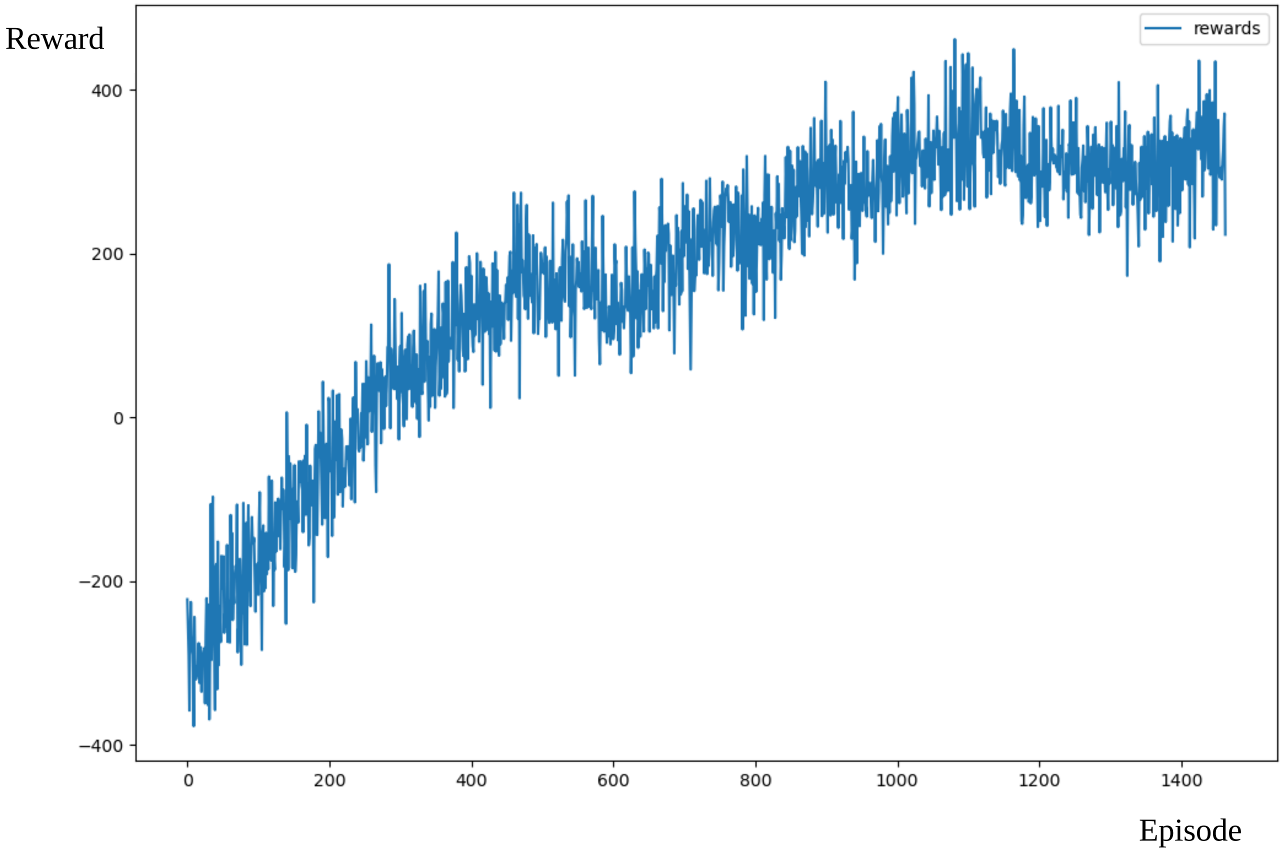

Figure 14: Rewards over time of training with an early version of my environment

However, achieving high rewards from the environment is of low value if the environment itself does not accurately reflect the real-world problem that one is trying to solve. When consulting with the controller at Overhalla Betongbygg, I was informed that although the production dates suggested by the agent did good in considering certain things like making reuse of casting frames possible, it did not comply with all constraints present at their production facility, most notably the constraint on finished goods inventory levels.

Therefore, I developed the environment further, adding a constraint on the level of inventory of finished goods that could be held at the facility according to what I was told was the most inventory of finished items they could hold at their facility. With this implemented, production dates suggested by the agent that would lead to the level of finished goods inventory exceeding this level would not be accepted.

Initially after this addition to the environment, the agent had a more difficult time learning to with the environment. The reward did not increase much even over long periods of training. Clearly some adjustments needed to be made.

A lot of time and work was put into configuring the environment to make learning feasible and finding the right reinforcement learning algorithm to efficiently learn with the environment. Besides levels of finished goods inventory, the environment was over time also extended to include constraints on how many of specific types of elements that could be produced each workday besides the existing constraints on how many of identical elements that could be produced on a given workday. Furthermore, the reward calculation was extended to reward keeping the level of finished goods inventory low.

Table 2 shows different algorithms and configurations attempted applied with the environment.

Algorithm	**Discarded because of**
DDPG (Lillicrap et al., 2015)	Inferior results
A3C (Mnih et al., 2016)	Inferior results
Impala (Espeholt et al., 2018)	Inferior results
SAC (Haarnoja et al., 2018a)	Inferior results
MAML (Finn et al., 2017)	Unable to make it work with the environment
PPO with curiosity (Schulman et al., 2017, Pathak et al., 2017)	Inferior results
PPO (Schulman et al., 2017)	**Not discarded**

Table 2: Algorithms and configurations attempted applied.

In attempts to keep the finished goods inventory low, the agent would over time learn to suggest production dates relatively close to the production deadline. This would lead to higher rewards for keeping the finished goods inventory low, but if the agent were to suggest too late production dates, it would often have a difficult time finding possible production dates for later concrete elements that did not break any constraints on the production capacity. When the production dates were set in a way that left no acceptable production dates for any upcoming concrete elements, the entire attempt at creating a production schedule would be considered failed and a large negative reward would be returned to the agent. Therefore, the agent would need to learn to balance the thin line between low levels of finished goods inventory and setting production dates too late to leave acceptable production dates for upcoming elements. This is clearly visible when looking at the reward returned over time of training with such an environment. This is shown in Figure 15.

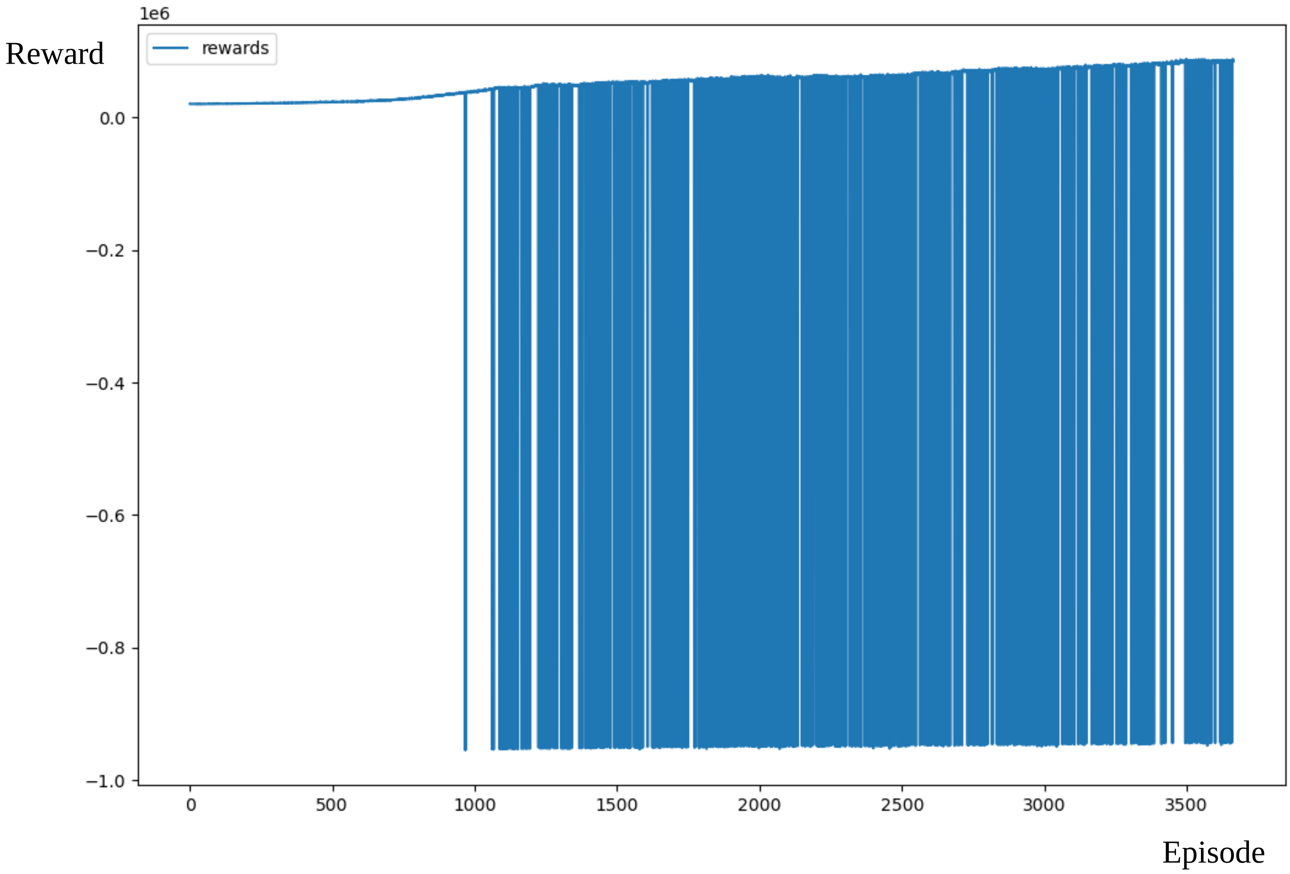

Figure 15: Rewards over time of training trying to keep finished goods inventory low.

In Figure 15, every time the returned reward fell way into the negatives, it was because of a failed attempt at setting a production schedule. We can clearly see how there was no failed attempts early on in the training process until around 900 episodes where we can see the reward graph drops into the negatives for the first time. The rewards have been increasing up until this point and can be seen to continue increasing after this first drop. After some more rise in the returned rewards, more failed attempt at setting a production scheduled can be seen and the reward graph can be seen to drop into the negatives more frequently from this point onwards.

However, when setting a production schedule does not fail, we can see that the positive reward keeps increasing. This increase is made more visible in Figure 16 where failed attempts at creating a production schedule are filtered out, leaving only the positive rewards.

Furthermore, we can, by looking at the x-axis, see that even though drops into the negatives seem to appear just as frequently as positive rewards in Figure 15, only about 600 of about 3600 episodes has been filtered out in the process of creating Figure 16, meaning only about 1 in 6 episodes failed in their attempt to create an acceptable production schedule.

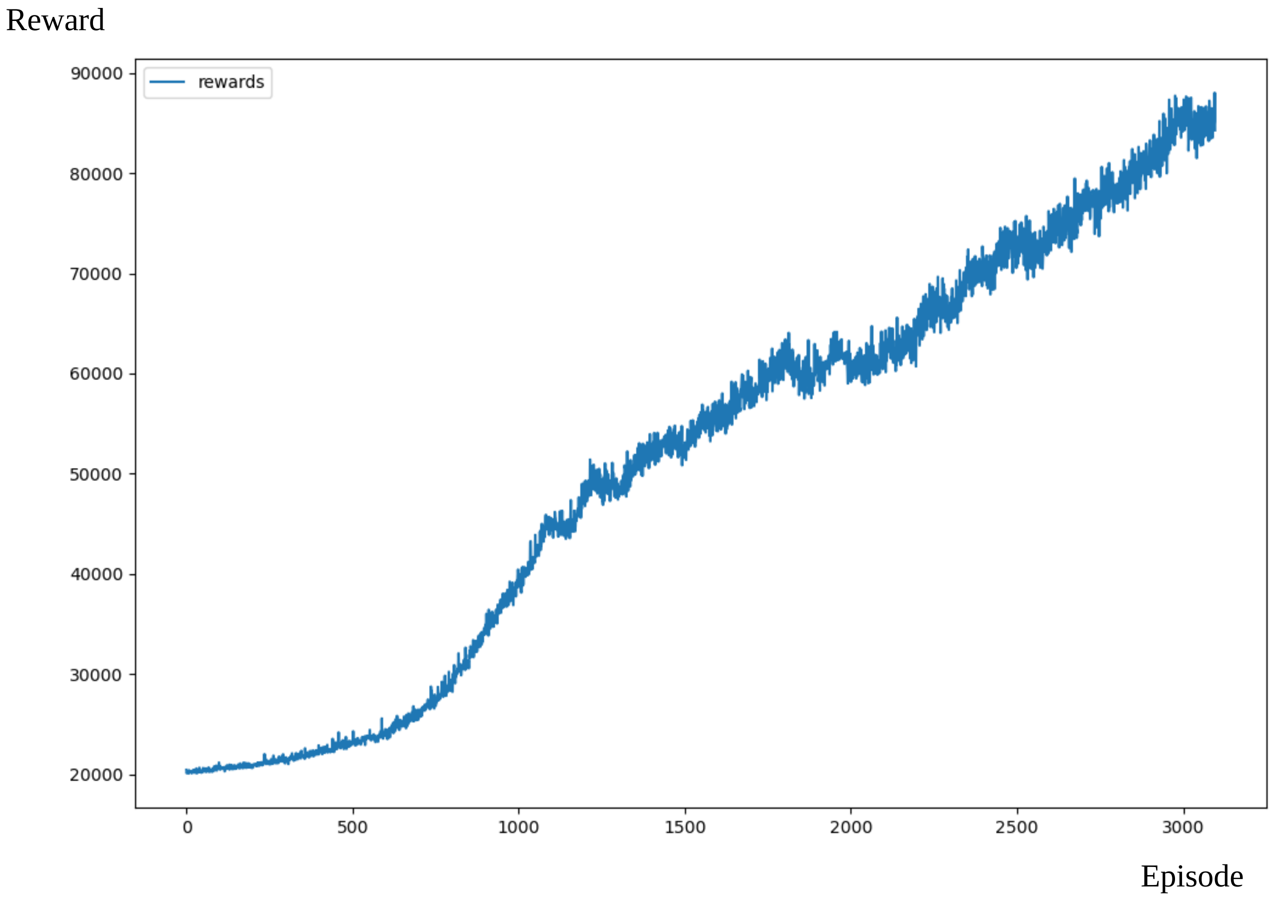

Figure 16: Rewards over time of training with failed attempts filtered out.

6.2 Performance of the trained model

Using the production schedule created using the model that the training shown in Figure 15 and Figure 16 resulted in, we can make comparisons between the performance of the trained model and the production schedule that was used at Overhalla Betongbygg. Figure 17 and

Figure 18 shows how many workdays prior to the assembly date the production was in the schedule created by the machine learning model and the schedule used at Overhalla Betongbygg, respectively.

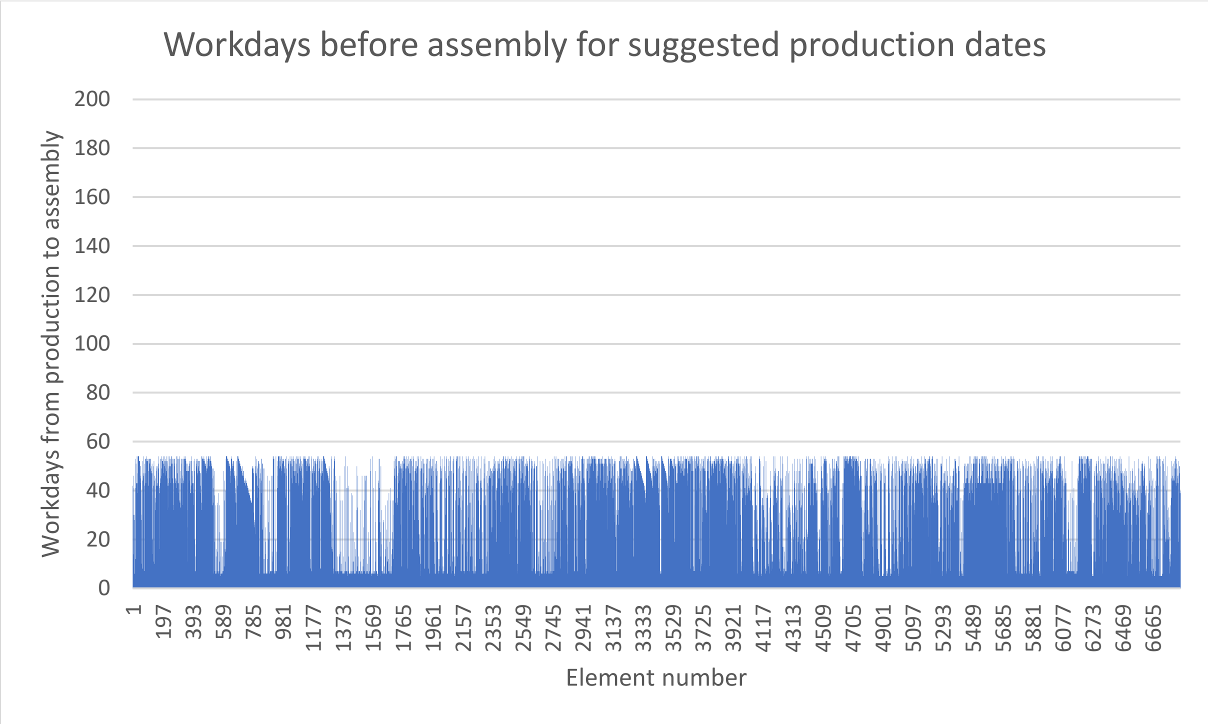

Figure 17: Workdays before assembly for production dates suggested by the trained model.

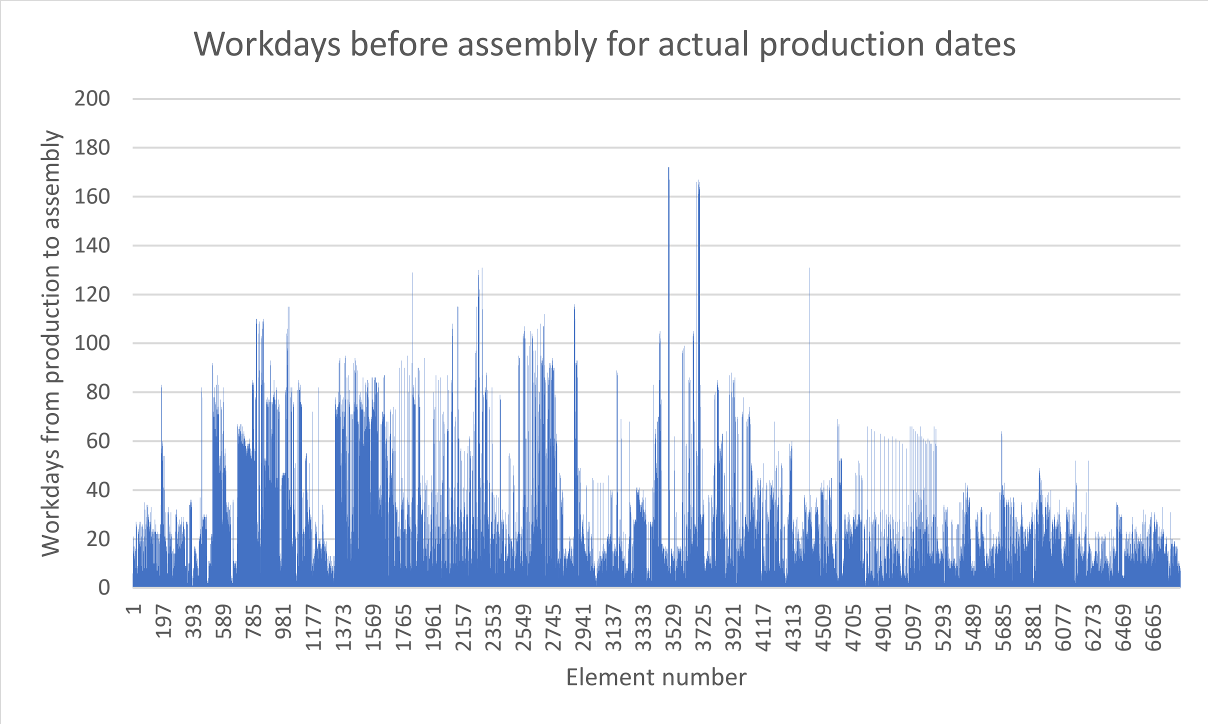

Figure 18: Workdays before assembly for the actual production dates at Overhalla Betongbygg.

A comparison between the use of overtime with the production scheduled used at Overhalla Betongbygg compared to the one created by the machine learning model is displayed in Figure 19 and Figure 20. The amount of overtime for the used schedule is based on overtime reported by the production at Overhalla Betongbygg. Meanwhile, the overtime displayed for the schedule created by the model is based on the assumption that when the constraints on the production capacity defined in the reinforcement learning environment is upheld, no overtime will be needed. With this assumption, the overtime of any production schedule created by using the reinforcement learning environment without failing will be zero.

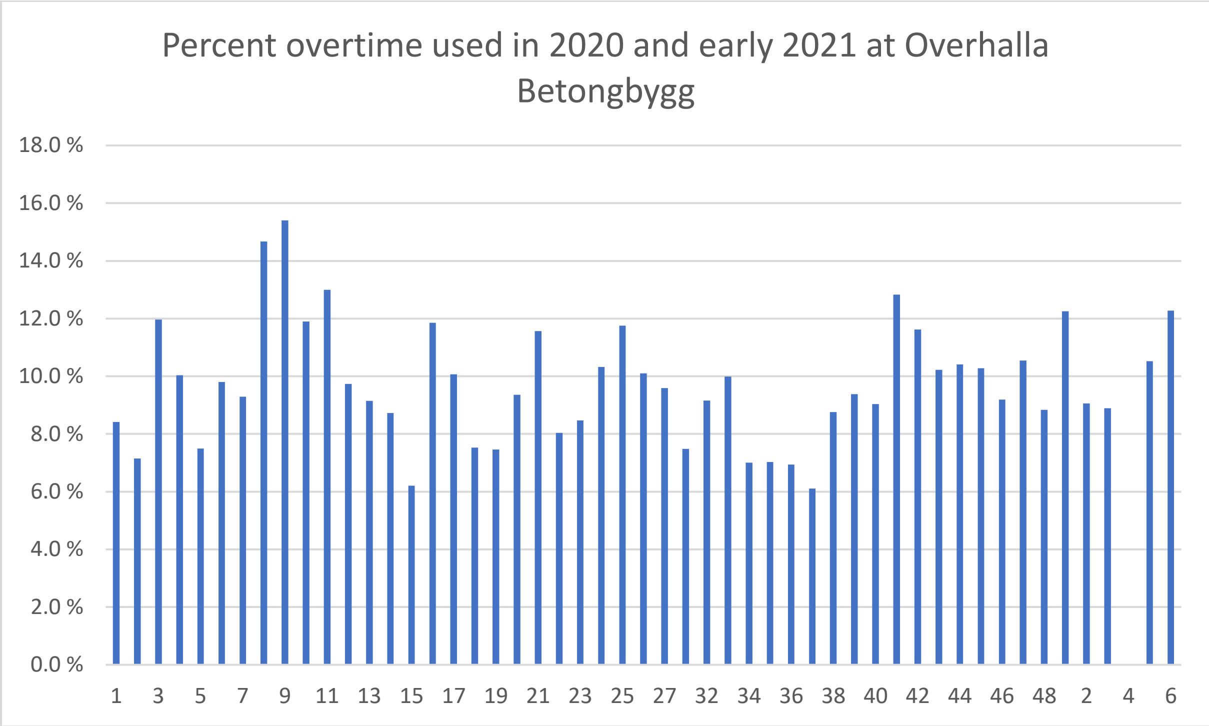

Figure 19: Overtime used at Overhalla Betongbygg in 2020 and early 2021 as a percent of regular worktime.

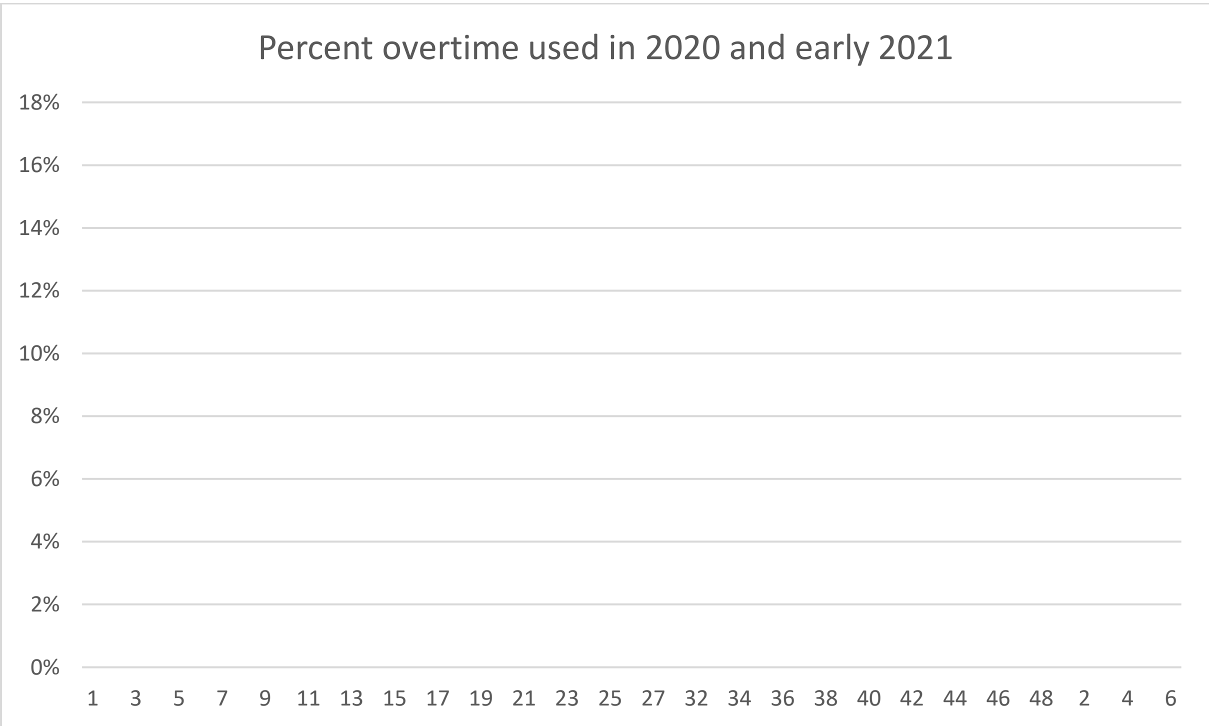

Figure 20: Overtime used with the production schedule created by the machine learning model.

Figure 21 and Figure 22 shows the balance of the workload over workdays for the production schedule used at Overhalla Betongbygg and the production schedule created by the trained machine learning model, respectively. Having a somewhat equal amount of work to do each day is one of the goals of the production planning at Overhalla Betongbygg. The standard deviation of the number of concrete elements produced at the different workdays was **7.873** for the schedule used at Overhalla. For the production schedule created by the trained model, the standard deviation was **9.225**.

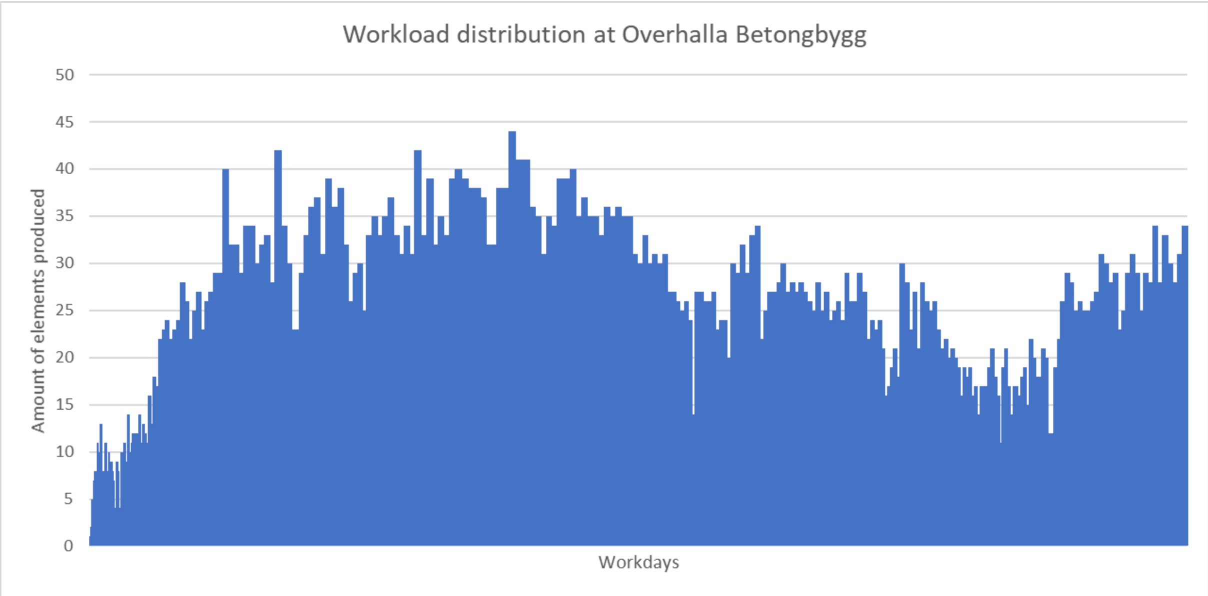

Figure 21: Workload distribution over workdays at Overhalla Betongbygg.

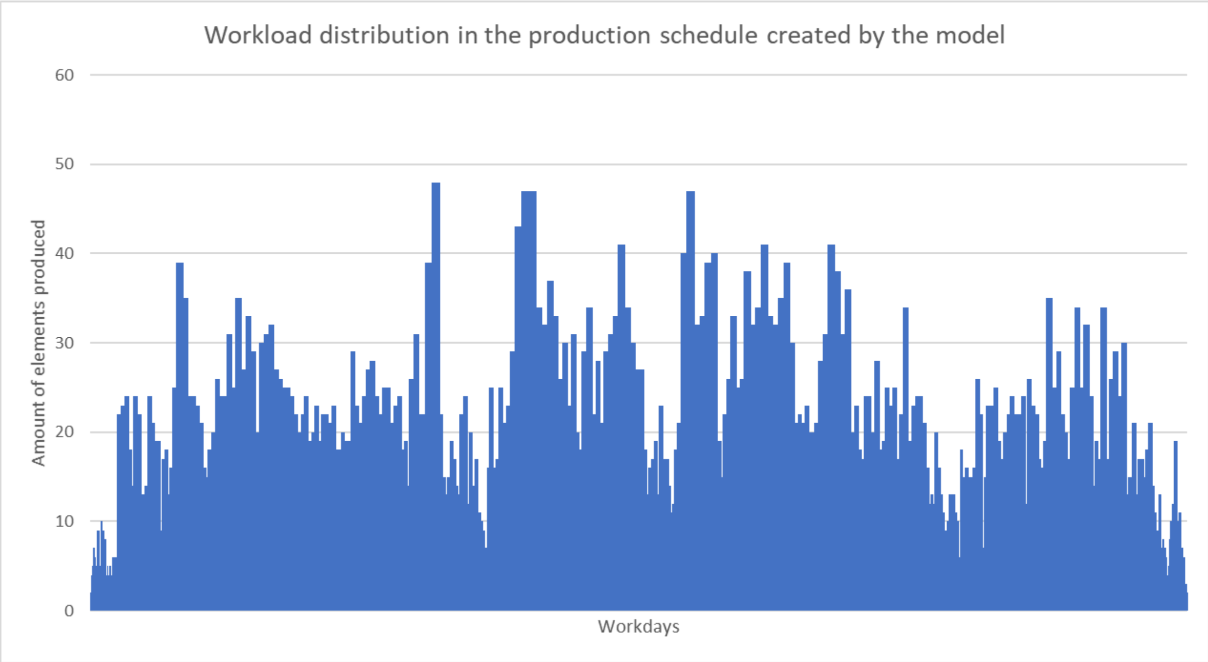

Figure 22: Workload distribution over workdays with the production schedule created by the trained machine learning model.

A summary of the comparisons between the production schedule that was used at Overhalla Betongbygg, and the production schedule created by the machine learning model is provided in Table 3.

	Actual schedule	**Model's schedule**
Reward from environment	~ -1.000.000	**~ 85.000**
Average workdays from production to assembly	**26.1478**	32.53162
Average percent overtime	9.55%	**0%**
Workload balance standard deviation	**7.873**	9.225

Table 3: Comparisons between actual production schedule and schedule created by model summary.

7 Discussion

7.1 Results Interpretations

From the results displayed in Figure 17, we can clearly see the limit of maximum 54 workdays between production and assembly, 49 between the earliest production date and the production deadline plus 5 workdays from the deadline to the assembly, being upheld in the schedule created by the model. At the same time, we can see that production dates are often set close to 54 workdays prior to the assembly date. There could be several different reasons for this.

Since the reward calculation used heavily rewards keeping inventory levels low, we would expect the model to favour production dates closer to the assembly date if it found a way to do that. Therefore, a probable reason for the early production dates in this production schedule is that constraints on production capacity forces many elements to be produced relatively early. The fact that the agent in the later stages of the training process quiet often failed at creating a valid production schedule, makes this reason more probable as it suggests that the production schedules being created by the model are near the limit of breaking constraints on production capacity.

Another possibility is that the agent simply has not been training for long enough to learn to set production dates closer to the assembly date. However, since setting production schedules relatively often failed during the later stages of the training, there seems to be a clear obstacle in the way of simply setting all production dates later due to the constraints on production capacity. This does however not necessarily mean that production schedules with later production dates are impossible. As we can see from the rewards given over time of training in Figure 15 and Figure 16, the rewards were still on an upwards trend at the time of ending the training. This suggests that with even more time with training, the rewards returned could get even larger, most probably as a result of the agent learning how to set production dates later without breaking the constraints on production capacity.

The agent was however in the training process discussed allowed to train over many hours. It is possible that with some tuning of the environment, agent parameters and the use of faster hardware that the agent could learn to achieve even larger rewards in a more reasonable amount of time.

From Figure 18, we see that, in reality, the number of workdays between the production date and the assembly date often exceeded way past the limit of 54 days used in my reinforcement learning environment. At the same time, this production schedule contains more production dates less than 40 workdays prior to the assembly date than the schedule created by the trained model does. This might be a result of some production dates set early creating opportunities for others to be set later. It might also show opportunities for later production dates that the machine learning model were not able to learn to take advantage of. This might be caused by the constraints of the reinforcement learning environment which the machine learning model had to learn to not break while the used schedule was in violation with.

An important consideration to make here is the fact that the production schedule used at Overhalla Betongbygg was in many cases not executed without any issues. An average of 9.55 % overtime was used in 2020 and early 2021. This might suggest that the production schedule that was used was too optimistic with regards to the production capacity at the facility of Overhalla Betongbygg.

Besides having a lower average number of workdays from the production date to the assembly date, the used production schedule came out slightly ahead in the comparison to the schedule created by the machine learning model regarding workload balancing. The used schedule might also here have achieved better number because of the use of overtime and not complying with the constraints of the reinforcement learning environment.

7.2 How the developed tool can be used by Overhalla Betongbygg.

In this thesis, a great emphasis has been placed at the performance of the training process for the reinforcement learning solution. However, when applied in practice at Overhalla Betongbygg's production facility, going through the training process will not be necessary to create suggestions for production schedules. When concluding the training process, the trained model will be saved and is available for later use.

Therefore, to create a production schedule for new concrete elements using the reinforcement learning solution, all that is needed is a trained model to restore. Then this model can be used to suggest production dates the way it learned to do during its training process.

However, it would be wise to use the new data, in the form of new concrete elements to be scheduled for production, to improve the trained model periodically. If the model were to be either retrained with the new data included or trained further with the new data, for example once a week, it will have a better foundation for recognising recent trends in elements to be scheduled for production and would learn to take actions to handle these in a better way. A suggested workflow at Overhalla Betongbygg utilising the developed solution is visualised in Figure 23.

As mentioned in 5.1, the data on concrete elements with assembly dates to suggest production dates on are stored in an Excel spreadsheet after being fetched from the database of Overhalla Betongbygg. I was given the impression that fetching data into a spreadsheet in this fashion was relatively normal practise at Overhalla Betongbygg. Therefore, I assume that this is the approach most likely to be used if the solution was to be taken into use at Overhalla Betongbygg. This would be the easiest way of doing it with the current version of the tool.

The work done up until the submission of this thesis has been focused on proving that the solution has the potential to be useful for Overhalla Betongbygg. Through the training process, it has been showed that the reinforcement learning algorithm suggested is able to learn how to achieve higher and higher rewards over time of training. That means that the model learns to act and set production dates in a way that creates production schedules considered increasingly good by Overhalla Betongbygg according to the environment defined which is intended to reflect Overhalla Betongbygg’s priorities for good production schedules.

Furthermore, the comparisons made between the production schedule created by the model and the one used at Overhalla Betongbygg shows that the trained model is capable of creating a production schedule performing similarly to the one executed at Betongbygg with less overtime and stricter constraints upheld.

However, some work does at the time of thesis submission remain in order for Overhalla Betongbygg to have a simple to use application, using a model trained in the way described, at their disposal.

In order to have a simple to use application using an already trained model to suggest production dates for new elements, some more work needs to be done.

Firstly, a simple to use interface should be created in order to make the solution useable for personnel with limited knowledge on software development. This interface should make it simple to import a new excel spreadsheet with concrete elements and have production dates suggested for elements which has not already been scheduled for production.

Another limitation of the software which preferably should be improved upon before taking the software in use is the handling of failed attempts at creating a production schedule. At the time of writing, an attempt at creating a production schedule will simply fail if all possible production dates for an element breaks constraints due to how the previous elements has been scheduled for production. During training, the model will be saved after it has performed well, making the models that were able to create a good production schedule available for later use. This does however not guarantee that this model will not fail in creating a production schedule for later concrete elements that were not part of the data set for the training of the model.

This could be handled in several different ways. One possibility is to have multiple models to automatically fall back if the first one fails. This way several different models could be given an attempt at creating a valid production schedule before it is considered failed.

Another solution could be to go back and reschedule previous elements in the event that no valid production date is found for a later element. This would possibly be an even better solution than attempting with multiple different stored models as it would not require having multiple models available and would eventually find a valid schedule if one is possible if this solution is implemented well. It would however probably take more work to implement than simply trying different models.

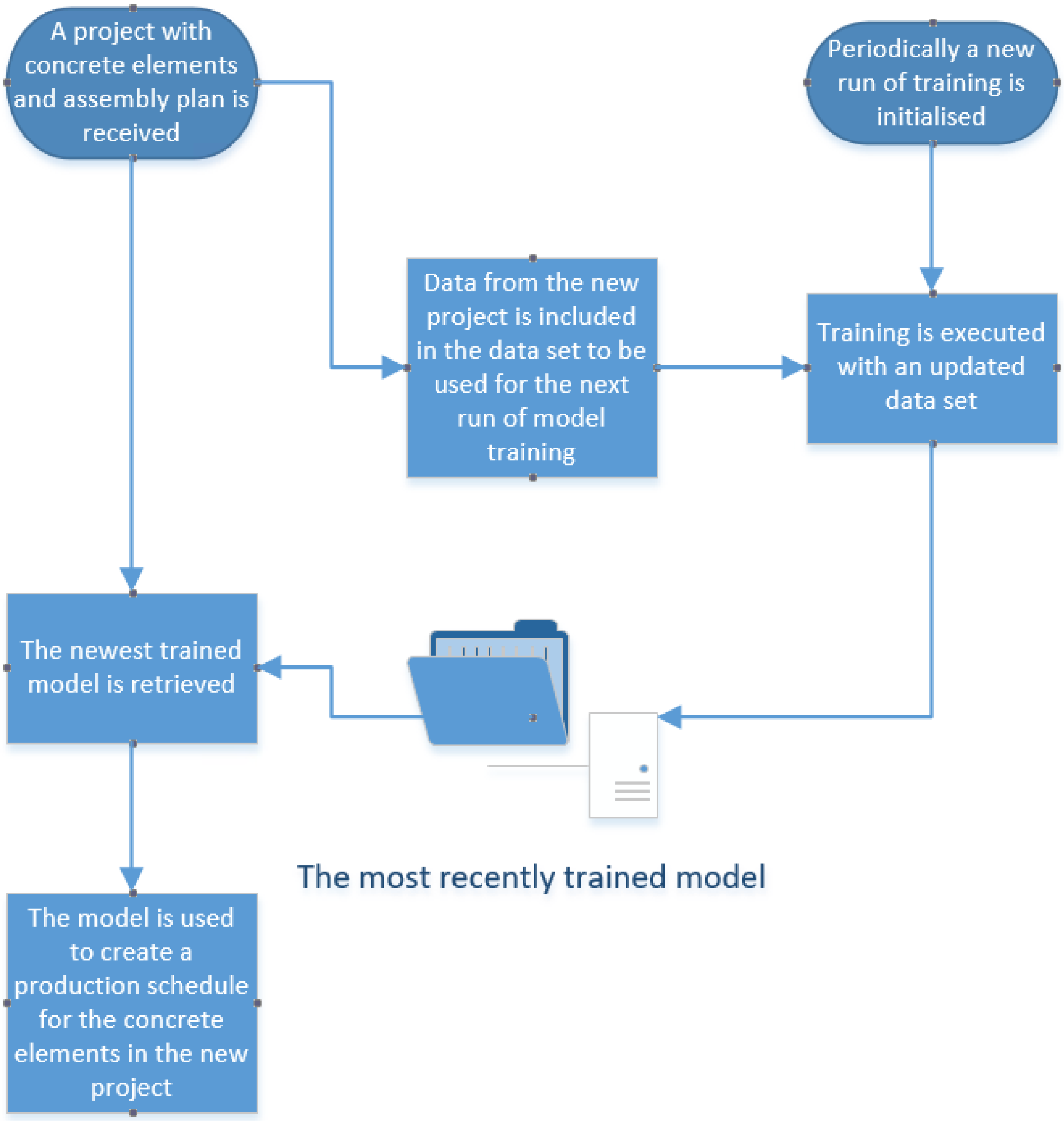

Figure 23: A suggested workflow with the use of the reinforcement learning solution developed at Overhalla Betongbygg.

7.3 Possible alternative solutions to the problem

The problem of scheduling the production of concrete elements at Overhalla Betongbygg is a problem that possibly could be solved with methods outside of artificial intelligence as well. Especially multi-objective optimisation seems to be promising for this problem. The reward calculation done in the reinforcement learning environment developed could possibly have been reformulated as multiple objective functions. Then these objective functions could be attempted maximised.

This way, a series of concrete elements to be produced at Overhalla Betongbygg could be assigned production dates in a way that optimises the objective functions that will be measures of how suitable the production schedule is.

However, these would not consider concrete elements to be scheduled for production at a later point in time.

With a reinforcement learning approach where one by one concrete element is assigned a production date and then a reward is calculated from measures of how suitable the assigned production date is with the previous assigned production dates considered and with the agent being adjusted using the summed reward for the entire production schedule, it is likely that the agent will learn to assign a production date in a way that makes assigning good production dates to upcoming elements as easy as possible.

If the agent were to adjust its model for every set production date based on the calculated reward, the model would be optimised to set the production date resulting in the maximum reward for that single element without considering the elements to be assigned a production date later. However, since the model is only adjusted once all of the elements at hand has been assigned a production date and it is adjusted using the sum of all the calculated rewards, the agent will have to opportunity to learn to assign production dates with the greater picture in mind: The whole production schedule. This is an advantage with the reinforcement learning approach over the multi-objective optimisation approach.

In addition, multi-objective optimisation can have problems with computation time for a complicated problem like this. Meanwhile a trained machine learning model can make decisions efficiently.

8 Conclusion

8.1 Contribution

Through my literature review, I was able to find multiple uses of artificial intelligence methods within production planning. These findings are described more in detail in 3.2. While I found older literature on the topic, it was the newer literature that had the most similarities and were of the highest relevance to this research were those from recent years. In the older literature, there was applications of artificial intelligence methods within production planning, but the artificial intelligence methods applied differ significantly from those typically applied in research from later years and from the method applied in this research. Only in more recent literature did I find other applications of methods from machine learning within production planning.

However, through my searches for relevant literature and relevant papers I found in other papers, I was not able to find reinforcement learning being used to train a model to be capable of creating a production schedule fitting of the characteristics of the production processes like what has been done for Overhalla Betongbygg in this project. This is the main contribution of my research.

Using reinforcement learning gives some advantages over using supervised learning for training. Reinforcement learning does not require labelled data to be able to learn. Instead, reinforcement learning offers more flexibility in the way it updates the model by using a calculated reward that can be calculated however is fitting to problem instead of a simply a measure of difference between the label and the model output like it is done with supervised learning. This should make reinforcement learning an attractive method to use in other planning problems where one often would not have the optimal solution to the problem ahead of time to use as a label.

Another interesting approach to such planning problems found in my literature review is genetic algorithms. Genetic algorithms seem to have been applied to planning problems similar to those present at Overhalla Betongbygg. However, from my previous knowledge of machine learning and reinforcement learning, I knew that reinforcement learning should be suitable for optimisation of these kinds of planning problems where I had seen genetic algorithms being applied. Therefore, trying to apply reinforcement learning to these problems arose as an even more interesting approach to me.

With the results of my research, I have showed that a solution using reinforcement learning can be capable of learning to perform better and better over time of training with an environment reflecting a production scheduling problem. It is of my opinion that these results show that reinforcement learning can be a suitable approaching in solving these kinds of planning problems and optimising the decision-making. Although the methods have its differences, it seems that in planning problems where genetic algorithms have been applied, using reinforcement learning could be an alternative approach.

8.2 Limitations and opportunities for further work

A limitation of this research is the comparisons made between the production schedule created by the machine learning model and the one created by the production planners at Overhalla Betongbygg. Such comparisons were difficult to make as, with the constraints defined in the reinforcement learning environment, the production schedule made by the production planners at Overhalla would be considered invalid. For this reason, the two schedules had a somewhat different basis for their comparison.

The production scheduling problem that is attempted solved in this research is deterministic. However, this is not taken advantage of with the reinforcement learning algorithm that was applied. That the problem is deterministic means that given a state and an action taken by the agent for this state, the new state the action would create could be known in advance. There are algorithms within reinforcement learning taking advantage of deterministic problems. However, due to time constraints, such an algorithm was not applied in this research.

As shown in Figure 6, at Overhalla Betongbygg, the processes within their production planning are highly dependent on each other. This project has only focused on one of these, the production scheduling of the concrete elements. As Figure 6 shows, the process following the production scheduling, the planning of transportation, is dependent on the production schedule and it would for this reason be beneficial to consider the transportation planning when creating the production schedule. This was not done in this project.

The same goes for the two processes in the production planning at Overhalla Betongbygg prior to the production schedule: dividing buildings into elements and creating assembly plans. In this project, a data set that was a result of these two processes done at Overhalla Betongbygg was used. The elements were already defined and the date of assembly for each of these elements were set. Perhaps the starting point for setting a good production schedule would have been even better if these two processes were executed differently with the production schedule and other processes even closer considered than what the workers at Overhalla Betongbygg were able to do due to human limitations. Integration of more of the production planning processes into a unified machine learning solutions could enable these complex considerations to be made efficiently.

Since genetic algorithms have been applied to similar planning problems, comparing a solution based on reinforcement learning to one based on genetic algorithms would be an interesting opportunity for further work.

9 References

AMIRI, R., MEHRPOUYAN, H., FRIDMAN, L., MALLIK, R. K., NALLANATHAN, A. & MATOLAK, D. A machine learning approach for power allocation in HetNets considering QoS. 2018 IEEE International Conference on Communications (ICC), 2018. IEEE, 1-7.

BETONGBYGG, O. *HISTORIKK* [Online]. overhallabetongbygg.no. Available: https://overhallabetongbygg.no/om-oss/historikk/ [Accessed 12.05.2021 2021].

BROWNLEE, J. 2017. *Why One-Hot Encode Data in Machine Learning?* [Online]. Machine Learning Mastery. Available: https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/ [Accessed 21.06.2021 2021].

BUENO, A. F., GODINHO FILHO, M. & FRANK, A. G. 2020. Smart production planning and control in the Industry 4.0 context: A systematic literature review. *Computers & Industrial Engineering*, 106774.

BUER, S.-V., STRANDHAGEN, J. W., STRANDHAGEN, J. O. & ALFNES, E. Strategic fit of planning environments: towards an integrated framework. International Conference on Information Systems, Logistics and Supply Chain, 2016. Springer, 77-92.

CHIH-TING DU, T. & WOLFE, P. M. 2000. Building an active material requirements planning system. *International Journal of Production Research,* 38, 241-252.

CHRISTODOULOU, P. 2019. Soft actor-critic for discrete action settings. *arXiv preprint arXiv:1910.07207*.

CHU, B., TOLONE, W., LONG, J., WILHELM, R., PENG, Y., FININ, T. & MATHEWS, M. L. 1998. Toward intelligent integrated manufacturing planning-execution. *The International Journal of Agile Manufacturing,* 1.

EL NAQA, I. & MURPHY, M. J. 2015. What is machine learning? *machine learning in radiation oncology.* Springer.

ESPEHOLT, L., SOYER, H., MUNOS, R., SIMONYAN, K., MNIH, V., WARD, T., DORON, Y., FIROIU, V., HARLEY, T. & DUNNING, I. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. International Conference on Machine Learning, 2018. PMLR, 1407-1416.

FINN, C., ABBEEL, P. & LEVINE, S. Model-agnostic meta-learning for fast adaptation of deep networks. International Conference on Machine Learning, 2017. PMLR, 1126-1135.

FOX, M. S., BARBUCEANU, M. & TEIGEN, R. 2001. Agent-oriented supply-chain management. *Information-based manufacturing.* Springer.

GEN, M., TSUJIMURA, Y. & IDA, K. 1992. Method for solving multiobjective aggregate production planning problem with fuzzy parameters. *Computers & Industrial Engineering,* 23**,** 117-120.

GOODALL, P., SHARPE, R. & WEST, A. 2019. A data-driven simulation to support remanufacturing operations. *Computers in Industry,* 105**,** 48-60.

GUAN, Y., LI, S. E., DUAN, J., LI, J., REN, Y., SUN, Q. & CHENG, B. 2019. Direct and indirect reinforcement learning. *International Journal of Intelligent Systems*.

HAARNOJA, T., ZHOU, A., ABBEEL, P. & LEVINE, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. International Conference on Machine Learning, 2018a. PMLR, 1861-1870.

HAARNOJA, T., ZHOU, A., HARTIKAINEN, K., TUCKER, G., HA, S., TAN, J., KUMAR, V., ZHU, H., GUPTA, A. & ABBEEL, P. 2018b. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

HSU, H.-M. & WANG, W.-P. 2001. Possibilistic programming in production planning of assemble-to-order environments. *Fuzzy sets and Systems,* 119**,** 59-70.

KACPRZYK, J. & STANIESKI, P. 1982. Long-term inventory policy-making through fuzzy decision-making models. *Fuzzy sets and systems,* 8**,** 117-132.

LEHTIMAKI, A. 1987. An approach for solving decision problems of master scheduling by utilizing theory of fuzzy sets. *International Journal of Production Research,* 25**,** 1781-1793.

LIANG, E., LIAW, R., NISHIHARA, R., MORITZ, P., FOX, R., GOLDBERG, K., GONZALEZ, J., JORDAN, M. & STOICA, I. RLlib: Abstractions for distributed reinforcement learning. International Conference on Machine Learning, 2018. PMLR, 3053-3062.

LIAO, S.-H. 2005. Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert systems with applications,* 28**,** 93-103.

LILLICRAP, T. P., HUNT, J. J., PRITZEL, A., HEESS, N., EREZ, T., TASSA, Y., SILVER, D. & WIERSTRA, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

METAXIOTIS, K. S., PSARRAS, J. E. & ERGAZAKIS, K. A. 2003. Production scheduling in ERP systems: An AI-based approach to face the gap. *Business Process Management Journal*.

MILLER, W., LEUNG, L., AZHAR, T. & SARGENT, S. 1997. Fuzzy production planning model for fresh tomato packing. *International Journal of Production Economics,* 53**,** 227-238.

MNIH, V., BADIA, A. P., MIRZA, M., GRAVES, A., LILLICRAP, T., HARLEY, T., SILVER, D. & KAVUKCUOGLU, K. Asynchronous methods for deep reinforcement learning. International conference on machine learning, 2016. PMLR, 1928-1937.

MNIH, V., KAVUKCUOGLU, K., SILVER, D., GRAVES, A., ANTONOGLOU, I., WIERSTRA, D. & RIEDMILLER, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

MUHAMMAD, I. & YAN, Z. 2015. SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY. *ICTACT Journal on Soft Computing,* 5.

MULA, J. 2004. *Models for production planning under uncertainty. Application in a company of the automobile sector.* Ph. D (in Spanish), Polytechnic University of Valencia, Spain.

MULA, J., POLER, R., GARCÍA-SABATER, J. P. & LARIO, F. C. 2006. Models for production planning under uncertainty: A review. *International journal of production economics,* 103**,** 271-285.

NACHUM, O., NOROUZI, M., XU, K. & SCHUURMANS, D. 2017. Bridging the gap between value and policy based reinforcement learning. *arXiv preprint arXiv:1702.08892*.

NGATCHOU, P., ZAREI, A. & EL-SHARKAWI, A. Pareto multi objective optimization. Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems, 2005. IEEE, 84-91.

NIKOLOPOULOS, C. & FELLRATH, P. 1994. A hybrid expert system for investment advising. *Expert Systems,* 11**,** 245-250.

OLUYISOLA, O. E. 2021. Towards Smart Production Planning and Control: Frameworks and case studies investigating the enhancement of production planning and control using internet-of-things, data analytics and machine learning.

OLUYISOLA, O. E., SGARBOSSA, F. & STRANDHAGEN, J. O. 2020. Smart production planning and control: concept, use-cases and sustainability implications. *Sustainability,* 12**,** 3791.

OPENAI. 2018. *Kinds of RL Algorithms* [Online]. openai.com: OpenAI. Available: https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html#part-2-kinds-of-rl-algorithms [Accessed 14.06.2021 2021].

PARK, K. S. 1987. Fuzzy-set theoretic interpretation of economic order quantity. *IEEE Transactions on systems, Man, and Cybernetics,* 17**,** 1082-1084.

PATHAK, D., AGRAWAL, P., EFROS, A. A. & DARRELL, T. Curiosity-driven exploration by self-supervised prediction. International conference on machine learning, 2017. PMLR, 2778-2787.

PORTER, B., MAK, K. & WONG, Y. Machine requirements planning and workload assignment using genetic algorithms. Proceedings of 1995 IEEE International Conference on Evolutionary Computation, 1995. IEEE, 711-715.

PROFF.NO. *Overhalla Betongbygg* [Online]. Available: https://www.proff.no/selskap/overhalla-betongbygg/overhalla/sement-m%C3%B8rtel-og-betong/IG5K9TW10KJ/ [Accessed 12.05.2021 2021].

RAMESH, A., KAMBHAMPATI, C., MONSON, J. R. & DREW, P. 2004. Artificial intelligence in medicine. *Annals of the Royal College of Surgeons of England,* 86**,** 334.

RASHIDI, H. H., TRAN, N. K., BETTS, E. V., HOWELL, L. P. & GREEN, R. 2019. Artificial intelligence and machine learning in pathology: the present landscape of supervised methods. *Academic pathology,* 6**,** 2374289519873088.

RAUCH, E., DALLASEGA, P. & MATT, D. T. 2018. Complexity reduction in engineer-to-order industry through real-time capable production planning and control. *Production Engineering,* 12**,** 341-352.

RAY. 2021. *Proximal Policy Optimization (PPO)* [Online]. https://docs.ray.io/: The Ray Team. Available: https://docs.ray.io/en/master/rllib-algorithms.html#ppo [Accessed 28.06.2021 2021].

REYNOSO, G., GRABOT, B., GENESTE, L. & VÉROT, S. Integration of uncertain and imprecise orders in MRPII. Proceedings of the 9th International Multi-Conference on Advanced Computer Systems. Miedzyzdroje, Poland (23-25 October), 2002.

RINKS, D. 1982. *The performance of fuzzy algorithm models for aggregate planning under differing cost structures*, North-Holland Publishing: Amsterdam.

RINKS, D. B. & DB, R. 1981. A heuristic approach to aggregate production scheduling using linguistic variables.

SCHULMAN, J., WOLSKI, F., DHARIWAL, P., RADFORD, A. & KLIMOV, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

SOMMER, G. 1981. Fuzzy inventory scheduling.

SUTTON, R. S., MCALLESTER, D. A., SINGH, S. P. & MANSOUR, Y. Policy gradient methods for reinforcement learning with function approximation. NIPs, 1999. Citeseer, 1057-1063.

SZEPESVÁRI, C. 2010. Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning,* 4**,** 1-103.

TURKSEN, I. 1988a. Approximate reasoning for production planning. *Fuzzy sets and Systems,* 26**,** 23-37.

TURKSEN, I. 1988b. An approximate reasoning framework for aggregate production planning. *Computer Integrated Manufacturing.* Springer.

VAN DIERDONCK, R. & MILLER, J. G. 1980. Designing production planning and control systems. *Journal of Operations Management,* 1**,** 37-46.

VAN ROSSUM, G. & DRAKE, F. L. 2009. *Python 3 Reference Manual*, CreateSpace.

WANG, R.-C. & FANG, H.-H. 2001. Aggregate production planning with multiple objectives in a fuzzy environment. *European journal of operational research,* 133**,** 521-536.

WARD, T., RALSTON, P. & DAVIS, J. 1992. Fuzzy logic control of aggregate production planning. *Computers & industrial engineering,* 23**,** 137-140.

WOLD, S., ESBENSEN, K. & GELADI, P. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems,* 2**,** 37-52.

YALE. 1997. *Categorical Data* [Online]. http://www.stat.yale.edu/: Yale University. Available: http://www.stat.yale.edu/Courses/1997-98/101/catdat.htm [Accessed 16.06.2021 2021].

YULDOSHEV, N., TURSUNOV, B. & QOZOQOV, S. 2018. Use of artificial intelligence methods in operational planning of textile production. *Journal of process management. New technologies,* 6**,** 41-51.

ZHANG, Y., WANG, J. & LIU, Y. 2017. Game theory based real-time multi-objective flexible job shop scheduling considering environmental impact. *Journal of Cleaner Production,* 167**,** 665-679.