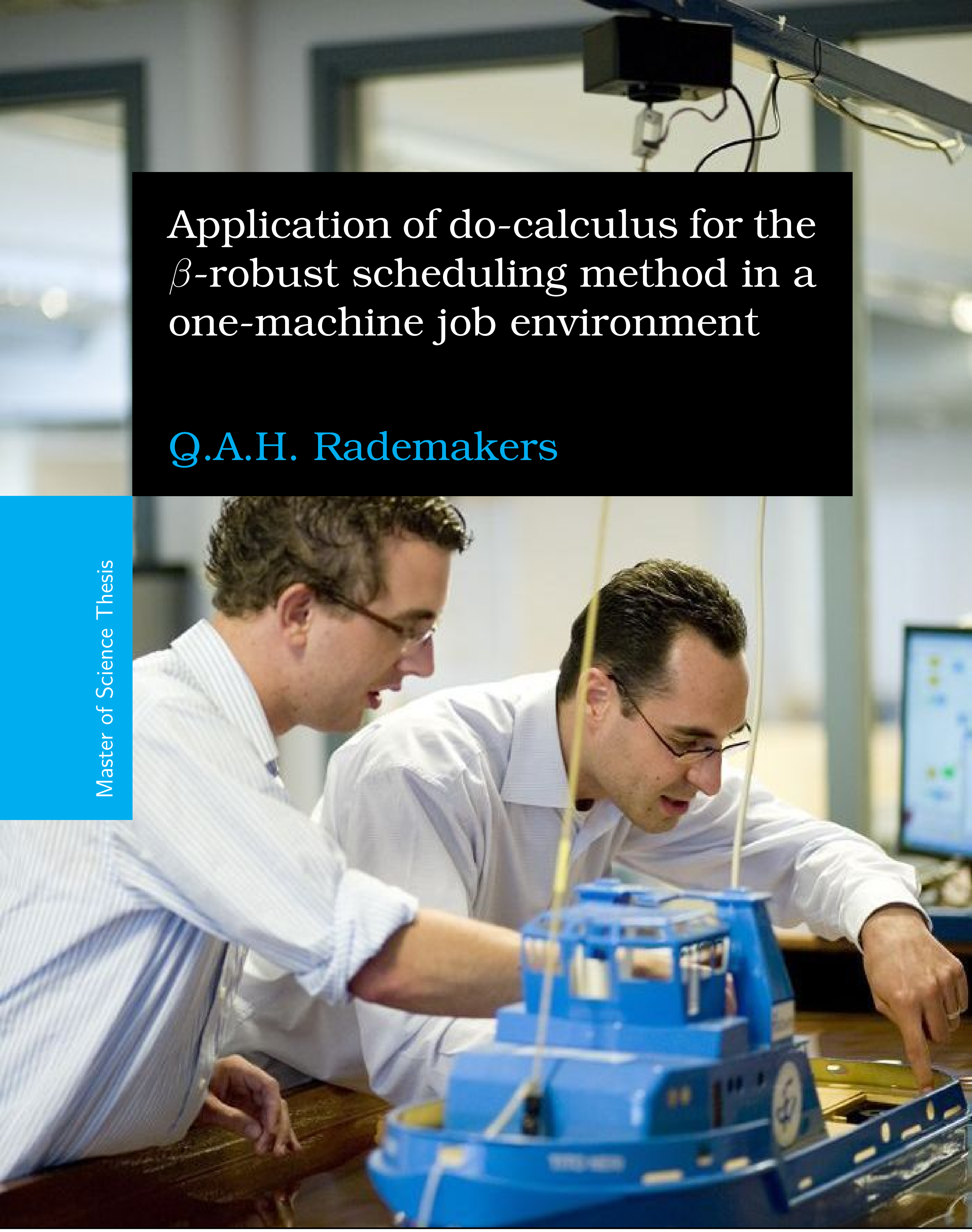

Application of do-calculus for the β-robust scheduling method in a one-machine job environment

Q.A.H. Rademakers

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

Application of do-calculus for the β-robust scheduling method in a one-machine job environment

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Q.A.H. Rademakers

August 28, 2021

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by Lely Industries N.V.. Their cooperation is hereby gratefully acknowledged.

Abstract

Job scheduling is the process where jobs are arranged in a specific sequence. This process has always been a crucial subject for companies in numerous industries. A company could significantly improve its business performance if the scheduling process is not performed optimally. As companies and products become more and more data-driven, new tools to improve processes are emerging. Robust job scheduling methods optimize job schedules with job duration uncertainties. Typical robust scheduling methods only use straightforward statistical metrics such as the job duration and variance. Data relevant to the process' job duration, stored as observational data, is not typically used in robust job scheduling methods. A field of research that is concerned with the use of observational data is causal inference. It can be applied to observational data to explain causal variable behaviour and therefore also the statistical metrics used in the scheduling process. In addition, it can be used to predict variable behaviour after certain variable modifications. This motivates our search for an approach that uses the tools of causal inference on available process data to identify the causal relations within these job processes. Based on this information, it predicts the effects of parameter modifications. These predictions are incorporated in the scheduling optimization to potentially further improve the job scheduling performance. In this thesis, we will therefore study research on causal inference and a method of robust job scheduling and propose an algorithm that extends the robust job scheduling method. The algorithm includes the tools of causal inference and additionally solution sensitivity analysis.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to make use of this moment to thank my supervisor Dr. P. Mohajerin Esfahani from the Delft Center for Systems and Control (DCSC) department for his guidance during the writing of this thesis. He encouraged me to explore a major part of my research that was previously unknown to me. He always did this in a highly motivating and positive manner. I am grateful to him for his support.

I would also like to thank my supervisors from Lely Industries N.V., Ir. M. Kerklaan and Ir. T. Horstink (currently Mainblades Inspections B.V.). They have actively provided me with support and feedback and stayed involved throughout the entire writing of this thesis. Their enthusiasm and involvement kept me motivated and comfortable during a period in which the amount of physical social interactions had been significantly restricted.

Finally, I would like to thank my family. Their presence truly makes me happy.

Chapter 1

Introduction

1-1 Background

Lely is one of the Netherlands' largest robotics companies and a market leader in the agricultural and robotics sector. They produce state-of-the-art autonomous robot systems and automation software for the dairy industry. One of their robot systems is the Lely Vector which manages the cow feed allocation within a farm. It consists of a mixing and feeding robot (MFR), a power distribution box (PDB) and a feed grabber (FG). In one job cycle, the MFR charges at the PDB, collects feed from the FG and mixes and distributes the feed along a path throughout the farm and finally returns to charge at the PDB. This happens numerous times per day along several paths and is performed autonomously. The entire process ensures continuous availability of feed for the cows which in turn improves high-quality dairy products while minimizing the workload of the farm owner. Since a farm is a dynamic environment, the MFR faces collisions on the pathways or diverse events such as system errors and breakdowns. The feed distribution cycle durations are therefore uncertain with multiple possible outcomes. The set of cycles have to be scheduled extensively in order to maximize efficiency in terms of time and energy consumption. In addition, to prevent food waste due to oxidation and discomfort of hungry cows the tasks should be finished within a reasonable amount of time. A certain deadline should therefore not be exceeded to ensure optimal business performance. Finding an optimal feeding schedule that corresponds to the lowest risk of this certain deadline being exceeded is one of the main challenges for the farm owner.

An approach that solves this type of robust scheduling problems is the β-robust scheduling method [1] which functions in a one-machine scheduling environment. The β-robust scheduling method is concerned with the maximization of the probability that the flow time of a schedule does not exceed a given deadline. This method applies to the Lely Vector since one job cycle can be modelled as a single job which makes the full feed distribution problem a one-machine scheduling problem. The β-robust scheduling model has also been presented as a constraint model by Wu et al. [2] to make it more convenient to combine the model with other scheduling models. This constraint representation however does not outperform the original β-robust model by Daniels and Carillo. An extension to the $\beta-$robust modelling schedule by Ullah [3] made the schedule applicable to two-machine flow-shops and Alimoradi [4] to parallel two-machine flow-shops. For β-robust scheduling problems where each operation has a common due date, an extension was developed by Khatami et al. [5]. The β-robust schedule is concerned with good schedule performance that is measured by predefined target threshold T. An approach to optimize the schedule for performance while minimizing the target threshold T has been developed by Wang et al. [6]. The benefit of this approach is that a schedule may be found that performs sufficiently for values of T that are lower than previously considered. On the other hand, a schedule for higher values of T may be found if schedule performance for the desired value of T is insufficient

A branch-and-bound optimization algorithm was used in the original β-robust scheduling method. Several other optimization methods are used in robust scheduling. Leon [7] developed a genetic algorithm for job shops that follow the "right-shift" policy. Sevaux [8] applied a genetic algorithm on a single-machine scheduling problem to minimize the weighted number of 'late' jobs, being jobs that exceed the makespan threshold. Jensen [9] performed a genetic algorithm approach for a job shop facing machine breakdowns and Sevaux [10] performed a genetic algorithm optimization for a single-machine job shop. Van Laarhoven et al. [11] applied a simmulated annealing technique, and Dell'Amico and Trubian [12] applied a tabu-search technique ta a job-shop scheduling problem. All of the mentioned optimization algorithms can be applied to obtain (near-)optimal solutions to hedge against job delays in an uncertain job shop scheduling environment.

Hedging against job delays could be done even more effectively. A data-driven approach could be considered to determine the events that have caused the job cycle duration. Several algorithms have been developed throughout the years to identify the causal relations within a data set. For this problem the PC algorithm [13] has been developed, which identifies the causal model of the variable set after performing conditional independence tests. The found independence constraints are used to determine the structure of the graph. This makes the PC algorithm a constraint-based algorithm. The PC algorithm functions under three assumptions, including the absence of unobserved variables that cause bias in the data set. It performs tests consequently for an ascending order of conditional independence relations. For high-order data sets the results could be variable and to improve on this, Colombo [14] developed the stable PC algorithm. The stable PC algorithm performs conditional independence tests order-independently. To combat undesirable run times resulting from high-order data sets, Le et al. [15] developed the parallelised PC algorithm which performs conditional independence tests that are unrelated in parallel, reducing the run time significantly.

In reality, the absence of unobserved variables that cause bias is naive to assume. For causal discovery on a data set for which unmeasured variables are present (incomplete data set), either the modified PC algorithm by Tu et al. [16] or the FCI algorithm [13] can be considered. Both of these algorithms are an extension to the PC algorithm, which make them part of the constraint-based causal discovery methods. They're able to estimate an equivalence class of a DAG under the assumption that unobserved variables are present that influence the observed system variables. Colombo et al. [17] added to the FCI algorithm to create the RFCI (Real Fast Causal Inference) algorithm, making it faster than the FCI algorithm and computationally feasible for high-dimensional data sets in the asymptotic limit. The Lingam discovery algorithm (Shimizu [18]) can estimate the complete causal structure of a model under the assumption that the disturbance data distribution is non-Gaussian and no unobserved variables are present.

Apart from the constraint-based causal discovery methods, the score-based algorithms are also worth noting. The result of these methods is a causal model which is considered optimal under a used score gauge, hence categorizing these methods as score-based discovery methods. Chickering [19] developed the Greedy Equivalence Search lgorithm (GES), which is the best known score-based discovery method. This is a two-phase algorithm that functions under the assumption that no confounding latent variables are present as is the case for the PC algorithm. This algorithm starts with an empty model and greedily adds single-edges until a local maximum for a predetermined score gauge is reached. In the second phase the algorithm greedily deletes single-edges until a local maximum is reached yet again. The result is a DAG which can be considered optimal under the used score gauge.

Figure 1-1: The Lely Vector MFR.

When the causal model is obtained, predictions on variable values can be made. Pearl invented do-calculus [20] as a tool to express predictions on the effects of manipulated variable values in terms of observational values. The completeness of do-calculus was proved by Huang and Valtorta [21]. With the focus on job scheduling, do-calculus could be used to predict the job duration values for certain modifications of system parameters. For each of these predicted scenarios a corresponding optimal schedule must then be determined. If a schedule corresponding to one of these predicted scenarios is found to outperform the schedule under current conditions, schedulers could apply these modifications in practice to optimize their business performance.

In practice, it could be that the values used for the optimization slightly differ from the true values. This can be easily explained as data is prone to noise. Minimizing the likeliness that these perturbations affect the optimality of the solution is crucial to safeguard the reliability of the scheduler. Therefore, it is very useful to know the maximum allowed perturbation of the input value for which the optimal solution remains optimal. An approach to do so in robust job scheduling was developed by Sotskov [22]. This maximum allowed perturbation is known as the stability radius of the solution. Ensuring a large stability radius means the scheduler has likely chosen the true optimal solution which in turn safeguards the reliability of the scheduler. After applying the tools of causal inference on eligible observational data, multiple scenarios and thus multiple input values can be considered for the optimization process to generate additional solutions. As more solutions can then be considered, the stability radius becomes a useful additional measure of performance that can be included.

1-2 Contribution

The main contribution of this work is the proposed algorithm that includes causal inference and solution sensitivity analysis in the β-robust scheduling optimization in a one-machine job environment. In this algorithm, the application of causal inference generates additional input values for the optimization based on given observational data. This means multiple optimal solutions are generated on which sensitivity analysis is performed. The addition of causal inference and sensitivity analysis allows the scheduler to optimize the job schedule for performance and solution sensitivity. Originally, the method of sensitivity analysis computes a perturbation bound for the only optimal solution based on the obtained job duration data which makes it an *a posteriori* approach. In our algorithm, sensitivity analysis is also performed on optimal solutions for predicted scenarios, based on observational data. This makes the algorithm a hybrid *a priori/a posteriori* approach. The benefit of the algorithm is that it might allow the scheduler to avoid a scenario with a predicted sensitive solution, where that isn't possible for the original *a posteriori* approach.

Combining these fields of research should illustrate the potential benefits of causal inference in job scheduling and incentivize the reader to investigate the applicability on a compatible real data set with the intent of improving a job scheduling process.

1-3 Outline

In Chapter 2 of this thesis the reader is introduced to basic deterministic one-machine robust job scheduling. Second, the concept of causal inference is discussed in Chapter 3. This consists of the concept of causal discovery, and predictions on the effects of variable manipulations following the rules of do-calculus. In Chapter 4 solution sensitivity analysis is explained. In Chapter 5 our problem is formulated. Finally, in Chapter 6 the results of simulations on a synthetic data set are shown and in Chapter 7 the conclusions of this thesis and suggestions for future work are presented.

Chapter 2

Robust scheduling

In real world applications it can not truly be expected for job durations to be consistent. Physical systems are subjected to wear and may fail after frequent use, decreasing their expected performance compared to their original performance, also humans are known to be imperfect and will always be capable of causing errors resulting in physical injuries, material damage and thus loss of time. Job durations should therefore be modelled as stochastic variables. Robust scheduling methods use these uncertain variables as input and determine a job schedule which is optimized for both job duration, sensitivity for job disruptions (robustness) or both. Having a schedule assigned to a system for which the process time is the least likely to exceed a given threshold provides should prove the system owner a certain amount of comfort. In this chapter β-robust scheduling will be discussed and applied on two example scenarios as method of predictive scheduling. β-robust scheduling is a probabilistic method that uses mean job duration values and the job duration variances as inputs and optimizes the schedule for the probability of the flow time not exceeding a given threshold T.

2-1 β-robust modelling

The job duration mean and variance values will be used as input for the optimization. The aim is to determine the job schedule with n independent jobs which has the highest probability of the total flow time time not exceeding a given threshold T. This threshold T can be chosen such that the resulting job schedule is either risk-averse or risk-seeking. The β-robust scheduling method is therefore a useful tool for robust scheduling. The variable notations of Section 2 [1] are used for the β-robust scheduling problem.

Input values

- job duration means: $\{\mu_1, \mu_2, ..., \mu_n\}$, $\quad \mu = \sum_{i=1}^{n} \mu_i$
- job duration variances: $\{\sigma_1^2, \sigma_2^2, ..., \sigma_n^2\}$, $\quad \sigma^2 = \sum_{i=1}^{n} \sigma_i^2$

Decision variable

- Assignment variable: $x_{ik} \in \{0,1\} : i = 1,2,...,n$ and $k = 1,2,...,n$.

x_{ik} indicates whether job J_i is scheduled at position k in the sequence, as is illustrated in Figure 2-1. Given that job J_i is scheduled at position k, $x_{ik} = 1$, and $x_{ij} = 0$ for $j = 1,...,n$, $j \neq k$.

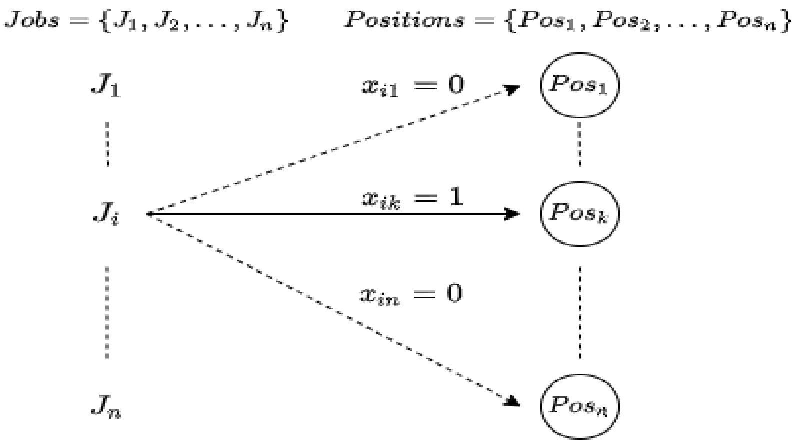

Figure 2-1: Decision variable x.

Constraints

Each order in the sequence can have a maximum of one job assigned, and each job can only be assigned once in the job sequence. Naturally, the control actions are assigning a job within the sequence. Together this is formulated as the following set of constraints

$$\sum_{k=1}^{n} x_{ik} = 1, \quad i = 1,2,...,n, \tag{2-1a}$$

$$\sum_{i=1}^{n} x_{ik} = 1, \quad k = 1,2,...,n, \tag{2-1b}$$

$$x_{ik} \in \{0,1\}, \quad i = 1,2,...,n, \quad k = 1,2,...,n. \tag{2-1c}$$

A sequence $\{J_4, J_2, J_3, J_5, J_1\}$ would then be given as

$$x = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\{J_4, J_2, J_3, J_5, J_1\}.$$

Performance variables

The goal of this optimization problem is to maximize the probability of the schedule's total flow time not exceeding threshold T. The total weighted flow time for scenario λ is given as

$$FT = \phi(x, p^\lambda) = \sum_{i=1}^{n}\sum_{k=1}^{n}(n-k+1)p_i^\lambda x_{ik}, \tag{2-2}$$

where p_i^λ is the actual job duration value of job J_i for scenario $\lambda \in \Lambda$. Λ is the set of all possible outcomes, which implies the uncertainty of this problem.

The total weighted mean flow time and total weighted variance are given as

$$\bar{\phi}(x) = \sum_{i=1}^{n}\sum_{k=1}^{n}(n-k+1)\mu_i x_{ik}, \tag{2-3}$$

$$\sigma^2(x) = \sum_{i=1}^{n}\sum_{k=1}^{n}(n-k+1)^2\sigma_i^2 x_{ik}. \tag{2-4}$$

Due to the added weights, the distribution of $\phi(x, p^\lambda)$ approaches a normal distribution for an increasing n. The distribution of $\phi(x, p^\lambda)$ approaches a normal distribution for even a small number of jobs ($n \approx 4$) when job duration time distributions are unimodal and nearly symmetric [23]. The probability that $\phi(x)$ does not exceed threshold T can therefore modelled in the standard-normal form

$$z(x, T) = \frac{T - \bar{\phi}(x)}{\sqrt{\sigma^2(x)}}. \tag{2-5}$$

Objective function

The objective of β-robust scheduling is to maximize the expression of Equation 2-5. Substituting Equations 2-3 and 2-4 into this equation and maximizing the resulting expression leads to the objective function

$$\max_x \frac{T - \sum\limits_{i=1}^{n}\sum\limits_{k=1}^{n}(n-k+1)\mu_i x_{ik}}{\sqrt{\sum\limits_{i=1}^{n}\sum\limits_{k=1}^{n}(n-k+1)^2\sigma_i^2 x_{ik}}}. \tag{2-6}$$

with $\Phi(x)$ being the objective value of the optimization problem. The value of Equation 2-5 is a z-statistic of a one-tailed normal distribution test. This value can be given as a probability value by directly converting the z-statistic to the normal cumulative distribution value. This can be done by following a normal distribution table or simply using Matlab's *normcdf* function, which then holds as

$$\text{normcdf}(\Phi(x)) = P(\phi(x) \leq T) \tag{2-7}$$

This optimization problem is concerned with a binary decision variable and a non-convex maximization function 2-6 with linear constraints 5-4-2-1c. The optimization problem will therefore be treated as a mixed-integer-nonlinear programming problem (MINLP).

To illustrate the advantage of a β-robust schedule, it will be compared with the schedule with the lowest expected flow time. Two examples ($n = 3$) are shown in this section.

First example

A one machine job shop system with jobs $\{J_1, J_2, J_3\}$ ($n = 3$) with job processing time uncertainties is considered. The schedule should preferably correspond to the lowest expected flow time possible, however, its main requirement is that it may not exceed $T = 62$. The specifications of each job are given in Table 2-1.

Job	J_1	J_2	J_3
μ	12	6	9
σ^2	1	24	1

Table 2-1: First example optimization input values.

If the objective for this problem would be to minimize the expected job flow time, the objective function would be given as

$$\min_x \bar{\phi}(x) = \min_x \sum_{i=1}^{n} \sum_{k=1}^{n} (n-k+1)\mu_i x_{ik}. \tag{2-8}$$

The optimal solution for this problem is the shortest expected processing time (SEPT) schedule x_{SEPT}. The jobs in the SEPT schedule are scheduled in ascending order starting from J_i for $\min_i(\mu)$ ($i = 1, 2, 3$). This corresponds to the schedule

$$x_{SEPT} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$
$$\{J_2, J_3, J_1\}.$$

This total weighted mean flow time of this schedule is $\bar{\phi}(x_{SEPT}) = 48$. The aim of β-robust scheduling however is to find the sequence for which the total flow time is most likely to not exceed target performance T (Equation 2-6). Given that $T > \phi(x_{SEPT})$, a risk-averse schedule must be chosen to maximize the probability that the total flow time does not exceed T (Theorem 2. [1]). The solution to the β-robust scheduling problem is then given as

$$x^* = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix},$$
$$\{J_3, J_2, J_1\}$$

For both schedules, the weighted mean flow time with corresponding weighted standard deviation is shown in Figure 2-2.

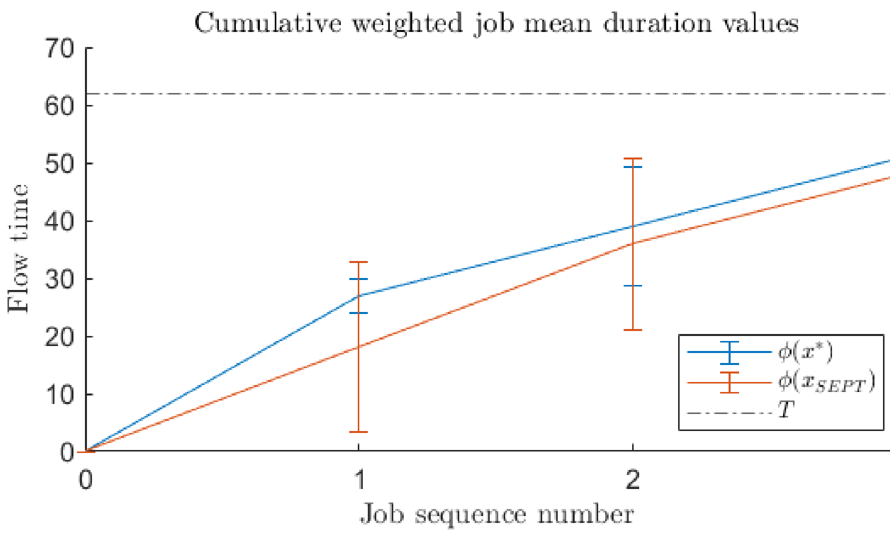

Figure 2-2: First example weighted flow times comparison.

It can be seen that schedule x_{SEPT} performs better than schedule x^* in terms of total weighted mean flow time. However, the total weighted variance of schedule x^* is smaller than that of x_{SEPT}, which is beneficial for risk-aversion. The total weighted variance should be low enough to compensate for the surplus in total weighted mean flow time. If this is the case, the actual flow time of schedule x^* is then less likely to exceed target performance $T = 62$, which is desired. As this is a one-machine scheduling problem with 3 jobs, there exist 3! feasible schedules. The performance values of both schedules are given in the top part of Table 2-2. Those of the remaining feasible schedules $\{x_3, x_4, x_5, x_6\}$ are given in the bottom part of the table.

	sequence(x)	$\bar{\phi}(x)$	$\sigma^2(x)$	$\Phi(x)$	$P(\phi(x) \leq T)$
x_{SEPT}	$\{J_2, J_3, J_1\}$	48	221	0.941	0.826
x^*	$\{J_3, J_2, J_1\}$	51	106	**1.068**	**0.857**
x_3	$\{J_3, J_1, J_2\}$	57	37	0.822	0.794
x_4	$\{J_1, J_3, J_2\}$	60	37	0.328	0.628
x_5	$\{J_2, J_1, J_3\}$	51	221	0.739	0.770
x_6	$\{J_1, J_2, J_3\}$	57	106	0.485	0.686

Table 2-2: First example schedule performance values.

From these example results, it can be seen how the β-robust schedule nearly performs optimally in terms of mean weighted total flow time. It performs optimally in terms of robustness, of which the latter has our main priority.

Second example

It is desired to maximize the probability that the total mean flow time corresponding to the schedule does not exceed $T = 38$. The total weighted mean flow time of schedule x_{SEPT} and x^* is given in Figure 2-3.

Job	J_1	J_2	J_3
μ	6	12	9
σ^2	1	1	24

Table 2-3: Second example optimization input values.

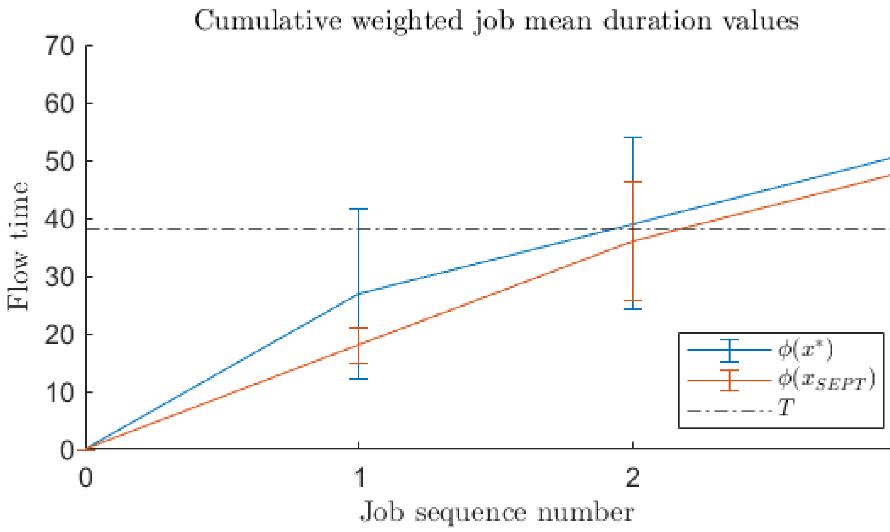

Figure 2-3: Second example weighted flow times comparison.

Performance target T is now significantly lower than in the previous example. Given that $T < \phi(x_{SEPT})$, a more risk-seeking schedule must be chosen to increase the probability that the total mean flow time does not exceed T (Theorem 2. [1]). It can be seen in Figure 2-3 that the total weighted variance of schedule x^* now is larger than for the previous example, which shows how the β-robust schedule now is a more risk-seeking schedule.

	sequence(x)	$\bar{\phi}(x)$	$\sigma^2(x)$	$\Phi(x)$	$P(\phi(x) \leq T)$
x_{SEPT}	$\{J_1, J_3, J_2\}$	48	106	-0.971	0.165
x^*	$\{J_3, J_1, J_2\}$	51	221	**-0.874**	**0.190**
x_3	$\{J_3, J_2, J_1\}$	57	221	-1.278	0.100
x_4	$\{J_2, J_3, J_1\}$	60	106	-2.136	0.016
x_5	$\{J_2, J_1, J_3\}$	57	37	-3.123	0.000
x_6	$\{J_1, J_2, J_3\}$	51	37	-2.137	0.016

Table 2-4: Dummy example schedule performance values.

In Table 2-4 it can be seen how the β-robust schedule once again nearly performs optimally in terms of total weighted mean flow time and performs optimally in terms of robustness. This proves β-robust modelling to be a useful tool for robust scheduling.

Chapter 3

Causal inference

Causal inference is the process of drawing causal conclusions based on observational data. Our aim is to apply this in robust job scheduling when a data set of a job was to be available where the job duration values along with observational values for different parameters have been measured. This data set should enclose all of the events concerned with the process, where the job duration time also is a measured event. The causal effect of a system variable adjustment on the performance variable can be predicted if the data set meets certain conditions. These predicted values will be used as additional inputs for the robust scheduling method.

Causal effect (Definition 2 [20]): Given two disjoint sets of variables, X and Y, the causal effect of X on Y, denoted $P(y|do(x))$, is a function from X to the space of probability distributions on Y. For each realisation x of X, $P(y|do(x))$ gives the probability of Y = y induced on deleting from the model all equations corresponding to variables X and substituting x for X in the remainder.

A large portion of statistics and causality uses graphical models to model variable relations. These graphical models represent variable relations visually and are constructed following the rules of graph theory and probability calculus. Also, these graphical models can be exploited to identify (in)dependence relations which enable us to make these predictions mentioned earlier. Therefore, several probability calculus rules and an introduction to some graph-theoretic elements are given and discussed in Section A-1 of the Appendix. The reader can consult this section for his/her understanding of the subject and the remaining of this chapter.

In Section 3-1 two conditions are discussed that must be satisfied by the observational data set. In Section 3-2 the concept of causal discovery is discussed and a method will be applied on an example. Finally, in Section 3-3 the rules of do-calculus will be explained and applied on an example.

3-1 Model assumptions

For our research, it is required for the models to satisfy two conditions before we are allowed to successfully draw conclusions on conditional independence relations within the model. If these assumptions hold, the causal and probabilistic dependencies within the model are connected. We will therefore assume these assumptions to hold throughout the rest of the thesis.

Causal Markov Condition (3.4.1 [13]): Let G be a causal graph with vertex set V and P be a probability distribution over the vertices in V generated by the causal structure represented by G. G and P satisfy the Causal Markov Condition if and only if for every W in V, W is independent of V\(Descendants(W) ∪ Parents(W)) given Parents(W).

The practical use of this condition is that now for each of these child-parent node relations the corresponding probability distribution can be determined. Then the joint probability function as a factorization of these individual probability distributions of the entire graph for which the Causal Markov Condition holds.

Faithfulness Condition (3.4.3 [13]): Let G be a causal graph and P a probability distribution generated by G. <G, P> satisfies the Faithfulness Condition if and only if every conditional independence relation true in P is entailed by the Causal Markov Condition applied to G.

This assumption guarantees that there are no additional underlying conditional dependencies that can not be visually determined from the causal graph. The importance of this assumption is that it guarantees the true probability distribution to be equal to that of the graphical representation. Both the Causal Markov Condition and the Faithfulness Condition must be satisfied for P to be faithful to G.

Example

Consider a data set X_D of a dairy farm that is equipped with a feed distributing robot system. The dairy farm owner has accurately collected a large set of observational data for a large number of job cycles. The set of measured variables is given below as

- x_m: elapsed hours since the latest software and hardware maintenance round,
- x_p: total measured hourly precipitation amount for the last 48 hours in millimeters,
- x_e: amount of error flags shown in software during job cycle, triggering a temporary standstill,
- x_r: measured rust content in robot components,
- x_t: job cycle timespan.

The directed acyclic graph G_D of variable set X_D is shown in Figure 3-1.

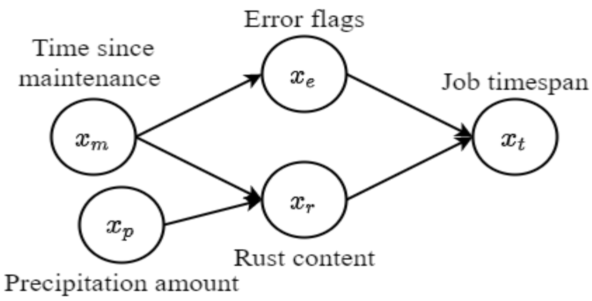

Figure 3-1: Directed graph G_D of variable set X_D.

Following the Causal Markov Condition, we can conclude

- Parents(x_m) = ∅ and Parents(x_p) = ∅ which shows $x_p \perp\!\!\!\perp x_m$,
- Parents(x_p) = ∅, $x_e \notin$ Descendants(x_p) and $x_m \notin$ Descendants(x_p) which shows ($x_p \perp\!\!\!\perp \{x_m, x_e\}$),
- Parents(x_r) = $\{x_p, x_m\}$, $x_e \notin$ Descendants(x_r) which shows ($x_r \perp\!\!\!\perp x_e | \{x_p, x_m\}$).

We can explain independence relation $x_p \perp\!\!\!\perp x_m$ as both the precipitation amount and the time since last maintenance logically cannot be a causal effect of another variable within the data set. The other events are incapable of altering the ways of nature and therefore the true directed acyclic graph of data set X_D will show no edges going into x_p. Also, the time since last maintenance cannot be influenced by any given event. Only the robot's responsible mechanic could execute the maintenance round and reset x_m to zero. However, this has not been the case during measurements and therefore is not be included in the model. Given this information, we can conclude $x_p \perp\!\!\!\perp x_m$.

3-2 Causal discovery

The aim of causal discovery is to identify causal relations in a model from its data and model structure. We can use these relations to update the estimate of the model. Consequently, the model can be used to make predictions on variable values. This makes causal discovery a useful tool and a crucial part of our β-robust scheduling model extension.

3-2-1 D-separation

The d-separation criterion is a powerful tool for the graphical identification of independence relations in a model. When a causal model satisfies both the Causal Markov Condition and the Faithfulness assumption hold, we can apply the d-separation criterion. When nodes X and Y are d-separated by a set of nodes Z, it is noted as $X \perp\!\!\!\perp Y | Z$. Holmes' definition of the d-separation is given below as

D-separation criterion (Definition 1 [24]): X and Y are d-separated given Z (for any subset Z of variables not including X or Y) if and only if each distinct path Φ between them is cut by one of the graph-theoretic conditions:

1. Φ contains a chain $X_1 \rightarrow X_2 \rightarrow X_3$ and $X_2 \in Z$.
2. Φ contains a common causal structure $X_1 \leftarrow X_2 \rightarrow X_3$ and $X_2 \in Z$.
3. Φ contains a common effect structure $X_1 \rightarrow X_2 \leftarrow X_3$ (i.e. an uncovered collision with X_1 and X_3 not directly connected) and neither X_2 nor any descendant of X_2 is in Z.

Consider directed graph G_d of variable set V_d which is shown in Figure 3-2. According to the d-separation criterion, several independence relations can be identified.

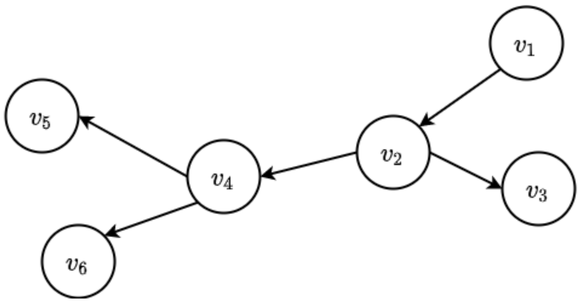

Figure 3-2: Directed graph G_d of variable set V_d.

First, two independence relations can be identified since

- v_2 blocks all paths through chain $v_1 \rightarrow v_2 \rightarrow v_3$ which indicates $(v_1 \perp\!\!\!\perp v_3 | v_2)$,
- v_2 blocks all paths through chain $v_1 \rightarrow v_2 \rightarrow v_4$ which indicates $(v_1 \perp\!\!\!\perp v_4 | v_2)$.

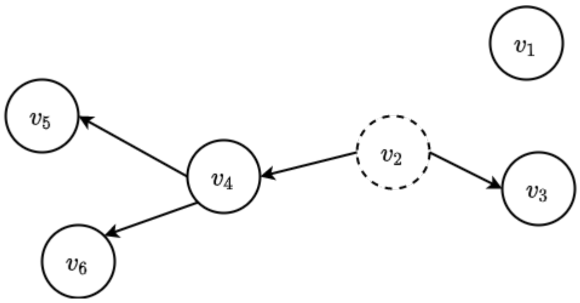

Figure 3-3: Directed graph G_d of variable set V_d conditioned on node v_2.

In Figure 3-3, the causal influence of v_1 on v_3 and v_4 is eliminated when conditioning on node v_2. Doing so makes v_1 independent of both v_4 and v_3 conditioned on v_2.

Second, another two independence relations can be indentified since

- v_4 blocks all paths through chain $v_2 \rightarrow v_4 \rightarrow v_5$ which indicates $(v_2 \perp\!\!\!\perp v_5 | v_4)$,
- v_4 blocks all paths through chain $v_2 \rightarrow v_4 \rightarrow v_6$ which indicates $(v_2 \perp\!\!\!\perp v_6 | v_4)$.

Similarly to node v_2, it can be seen in Figure 3-4 how the causal influence of v_2 on v_5 and v_6 is eliminated when conditioning on node v_4. Doing so makes v_2 independent of both v_5 and v_6 conditioned on v_4.

Also after following this criterion for the graph from Figure 3-1 it can be seen that

- node pair $\{x_e, x_r\}$ blocks all paths from x_m to x_t which indicates $(x_t \perp\!\!\!\perp x_m | \{x_e, x_r\})$,
- node x_r blocks all paths from node x_p to node x_t which indicates $(x_t \perp\!\!\!\perp x_p | x_r)$.

3-2-2 PC algorithm

A widely used causal discovery method is the PC algorithm. For this algorithm to apply the additional assumption that no unmeasured variables that cause bias are present must hold for

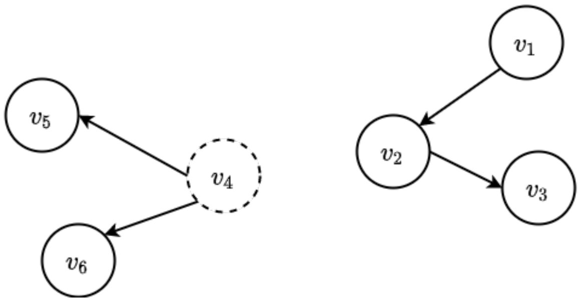

Figure 3-4: Directed graph G_d of variable set V_d conditioned on node v_4.

the underlying data set. The PC algorithm applies conditional independence tests and edge orientation tests on variable sets within a given data set. The independence constraints are used to determine the structure of the graph, making the PC algorithm a constraint-based causal discovery method. The PC algorithm determines variables to be causally related if no subset exists on which can be conditioned for the variables to be independent. Kalisch and Bühlmann [25] proved this algorithm to estimate the equivalence class of high-dimensional DAGs under a sparseness assumption. Sparse graphs can consist of a large number of nodes but only a few edges connecting these nodes, limiting the complexity of these graphs.

The output of the causal discovery method may be the true DAG or a Markov equivalent of the true DAG. A Markov equivalence class model of a true model represents the same set of conditional independence relations as the true model. The orientation of an edge connecting 2 variables can switch, but the same conditional independence relation by d-separation holds. An algorithm therefore will not always be able to distinguish between the true and Markov equivalent models. In certain scenarios this is an inconvenience. If two estimated Markov equivalent DAGs differ in one edge orientation, the true outcome of an intervention on a variable connected to this edge is uncertain. It may not even come close to one of two predicted values. The reason causal inference is applied is to be able to estimate what the causal effect is for an intervention, without having to apply the intervention in practice. Therefore, this uncertainty is undesired and this characteristic of a Markov equivalent model is considered a limitation in causal inference.

As the variable sets we will consider are assumed to be complete (no unobserved variables causing bias) and represent low-dimensional variable sets, the PC algorithm is sufficient for our research and is therefore chosen as our causal discovery method. The algorithm is also given in the Appendix in Section A-2.

Application of PC algorithm

After a set of variables is given as input, the first part of the algorithm is initiated and a fully connected DAG of the variable set is formed. Once the connected graph is formed, edges are removed by d-separation between node pairs that are determined to be independent when conditioned on a corresponding node. If an edge connecting node pair (i,k) is removed by d-separation after conditioning on node j, where both node pairs (i,j) and (j,k) are adjacent, node j forms the separation set S_{ik} of node pair (i,k). During the first removal step, the algorithm addresses zero-order conditional independence relations. The order increases by one per algorithm step. The algorithm continues with the edge removal steps until no conditional independence relation can be found that applies to the order of the step. At this point the skeleton of the DAG has been determined. The skeleton of a DAG consists of the same set of nodes and edges as the true DAG, but these edges are undirected.

In the second part, the edge directions will be assigned. For each node triplet (i,j,k) where (i,j) and (j,k) are an adjacent pair of nodes but not (i,k), the edge orientation will become $i \longrightarrow j \longleftarrow k$ if and only if node j does not belong to separation set $S_{i,k}$.

Finally, for each node triplet (i,j,k), where node pairs (i,j) and (j,k) are adjacent but not (i,k), now with edge orientation $i \longrightarrow j - k$, an edge direction can be assigned such that $j \longrightarrow k$ while avoiding the creation of cycles or nodes with only inwards directed edges (colliders).

Example

Consider variable set X_D of Figure 3-1 once again, with set $Y_i \subseteq X_D$ as the set of nodes that is adjacent to node $x_i \in X_D$. X_D is assumed to satisfy both the Causal Markov condition and the Faithfulness assumption. The PC algorithm is assumed to perform successfully. The goal of the PC algorithm is to identify the true directed acyclic graph of the variable set without any prior knowledge of causal relations. This is done by performing conditional independence tests on the data set. In this case, X_D is a discrete data set, which means a linear or discrete conditional independence test should be applied. In this section, we will provide proof for these conditional independence relations by following the d-separation criterion. In Figures 3-5 − 3-8 the edge removal steps are shown.

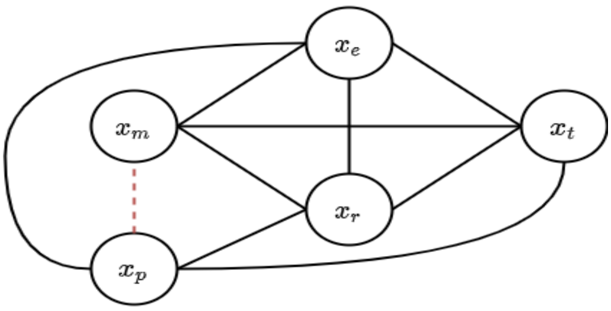

Figure 3-5: Connected graph G' of variable set X_D.

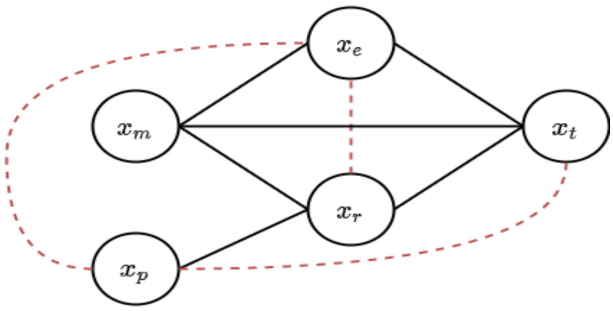

Figure 3-6: Graph G' of variable set X_D after the first edge removal round.

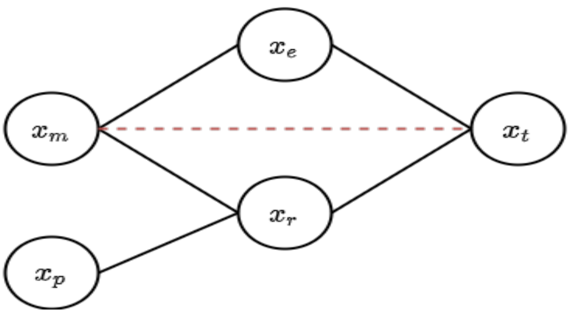

Figure 3-7: Graph G' of variable set X_D after the second edge removal round.

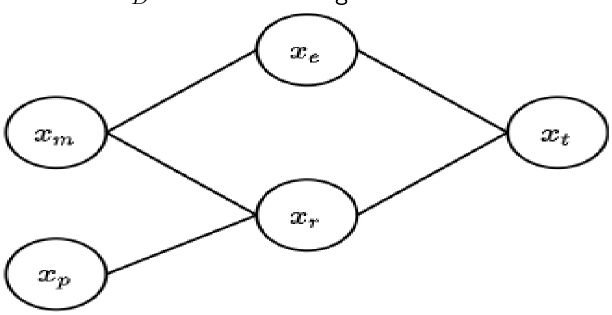

Figure 3-8: Graph G' of variable set X_D after the third edge removal round.

First edge removal step: For the first edge removal step, the zero-order conditional independencies ($n = 0$) are tested by the algorithm, which address the nodes with no incoming edge. When looking at Figure 3-1 it can be seen that the only two nodes with no incoming edges are x_m and x_p, making them both independent $(x_m \perp\!\!\!\perp x_p) = (x_m \perp\!\!\!\perp x_p|\varnothing)$. The algorithm should identify this conditional independence relation, remove the edge connecting both nodes and create separation set $S_{m,p} = \varnothing$ and then update the adjacency sets. The edge to be removed is shown as a dotted red line in Figure 3-5. The updated graph is shown in Figure 3-6.
Second edge removal step: The algorithm then repeats the test for first-order independencies ($n = 1$). The test should identify several conditional independence relations, which we can identify following the d-separation criterion. First, we can see in Figure 3-1 that node x_r blocks all active paths from x_p to the remaining nodes. This indicates the independence relations $(x_p \perp\!\!\!\perp x_e|x_r)$ and $(x_p \perp\!\!\!\perp x_t|x_r)$. Second, there is no active path from node x_r to x_e relative to node x_m. This indicates independence relation $(x_r \perp\!\!\!\perp x_e|x_m)$. After the algorithm identifies these independence relations, it removes the corresponding edges, adds node x_r to subset $S_{p,t}$, node x_r to subset $S_{p,t}$, node x_m subset $S_{r,e}$ and updates the adjacency sets again. The updated graph is shown in Figure 3-7.
Third edge removal step: The algorithm then performs the tests for second-order conditional independencies ($n = 2$). We see that there is no active path from node x_m to node x_t relative to node pair $\{x_e, x_r\}$, which indicates the independence relation $(x_m \perp\!\!\!\perp x_t|(x_e, x_r))$. The algorithm should remove the edge connecting nodes (x_m, x_t), add node set $\{x_e, x_r\}$ to $S_{m,t}$ and update the adjacency sets. The updated graph is shown in Figure 3-8

We see that there are no tests to perform for third-order conditional independencies ($n = 3$). The algorithm should end the edge removal part and determine the undirected graph of figure 3-8 to be the skeleton of the directed acyclic graph G.

Step	Independence	Separation set
1	$(x_m \perp\!\!\!\perp x_p)$	$S_{m,p} = \varnothing$
2	$(x_p \perp\!\!\!\perp x_e|x_r)$	$S_{p,e} = x_r$
2	$(x_p \perp\!\!\!\perp x_t|x_r)$	$S_{p,t} = x_r$
2	$(x_r \perp\!\!\!\perp x_e|x_m)$	$S_{r,e} = x_m$
3	$(x_m \perp\!\!\!\perp x_t|\{x_e, x_r\})$	$S_{m,t} = \{x_e, x_r\}$

Table 3-1: Conditional independence relations in variable set X_D.

Edge direction assignment: Now that the skeleton has been determined, the algorithm starts its second part. This step is concerned with the assignment of directions to the eligible edges. Considering each node triplet (v_i, v_j, v_k) where only node pairs (v_i, v_j) are adjacent and (v_j, v_k) are adjacent, the algorithm assigns edges assigned as $v_i \longrightarrow v_j \longleftarrow v_k$ unless node v_j is contained in subset $S_{i,k}$. From Figure 3-8 we see that this condition is only satisfied for node triplets (x_m, x_r, x_p) and (x_e, x_t, x_r). The graph after the corresponding edge directions are assigned the graph is shown in Figure 3-9.

For the final step of the algorithm each node triplet (v_i, v_j, v_k) where $v_i \longrightarrow v_j$, $v_j - v_k$ and node pair (v_i, v_k) not being adjacent, a directed edge should be drawn so $v_j \longrightarrow v_k$. We can see that both edge directions are possible for node pair (x_m, x_e). This means that after following the d-separation criterion, two estimates of the DAGs are justified graphically.

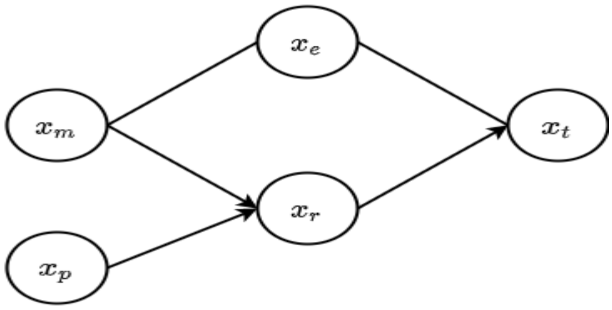

Figure 3-9: Graph G' of variable set X_D after first edge direction.

Applying appropriate conditional independence tests on the data set would hopefully deliver an estimate of the true DAG alone.

3-3 Do-calculus

System parameters can be modified to obtain experimental results, or to learn about the structure or dynamics of the experimental system. Due to practical restrictions or ethical reasons it will not always be possible to change the experiment parameters to obtain desired data. In addition, when it is unknown how the modification of a parameter may affect the system it does not seem very convenient to apply this modification in practice. An external modification where a variable $X_i \in X$ is fixed to a specific observed value x_i' is known as an intervention and is noted as $do(X_i = x_i') = do(x_i')$.

With a causal model, predictions on the effects of variable modifications may be obtained, without actually modifying these variables in practice. This is possible if the causal effect is found to be identifiable.

Causal Effect Identifiability (Corollary 1 [20]): A causal effect $P(y|do(x))$ is identifiable in a model characterised by a graph G if there exists a finite sequence of transformations, each conforming to one of the inference rules in Theorem 3, which reduces $P(y|do(x))$ into a standard probability expression involving only observed quantities.

The causal effect identifiability of $P(y|do(x))$ therefore guarantees that the effect of $do(x)$ onto y can be computed by the use of observational data only and the information provided by causal graph G. This summarizes the concept of do-calculus.

An intervention on X_i removes the incoming causal influences from the model onto X_i and substitutes $X_i = x_i'$ into the rest of the distributions in X that are conditionally dependent of variable X_i. This phenomenon is formulated mathematically as Equation (3.10) [26], which holds as

$$P(x_1, ..., x_n|do(x_i')) = \begin{cases} \Pi_{j \neq i} P(x_j|Pa(x_j)) & \text{if } x_i = x_i', \\ 0 & \text{if } x_i \neq x_i'. \end{cases} \tag{3-1}$$

Confounding variables

When a pair of variables $\{X, Y\}$ maintains a causal relation, while another variable Z has a causal influence on both $\{X, Y\}$, it gets more tedious to predict the causal effect of X on Y. We speak of confounding if a variable has a causal influence on two variables, one being a cause and one being its effect, while it is not on the causal path itself. If we wish to predict the effect onto Y for an intervention on X, we have to 'adjust' our measurements to remove spurious influences from this confounder Z. Adjusting in this case is dividing each separate group relative to Z, estimating the effect of X on Y in each group, and then averaging the results. Pearl constructed a criterion that must be satisfied by the set of variables $Z \subseteq V$ relative to variables (X,Y) for identifiability of $P(y|do(x))$. This is called the back-door criterion.

Back-door criterion (Definition 3.3.1 [20]): The back-door criterion is satisfied by a set of variables Z relative to an ordered pair of variables (X_i, X_j) in a DAG G if

- No node in Z is a descendant of X_i; and
- Z blocks every path between X_i and X_j that contains an arrow into X_i

Since if the only paths towards X_i are blocked then there should be no causal influence on X_i, making it possible to yield an estimate of Y after manipulation of X_i. If the criterion is satisfied by set of variables Z relative to (X,Y), then the causal effect of X on Y is identifiable and is given by

$$P(y|do(x)) = \sum_z P(y|x, z)P(z). \tag{3-2}$$

Pearl incorporated the d-separation criterion to introduce a set of 3 inference rules, known as the rules of do-calculus. These rules are used to write an expression of the distribution of Y for manipulated variable X in terms of observed variables only. This is possible if the causal effect of X onto Y is identifiable. The rules are given in Theorem 3.

The rules of do-calculus (Theorem 3 [20])

Let G be the DAG for V, where X,Y,Z and W are disjoint subsets of V.
Rule 1: Insertion/deletion of observations

$$P(y|do(x), z, w) = P(y|do(x), w) \quad \text{if} \quad (Y \perp\!\!\!\perp Z|X, W)_{G_{\bar{X}}}; \tag{3-3}$$

Rule 2: Action/observation exchange

$$P(y|do(x), do(z), w) = P(y|do(x), z, w) \quad \text{if} \quad (Y \perp\!\!\!\perp Z|X, W)_{G_{\bar{X}\underline{Z}}}; \tag{3-4}$$

Rule 3: Insertion/deletion of actions

$$P(y|do(x), do(z), w) = P(y|do(x), w) \quad \text{if} \quad (Y \perp\!\!\!\perp Z|X, W)_{G_{\bar{X}, \overline{Z(W)}}}. \tag{3-5}$$

The proof of these rules is also given in Section 4.3 of [20]. (Huang and Valtorta) [21] proved the completeness of do-calculus. This means that if a causal effect is identifiable, then the inference rules can be applied to convert the causal effect formula into one that only contains observational quantities.

First example

Consider a highly complex robotic feed distributing system. Throughout its processes, it has collected data for the farm owner to analyze. Data set X_R contains the measurements of all the relevant system parameters and whether a system failure has occurred or not. This data set satisfies both the Causal Markov Condition and the Faithfulness assumption. The directed acyclic graph G_R of X_R is given in Figure 3-10.

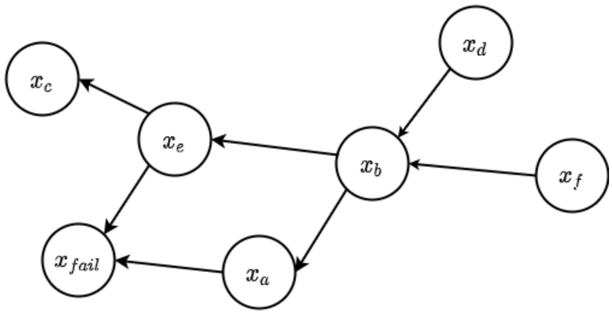

Figure 3-10: Directed graph G_R of variable set X_R.

Following Equation Eq. (A-6) the joint probability distribution function of X_R is given as

$$\begin{aligned} P(X_R) &= P(x_b, x_a, x_e, x_{fail}, x_c, x_d, x_f) \\ &= P(x_d)P(x_f)P(x_b|x_d, x_f)P(x_e|x_b)P(x_a|x_b)P(x_c|x_e)P(x_{fail}|x_e, x_a) \end{aligned}$$

In this case it is desired to know $P(x_{fail} = 1|do(x_b = t))$, which is the probability of a system failure ($x_{fail} = 1$) in case system parameter x_b were to be fixed to t. Variable x_b has been observed to take on this value. Following intervention $do(x_b = t)$, the edges going into x_b are removed. As the value of x_b is fixed, the values of x_d and x_f are no longer relevant for x_b and therefore any causal influence going into x_b is no longer considered for this case. This means that $P(do(x_b = t)|x_d, x_f) = P(do(x_b = t))$. In addition, $P(do(x_b = t)) = P(x_b = t)$, which is a function made up of observational data only. The corresponding post-interventional directed acyclic graph is given in Figure 3-11.

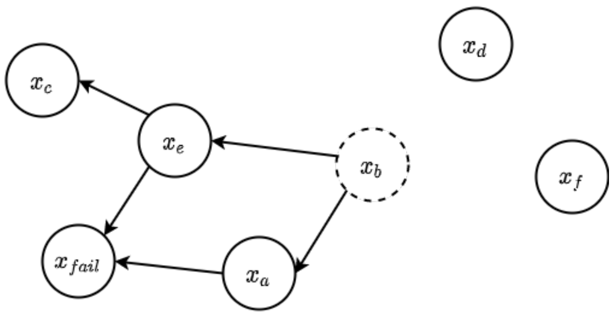

Figure 3-11: Directed graph G_R of variable set X_R after intervention $do(x_b = t)$.

Following Equation 3-1, the post-interventional joint probability fuction is now given as

$$\begin{aligned} P(do(x_b = t), x_a, x_e, x_{fail}, x_c) = P(x_b)P(x_e|x_b)P(x_a|x_b) \\ P(x_c|x_e)P(x_{fail}|x_e, x_a) \end{aligned}$$

x_c does not have any causal influence on x_{fail} is left out of the equation when computing $P(x_{fail} = 1|do(x_b = t))$. Using the information given and following Equation A-5 we find

$$\begin{aligned} P(x_{fail} = 1|do(x_b = t)) &= \frac{P(do(x_b = t), x_a, x_e, x_{fail} = 1)}{P(do(x_b = t))} \\ &= \frac{P(do(x_b = t))P(x_a|do(x_b = t))P(x_e|do(x_b = t))P(x_{fail} = 1|x_a, x_e)}{P(do(x_b = t))} \\ &= P(x_a|do(x_b = t)P(x_e|do(x_b = t))P(x_{fail} = 1|x_a, x_e) \\ &= P(x_a|x_b = t)P(x_e|x_b = t)P(x_{fail} = 1|x_a, x_e) \end{aligned}$$

We see that we can give the post-interventional distribution function of $x_{fail} = 1$ in strictly observational terms. This means that the causal effect is identifiable.

The application of the backdoor adjustment Equation 3-6 will be explained in the second example. This is done on an example subset V_{bd} given in Figure 3-12.

Second example

Consider variable set V_{bd} of which graph G_{bd} is given in Figure 3-12. It is desired to predict the effect of intervention $do(v_{OX} = t)$ onto v_Q.

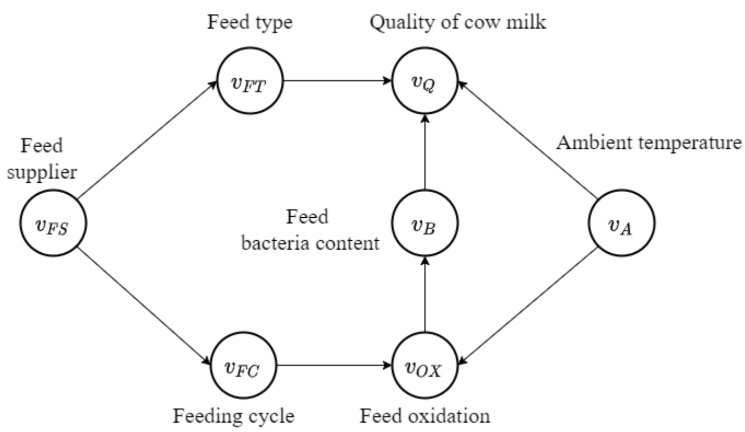

Figure 3-12: Directed graph G_{bd} of variable set V_{bd}.

For a consistent prediction of this causal effect to be made, an adjustment should be made for confounding variables v_{FS} and v_A. This adjustment is the backdoor adjustment and can be made since

- neither of node pair $\{v_{FS}, v_A\}$ are descendants of v_{OX},
- node pair $\{v_{FS}, v_A\}$ block every path between v_{OX} and v_Q that contain an arrow into v_{OX}.

When v_{OX} is set to value t after intervention $do(v_{OX} = t)$, the incoming edges from v_{FC} and v_A, being all backdoor paths, are removed. The graph after intervention is shown in Figure 3-13.

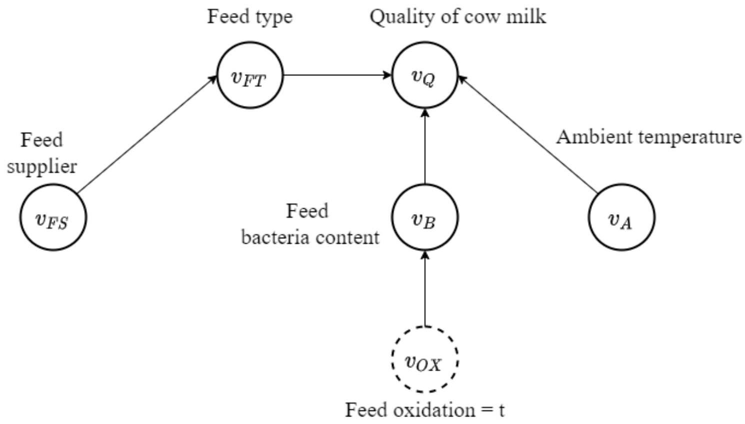

Figure 3-13: Directed graph G_{bd} of variable set V_{bd} after intervention $do(v_{OX} = t)$.

Now observation $v_{OX} = t$ can not be differentiated from intervention $do(v_{OX} = t)$ since the incoming causal influences are removed and $P(v_Q|do(v_{OX} = t))$ can be computed in terms of observational data only. When adjusting for v_{FT} the causal influence of v_{FS} onto v_Q is removed by d-separation. We therefore adjust for v_{FT} and v_A to remove all spurious confounding influences.

Following Equation 3-2, the probability distribution of v_Q for intervention $do(v_{OX} = t)$ is given as

$$\begin{aligned} P(v_Q|do(v_{OX} = t)) &= \sum_{v_A} \sum_{v_{FT}} P(v_B|do(v_{OX} = t))P(v_A)P(v_{FT})P(v_Q|v_{FT}, v_B, v_A) \\ &= \sum_{v_A} \sum_{v_{FT}} P(v_B|v_{OX} = t)P(v_A)P(v_{FT})P(v_Q|v_{FT}, v_B, v_A). \end{aligned} \tag{3-6}$$

We have discussed how the observational data set is required to satisfy both the Causal Markov Condition and the Faithfulness assumption. In addition, we have discussed predictions on the effects of variable interventions and how the causal effect should be identifiable to do so. We will apply the rules of do-calculus to make predictions on job duration values and use these predictions as additional inputs for the β-robust scheduling optimization.

Chapter 4

Sensitivity analysis

Sensitivity analysis of the optimal schedule is the process where the preservation of optimality of this schedule is investigated for perturbations of the numerical input data. Sensitivity analysis is also known as stability analysis. Measurement data contains noise which means its accuracy is limited. Inaccuracies in data can be modelled as uncertainties and when occurring, the solver is supplied with inaccurate input data. Given that a perturbation in measurement data has occurred, a probability exists that the solver finds an optimal schedule for this input data, which then is not the optimal schedule in reality. If this is the case, the reliability of the scheduler may be harmed. After a job schedule has been determined to be optimal given the input parameters, they could verify whether the schedule will remain optimal for a certain perturbed input vector μ'. This can be done by following the approach of Sotskov et al. [22], where the stability radius ρ of the optimal job schedule is determined. This stability radius encloses the maximum allowed input parameter perturbation for which optimality of the schedule is preserved.

4-1 Stability radius

Consider set of jobs $J \in \{J_1, .., J_n\}$ for general one-machine problem. Under nominal behaviour, meaning input perturbed input vector $\mu' = \mu$, the optimal objective function is given as

$$\Phi(x^*) = \frac{\sum_{i=1}^{n}\sum_{k=1}^{n}(n-k+1)\mu_i x^*_{ik} - T}{\sqrt{\sum_{i=1}^{n}\sum_{k=1}^{n}(n-k+1)^2\sigma_i^2 x^*_{ik}}}. \tag{4-1}$$

Let X be the set of all feasible schedules. Schedule x^* is optimal if

$$\Phi(x^*) = \max_{x \in X} \Phi(x) \tag{4-2}$$

With optimal schedule x^*, the next step is to determine the maximum parameter perturbation of μ such that the optimality of schedule x^* is preserved. Sotskov developed his approach for a multi-machine environment where jobs consist of multiple processes. Since we only consider a one-machine job environment, the inclusion of his research may seem redundant. However, we want to incentivize the reader to investigate the improvement of a robust job scheduling method by applying causal inference. Our approach can easily be extended for multi-machine job environments with the remaining of Sotskov's research. Therefore, we see no reason to exclude Sotskov's approach. In this section, we follow Sotskov's definition and explanation of the stability ball and radius of Section 2 [27].

Stability ball (Definition 1 [22]): The closed ball $O_\rho(\mu)$ with the radius $\rho \in R$ and center $\mu \in R^n_+$ is known as the stability ball O_ρ of schedule x^* if for any vector $\mu' \in O_\rho(\mu) \cap R^n_+$ of the job duration means, schedule x^* remains optimal. The minimum value $\rho(\mu)$ of such a radius ρ of a stability ball $O_\rho(\mu)$ of schedule x^* is called the stability radius of x^*:

$$\rho(\mu) = \min\{\rho \in R^1_+ : \text{If} \quad \mu' \in O_\rho(\mu) \cap R^n_+, \quad \text{schedule } x^* \text{ is optimal}\}. \tag{4-3}$$

The stability radius is enclosed in a maximum parameter perturbation space R^n_+. It is desired to have a stability radius that is large enough so the influence of approximation errors and minor random errors on the optimality of the schedule can be ruled out. The maximum parameter perturbation space should be determined to verify this can be done.

4-2 Problem formulation

Decision variable

- Perturbed job duration mean: $\mu' \in R^n$

Objective function

The stability radius can be found by solving the following maximization function

$$\max_{\mu'} \qquad |\mu'_i| \tag{4-4}$$

Constraint

$$\text{subject to} \qquad x^*(\mu) = x^*(\mu'). \tag{4-5}$$

The solution to Equation (4-4) is given as $\theta \in R^n$. The entry corresponding to the smallest value of allowed perturbation gives the stability radius $\rho(\mu)$, which is done by following

$$\rho(\mu) = \min_i |\theta_i - \mu_i|. \tag{4-6}$$

A large value of $\rho(\mu)$ is desired as this guarantees the schedule to remain optimal for large perturbations of μ. If the optimal schedule has a maximum allowed perturbation $\rho(\mu) < \min_i \alpha\mu_i$ with $\alpha \in \mathbb{R}^+$ being a very small non-negative value, the schedule is not a well-performing solution in terms of robustness. There is a probability that the schedule is not optimal in practice. This scenario makes the application of causal inference extra useful as this would supply the β-robust scheduling optimization with an additional set of input values. If a post-interventional solution's stability radius is predicted to be larger than the initial stability radius, then the scheduler could consider applying the corresponding intervention. Doing so would safeguard the optimality of the solution and thus the reliability of the scheduler.

Chapter 5

Methodology

5-1 Proposed methodology

We will solve a one-machine scheduling problem for nominal and post-interventional scenarios. We have generated the input values μ and σ^2 synthetically, where the input values of one job are a function of the variables of a synthetic data set V_1. We will exploit this data set for predictions by applying the tools of causal inference. We will apply the stable PC algorithm on the data set with the aim of identifying the causal model of V_1. Predictions on the effects of interventions will be made based on this model. These predictions will be used as post-interventional input values for an optimization problem which will be a maximization of the trade-off between the improvement on job schedule performance Φ and stability radius ρ.

5-2 Problem formulation

For our problem we consider

- a one-machine scheduling environment,
- n independent jobs,
- 1 data set V_i $(i = 1, ..., n)$ of job J_i eligible for causal inference,
- only single interventions are considered.

Input variables

Job J_i's duration mean μ_i is now a function of the variable set V_i, if given. It is the first input variable and is defined as

$$\mu_i(V_i) = \begin{cases} \mu_i, & \text{if} \quad V_{i,I} = \varnothing \\ \frac{\sum\limits_y P(y|do(v))y}{\dim(y)}, & \text{if} \quad V_{i,I} \neq \varnothing \end{cases} \tag{5-1}$$

$V_{i,I} \subset V_i$ is the intervention set and it holds that

- $y \in V_i$ corresponds to the job duration value and $v \in V_{i,I}$ to the intervention variable,
- $v \in V_{i,I} \neq \varnothing$ if the causal identifiability condition is satisfied for variable v onto y in set V_i. $V_{i,I} = \varnothing$ otherwise,
- $P(y|do(v))$ is computed by applying the rules of probability calculus and do-calculus, in case predictions on the effects of an intervention can be made for data set V_i.

The second input variable, being the variance of the job duration $\sigma_i^2(V_i)$, is given as

$$\sigma_i^2(V_i) = \begin{cases} \sigma_i^2, & \text{if} \quad V_{i,I} = \varnothing \\ \frac{\sum_{j=1}^{N} (y_j(V_i) - \mu_i(V_i))^2}{N}, & \text{if} \quad V_{i,I} \neq \varnothing^* \end{cases} \tag{5-2}$$

$y_j(V_i)$ $(j = 1, ..., N)$ is the j-th sample of the post-interventional job duration value which is generated synthetically following the post-interventional probability distributions.

Decision variables

- Assignment variable: $x_{ik} \in \{0, 1\} : i = 1, 2, ..., n$ and $k = 1, 2, ..., n$,
- Variable set: V_i.

Objective function

$$\max_{x, V_i} \gamma(\Phi_I(x, V_i) - \Phi) + (1 - \gamma)(\rho_I(\mu(V_i)) - \rho), \qquad \gamma \in [0, 1]. \tag{5-3}$$

Constraints

$$\text{subject to} \qquad \sum_{k=1}^{n} x_{ik} = 1, \quad i = 1, 2, ..., n, \tag{5-4}$$

$$\sum_{i=1}^{n} x_{ik} = 1, \quad k = 1, 2, ..., n, \tag{5-5}$$

$$x_{ik} \in \{0, 1\}, \quad i = 1, 2, ..., n, k = 1, 2, ..., n. \tag{5-6}$$

In this objective function we have

- $\gamma \in [0, 1]$: optimization weight,
- Φ: nominal objective value,
- ρ: nominal stability radius,
- $\Phi_I(x, V_i)$: nominal or post-interventional objective value.
- $\rho_I(\mu(V_i))$: nominal or post-interventional stability radius.

The goal is to maximize the effect of an intervention on a trade-off between the objective value and stability radius. The new objective function value is $\Phi_N(x, V_i)$. If we prefer an increase of performance of objective value, then we should choose γ larger than 0.5. If we only wish a maximized job schedule stability radius, then we should choose γ as 0. We have $\Phi_N(x, V_i) = 0$ for $V_{i,I} = \varnothing$.

Post-interventional objective value $\Phi_I(x, V_i)$ and stability radius $\rho_I(\mu(V_i))$ are given as

$$\Phi_I(x, V_i) = \frac{T - \sum_{i,k}^{n}(n-k+1)\mu_i(V_i)x_{ik}}{\sqrt{\sum_{i,k}^{n}(n-k+1)^2\sigma_i^2(V_i)x_{ik}}}, \tag{5-7}$$

$$\rho_I(\mu(V_i)) = \min_i |\theta_i(\mu_i(V_i)) - \mu_i(V_i)|, \tag{5-8}$$

where $\theta_i(\mu(V_i))$ is the solution to the following modified sensitivity analysis maximization problem

$$\max_{\mu'} \quad |\mu_i'| \tag{5-9}$$

$$\text{subject to} \quad x^*(\mu) = x^*(\mu'). \tag{5-10}$$

5-3 Simulation approach

We treat the β-robust optimization problem as a mixed-integer nonlinear programming problem (MINLP) and will solve it with the use of Matlab. Yalmip's *bmibnb* will be applied as solver, which uses a branch-and-bound approach for nonconvex problems. It consists of a lower bound solver for a convex relaxation of the problem and an upper bound for the original nonlinear problem. In our case, we apply *fmincon* as upper bound solver and *Gurobi* as lower bound solver. *fmincon* is a local nonlinear solver. The size of our scheduling example is chosen such that we are able to verify that our optimization algorithm successfully solves our problem in the root node within reasonable run time. We verify in our simulations that this is the case for the branch-and-bound optimization algorithm *bmibnb*. Therefore, we choose it as optimization algorithm for the β-robust modeling approach.

We treat the sensitivity analysis as a nonlinear problem (NLP) due to the nonlinear constraint. We will solve this problem with *Gurobi* in Matlab. This state-of-the-art solver successfully solves nonlinear constrained problems.

We will use the stable PC algorithm from Rstudio's *bnlearn* package as causal discovery method on the synthetic data set for the causal inference part. This algorithm is an improved version of the PC algorithm, which does not necessarily perform conditional independence tests in an ascending order. This algorithm is the standard version of the PC algorithm from the *bnlearn* package. The mutual information test will be used as (conditional) independence test, which is a conditional independence test that can be applied to discrete variable sets. We will use a score-based causal discovery method from the *bnlearn* package to verify our DAG estimate, which is the Grow-Shrink method.

5-4 Artificial data set

We will test the proposed method on a 5 job data set. 4 of these job's input parameters $\{(\mu_2, \sigma_2^2), ..., (\mu_5, \sigma_5^5)\}$ are generated in the same manner as in Chapter 5 of [1]. μ_1 and σ_1^2 are the product of data set V_1 which has been generated in Rstudio. For the generation of this data we have used R's *sample* function which creates samples by running a random number generator for a set of given probability distributions that satisfy Equations A-1 – A-2 – A-3.

μ_1 and σ_1^2 satisfy the same bounds as used for the generation of the remaining job input parameters. Variable information of data set V_1 can be found in the Appendix. The post-interventional data sets have been generated in the same manner, but with a different seed for the random number generator. In Section A-3 of the Appendix we show how μ_1 and $\mu_1(V_1)$ are computed.

Chapter 6

Results

It is given that out of all available data, data set $V_1 \in \mathbb{R}^{7x1000}$ of job J_1 is complete and sufficient for analysis of causal relations. Data set $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7\} \in V_1$, with variable $v_6 \in \{14, 15, 16, 17, 18, 20, 23, 24\}$ being the observed job duration values. This data set is generated by running *datacreator.R*. The observed values and their observation count of the job duration variable $v_6 \in V_1$ can be seen in Table 6-1. The remaining variable observational values can be found in Table A-2 in the Appendix.

v_6	14	15	16	17	18	20	23	24
n	2439	1573	1376	808	711	704	1311	1078

Table 6-1: Observed job duration values and their count.

6-1 Nominal optimization

The nominal input values are given in Table 6-2.

Job	J_1	J_2	J_3	J_4	J_5
μ	17.64	28.01	10.00	17.56	13.67
σ^2	13.26	16.23	3.84	13.59	11.19

Table 6-2: Nominal input values.

The goal is to maximize the probability that the total flow time does not exceed a given threshold $T = 250$. We note the set of all feasible schedules as X. Given this information, we compute the β-robust schedule by solving the maximization function Equation 2-6, with the corresponding objective value given as $\Phi(x^*)$. Matlab's *bmibnb* solver finds the beta-robust optimal solution to be

$$x^* = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

$$\{J_3, J_5, J_4, J_1, J_2\}$$

The performance variable values of this sequence are given as

	$\bar{\phi}(x)$	$\sigma^2(x)$	$\rho(\mu)$	$\Phi(x)$	$P(\phi(x) \leq T)$
x^*	220.65	466.61	0.02	1.359	0.913

Table 6-3: Nominal optimization performance variable values.

6-2 Causal inference

As this data set is generated by ourselves, we assume both the Faithfulness assumption and the Causal Markov condition to hold. In addition, we assume that no unobserved confounding variables are present. This means that the PC algorithm can be applied to the data set, with the aim to identify the underlying DAG. We have done this in *Rstudio* with the *bnlearn* toolbox and we've chosen the mutual information test as conditional independence test. This is a statistical hypothesis test that we've used to verify whether the distributions of separate discrete sets of variables are independent of each other.

Causal discovery

Our estimated DAG of data set V_1 is shown in Figure 6-1.

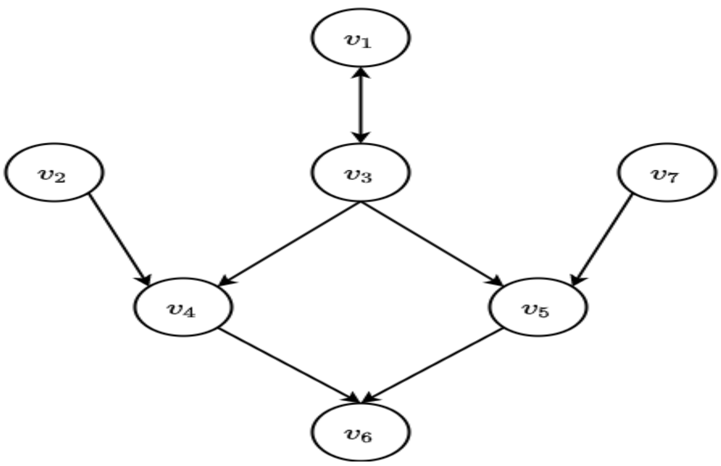

Figure 6-1: Directed graph of variable set V_1 estimated with Rstudio's *bnlearn* package.

We see that a bi-directed edge between nodes v_1 and v_3 has been determined. This means that two separate DAGs (G_1 and G_2) have been estimated by the PC algorithm. The algorithm cannot distinguish between these models, based on the given observational data. The difference between both DAGs is the edge orientation between nodes v_1 and v_3. Graph G_1 is considered as the graph with edge orientation $v_1 \rightarrow v_3$, and G_2 for $v_1 \leftarrow v_3$. Both graphs consist of the same set of conditional independence relations since by d-separation we have $v_1 \perp\!\!\!\perp v_4|v_3$ and $v_1 \perp\!\!\!\perp v_5|v_3$ for both graphs. Therefore, one graph is a Markov Equivalent and the other is the true DAG of variable set V_1.

Having these two different estimates is not ideal, since we don't know if we can exclude an intervention on v_1. If G_2 is the true DAG, then applying an intervention on v_1 will not influence the job duration time v_6 and then doing so would be an unnecessary action. If we cannot permit ourselves to perform unnecessary actions, we should exclude v_1 as an intervention variable in our optimization. On the other hand, excluding v_1 would be a waste of opportunity to improve our job scheduling process if G_1 is the true DAG. This scenario is not what we desired. However, we can still investigate interventions on the remaining variables.

In Table 6-4 the conditional probabilities of the variables are given for both DAG estimates.

x	$Pa_1(x)$	$P(x\|Pa_1(x))$	$Pa_2(x)$	$P(x\|Pa_2(x))$
v_1	$\varnothing$	$P(v_1)$	v_3	$P(v_1\|v_3)$
v_2	$\varnothing$	$P(v_2)$	$\varnothing$	$P(v_2)$
v_3	v_1	$P(v_3\|v_1)$	$\varnothing$	$P(v_3)$
v_4	$\{v_2, v_3\}$	$P(v_4\|v_2, v_3)$	$\{v_2, v_3\}$	$P(v_4\|v_2, v_3)$
v_5	$\{v_3, v_7\}$	$P(v_5\|v_3, v_7)$	$\{v_3, v_7\}$	$P(v_5\|v_3, v_7)$
v_6	$\{v_4, v_5\}$	$P(v_6\|v_4, v_5)$	$\{v_4, v_5\}$	$P(v_6\|v_4, v_5)$
v_7	$\varnothing$	$P(v_7)$	$\varnothing$	$P(v_7)$

Table 6-4: Conditional probabilities of variables of directed graphs G_1 and G_2.

We've computed the probability values by dividing the observed values by their corresponding observation count. Next, we've derived the conditional probabilities $P(x|Pa(x))$ by following Equation A-5 and following the independence rule (Equation A-7). For example, when determining $P(v_4|v_2, v_3)$ for graph G_1 we first incorporate the following:

- by the Causal Markov Condition v_2 and v_3 are independent when conditioning on v_4 which means $v_2 \perp\!\!\!\perp v_3|v_4$ and $P(v_2, v_3) = P(v_2)P(v_3)$,
- by d-separation it follows that $v_4 \perp\!\!\!\perp v_1|v_3$ and therefore the edge going into v_1 can be left out.

These conclusions and Equation A-5 allow us to compute $P(v_4|v_2, v_3)$ as

$$\begin{aligned} P(v_4|v_2, v_3) &= \frac{P(v_4, v_2, v_3)}{P(v_2, v_3)} \\ &= \frac{P(v_4, v_2, v_3)}{P(v_2)P(v_3)} \end{aligned}$$

For graphs G_1 and G_2, we follow Equation A-6 and find two joint probability distribution functions $P_1(x)$ and $P_2(x)$ respectively as

$$P_1(x) = P(v_1)P(v_2)P(v_7)P(v_3|v_1)P(v_5|v_7, v_3)P(v_4|v_2, v_3)P(v_6|v_4, v_5) \tag{6-1}$$
$$P_2(x) = P(v_3)P(v_2)P(v_7)P(v_1|v_3)P(v_5|v_7, v_3)P(v_4|v_2, v_3)P(v_6|v_4, v_5) \tag{6-2}$$

Do-calculus

We see that for v_4, both v_3 and v_5 satisfy the backdoor criterion, as both nodes block each other's backdoor path to node v_6. For v_5, both v_4 and v_3 satisfy the backdoor criterion. This means that the causal effects of both v_4 and v_5 onto v_6 are identifiable. We can obtain an unbiased prediction of v_6 for both interventions on v_4 and v_5, which we can use to generate the inputs for the β-robust scheduling method.

For demonstration purposes, we consider the effects on v_6 for an intervention on variables v_4 and v_5 only and $\{v_4, v_5\} \in V_{1,I}$. As is given in Table A-2 in the Appendix, the set of observational values of v_4, v_5 and v_6 are

- $v_4 : \{1, 2, 3, 4, 5, 6, 7, 8\} = \{v_{41}, v_{42}, v_{43}, v_{44}, v_{45}, v_{46}, v_{47}, v_{48}\}$,
- $v_5 : \{60, 62.5, 65, 67.5, 70, 72.5, 75, 77.5\} = \{v_{51}, v_{52}, v_{53}, v_{54}, v_{55}, v_{56}, v_{57}, v_{58}\}$,
- $v_6 : \{14, 15, 16, 17, 18, 20, 23, 24\} = \{v_{61}, v_{62}, v_{63}, v_{64}, v_{65}, v_{66}, v_{67}, v_{68}\}$,

We can predict the effects of an intervention on v_4 and v_5 for both 8 different values each. We use each of these variable values as intervention values and thus for the prediction of a post-interventional probability distribution of v_6 following Equation 3-2. We calculate these post-interventional probability values by applying the backdoor adjustment Eq. (3-2), which results in the following equations

$$P(v_6|do(v_4)) = \sum_{v_5} P(v_6|v_4, v_5)P(v_5), \tag{6-3}$$
$$P(v_6|do(v_5)) = \sum_{v_4} P(v_6|v_4, v_5)P(v_4). \tag{6-4}$$

The post-interventional probability distributions of v_6 for interventions $do(v_4)$ and $do(v_5)$ are given in Table A-3 in the Appendix.

Post-interventional optimization

We compute the β-robust optimization input variables by substituting Equations 6-3 and 6-4 into Equation 5-1 for $\mu_1(V_1)$ and following Equation 5-2 for $\sigma_1^2(V_1)$.

We see several scenarios for both interventions for which we predict the schedule to outperform the nominal optimal schedule. The post-interventional objective value $\Phi_I(x_I^*(V_1))$ is the highest for intervention $do(v_{51})$, which means $\max(\max(\Phi_{IS1}), \max(\Phi_{IS2})) = \Phi_{I9}$.

$v_4 \in V_1$	$do(v_{41})$	$do(v_{42})$	$do(v_{43})$	$do(v_{44})$	$do(v_{45})$	$do(v_{46})$	$do(v_{47})$	$do(v_{48})$
$\mu_1(V_1)$	17.47	17.60	17.60	17.43	17.62	17.77	18.00	18.11
$\sigma_1^2(V_1)$	12.24	12.59	13.13	12.92	13.75	13.31	14,73	13.35
$\Phi_I(x_I^*, V_1)$	1.394	1.371	1.365	1.391	1.358	1.346	1.317	1.304
$v_5 \in V_1$	$do(v_{51})$	$do(v_{52})$	$do(v_{53})$	$do(v_{54})$	$do(v_{55})$	$do(v_{56})$	$do(v_{57})$	$do(v_{58})$
$\mu_1(V_1)$	16.88	17.08	17.46	17.55	18.05	18.30	18.46	19.05
$\sigma_1^2(V_1)$	12.24	12.59	13.13	12.92	13.75	13.31	14.73	15.35
$\Phi_I(x_I^*, V_1)$	**1.505**	1.458	1.391	1.375	1.316	1.290	1.272	1.218

Table 6-5: Input and objective values after interventions on v_4 and v_5.

6-3 Effect of changing parameter T on objective value

In the previous section, we have concluded that we predict a job schedule in a post-interventional scenario to outperform the schedule for the nominal situation for target performance $T = 250$ in terms of objective value. We will now analyze the maximum predicted objective value after intervention for different values of performance target $T \in [210, 250]$. For each T $\in [210, 250]$, we compute the sets Φ_{IS1} and Φ_{IS2} again.

The maximum value of sets Φ_{IS1} and Φ_{IS2} are shown in Figure 6-2 for each T against the nominal optimal objective value $\Phi(x^*)$.

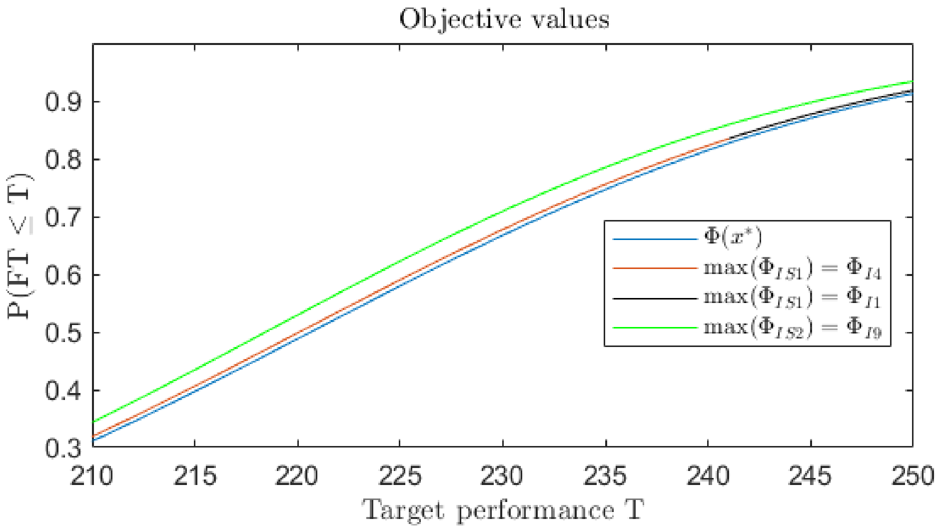

Figure 6-2: Cumulative distribution function values of Φ for nominal and post-interventional scenarios for different T.

We see that for any $T \in [210, 250]$, post-interventional schedule performance Φ_{I9} is higher than for any post-interventional performance value in Φ_{IS1}. Φ_{I9} is also higher than for any other value in Φ_{IS2} for each $T \in [210, 250]$, which is remarkable. We also see that for $T \in [210, 241]$, the optimal schedule performance in Φ_{IS1} is for intervention $do(v_{44})$. For $T \in [242, 250]$ the optimal schedule performance in Φ_{IS1} is for intervention $do(v_{48})$.

We can conclude that we predict a schedule for a post-interventional scenario to outperform the schedule under the nominal circumstances for each $T \in [210, 250]$.

6-4 Sensitivity analysis

Nominal optimization

We look further at our optimization problem for $T = 250$. We have $\Phi(x^*) = 1.32$ and compute $\rho(\mu) = 0.02$ corresponding to job J_4 by following Equations 4-4 and 4-6.

Post-interventional optimization

The set of post-interventional stability radii is found by solving Equations 5-9 and 5-10 and is given in Table 6-6. The job tolerating the lowest amount of delay, therefore responsible for the stability radius is also given. For intervention $do(v_4)$ we get

- $\rho_{Ii} = \rho_I(\mu_{1i}(V_1)), \qquad \text{for } i = 1, 2, ..., 8,$
- $\rho_{IS1} = \{\rho_{I1}, \rho_{I2}, ..., \rho_{I8}\}.$

$v_4 \in V_1$	$do(v_{41})$	$do(v_{42})$	$do(v_{43})$	$do(v_{44})$	$do(v_{45})$	$do(v_{46})$	$do(v_{47})$	$do(v_{48})$
$\rho_I(\mu_1(V_1))$	0.29	0.11	0.03	0.23	0.08	0.16	0.60	**0.78**
J_ρ	1	1	1	1	4	4	4	4

Table 6-6: Post-interventional stability radii for interventions on v_4.

We see that the largest stability radius is $\max(\rho_{IS1}) = \rho_{I8}$. μ_4 of job J_4 is allowed to encounter a delay of up to $\delta_4 = \mu_4' - \mu_4 = 0.78$ for its schedule to remain optimal. Each of the objective values and corresponding stability radii are plotted in Figure 6-3.

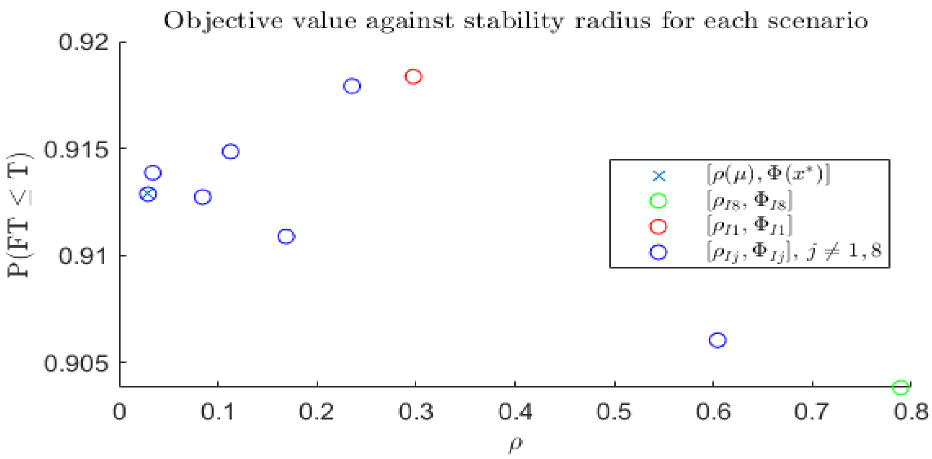

Figure 6-3: Cumulative distribution function values of Φ shown against the stability radius ρ for nominal scenario and for interventions on v_4.

We see that objective value Φ_{I8} is the lowest of all objective values. Intervention $do(v_{48})$ performs optimally for both characteristics and this means we should determine which schedule characteristic is most important to us.

To verify the stability radius has been computed correctly, we compare the schedule performances $\Phi(x^*)$ and Φ_{I8} for perturbations δ_4 of job mean μ_4. For both scenarios, the job corresponding to the stability radius is job J_4. We've given the delay δ as a percentage of its job duration mean. This is shown in Figure 6-4, where we have

- $\Phi_F = \{\Phi(x') \mid x' : x' \in X \backslash \{x^*\}\}$,
- $\Phi_{I8F} = \{\Phi_{I8}(x') \mid x' : x' \in X \backslash \{x_I^*\}\}$.

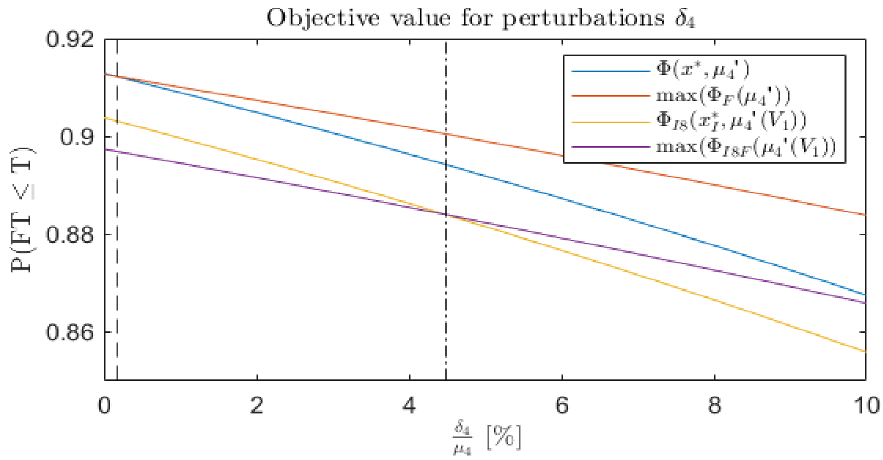

Figure 6-4: Cumulative distribution function values of perturbed Φ for nominal scenario and for intervention $do(v_{48})$.

We see that the optimality of the schedules is lost when the perturbation equals the stability radius of the schedule. Therefore, we've performed the sensitivity analysis correctly.

For intervention $do(v_5)$ we have ρ_{IS2}

- $\rho_{IS2} = \{\rho_{I9}, \rho_{I10}, ..., \rho_{I16}\}$.

In Table 6-7 the values are given and in Figure 6-5 the objective values and corresponding stability radii are plotted for each scenario.

$v_5 \in V_1$	$do(v_{51})$	$do(v_{52})$	$do(v_{53})$	$do(v_{54})$	$do(v_{55})$	$do(v_{56})$	$do(v_{57})$	$do(v_{58})$
$\rho_I(\mu_1(V_1))$	1.18	0.78	0.24	0.11	0.57	0.87	1.11	**1.67**
J_ρ	1	1	1	1	4	4	4	4

Table 6-7: Post-interventional stability radii for interventions on v_5.

We see that the post-interventional objective value and stability radius of the optimal schedule for intervention $do(v_5)$ are significantly higher than for intervention $do(v_4)$. Therefore, we should only consider an intervention $do(v_5)$.

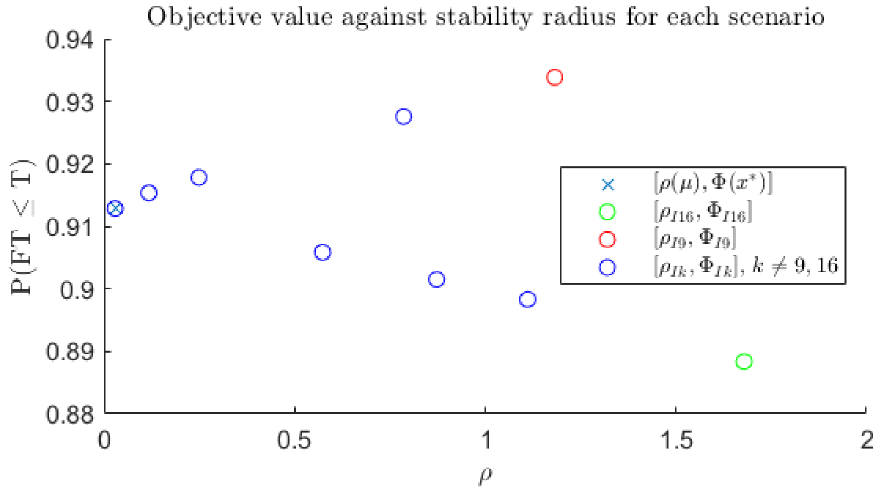

Figure 6-5: Cumulative distribution function values of Φ shown against the stability radius ρ for nominal scenario and for interventions on v_5.

6-5 New objective function values

We consider the objective function Equation 5-3. When looking at nominal values $\Phi(x^*) = 1.359$ and $\rho(\mu) = 0.02$, we conclude that the nominal schedule for the nominal situation is too sensitive. A nearly negligible delay of job duration mean μ_4 will result in the found schedule to no longer being optimal. This motivates us to focus on improving the the stability radius for our new optimization and therefore choose $\gamma = 0.3$ as optimization weight.

Numeric results

In Table 6-8, the new objective values Φ_N are given along with the corresponding post-interventional objective values Φ_I and stability radii ρ_I.

$v_4 \in V_1$	$do(v_{41})$	$do(v_{42})$	$do(v_{43})$	$do(v_{44})$	$do(v_{45})$	$do(v_{46})$	$do(v_{47})$	$do(v_{48})$
$\Phi_I(x_I^*, V_1)$	1.394	1.371	1.365	1.391	1.358	1.346	1.317	1.304
$\rho_I(\mu(V_1))$	0.29	0.11	0.03	0.23	0.08	0.16	0.60	0.78
$\Phi_N(x_I^*, V_1)$	0.189	0.059	0.004	0.146	0.039	0.097	0.400	0.529
$v_5 \in V_1$	$do(v_{51})$	$do(v_{52})$	$do(v_{53})$	$do(v_{54})$	$do(v_{55})$	$do(v_{56})$	$do(v_{57})$	$do(v_{58})$
$\Phi_I(x_I^*, V_1)$	**1.505**	1.458	1.391	1.375	1.316	1.290	1.272	1.2184
$\rho_I(\mu(V_1))$	1.18	0.78	0.24	0.11	0.57	0.87	1.11	**1.67**
$\Phi_N(x_I^*, V_1)$	0.813	0.534	0.155	0.063	0.379	0.587	0.753	**1.147**

Table 6-8: Performance values after interventions on v_4 and v_5.

We predict for $\gamma = 0.3$, which implies a preference on improving the stability radius, the optimal scenario to be the post-interventional scenario for $do(v_{58})$. The post-interventional objective value for this scenario is lower than that for the nominal situation, meaning the schedule performs worse. However, we predict the increase in stability radius to be significant enough such that this scenario corresponds to the optimal schedule.

Chapter 7

Conclusion

7-1 Conclusion

In this thesis, we have proposed a robust scheduling algorithm for a one-machine scheduling environment. Our algorithm is an extension to the β-robust scheduling method and applies causal inference on a given job data set and solution sensitivity analysis. The requirement for the observational job data set is that it's eligible for causal inference. The addition of sensitivity analysis has allowed us to include the stability radius as an additional performance measure for the optimization. We have discussed the practical use of both additions and several causal discovery methods in the literature review. We have discussed β-robust scheduling and shown its practical use and we have shown the application of causal inference for some toy examples. For the latter, we have provided graphical for the understanding of the reader.

Our demonstration showed how the performance of a synthetic 5-job scheduling problem is predicted to improve after applying certain system parameter modifications. A Markov equivalent and the true causal model of the process have been identified and the models have been exploited for predictions on the effects of variable interventions. We explained how having multiple estimates of the causal graph could be an issue for making predictions. This phenomenon shows that we cannot solve any problem entirely by consulting data. We made predictions by following the rules of do-calculus and probability calculus and used these to provide the β-robust scheduling problem and sensitivity analysis with additional input values. We used these additional inputs for the optimization, which means that we included the predicted scenarios in the decision-making progress. We observed the schedules for several predicted scenarios to outperform the schedule under nominal behaviour for objective value Φ and stability radius ρ. Given this information, we can conclude that our scheduling process can be improved by applying these modifications to the system parameters in practice.

Due to the strong constraints that data used in causal inference is subjected to, we haven't applied our algorithm to an existing scheduling problem. We could not find a robust scheduling problem with an observational data set of a job in a scheduling environment that encloses each relevant event without bias. Therefore, we demonstrated our algorithm on a 5-job scheduling problem with one synthetic data set that we assumed to satisfy the three required conditions. This thesis should therefore be seen as a theoretical approach to improve a job scheduling process. Given that the trend continues, machines and processes will become more advanced and data-driven. Process data will be collected and stored in a manner that makes it resemble the characteristics of the underlying process more accurately. This means that the application of causal inference in robust job scheduling will become more accessible. Therefore, the results presented in our thesis should encourage robust job schedulers to investigate ways to get their process data eligible for causal inference. If eligible, our algorithm can be applied to the scheduling problem with deadline or it can be extended to apply to robust scheduling problems in multi-machine job environments.

7-2 Limitations and recommendations for future work

1. Both causal discovery methods from the *bnlearn* package estimated the same two graphs that differed in one edge orientation. Both algorithms cannot distinguish between graphs G_1 and G_2 based on the given data set and the performed conditional independence tests. Increasing the number of data samples to $n = 100000$ did not result in the algorithms only finding the true DAG. This is an interesting limitation in causal inference. Solely analyzing the distribution of variables v_1 and v_3 will not let our algorithms distinguish the Markov equivalent model from the true causal model. After a certain number, adding more samples to the data set will not change the result of the conditional independence tests as the distribution of these variables will approach an equilibrium. This is where our causal discovery method has reached its limit. We could improve our model estimate by looking into the process to determine which measured event occurs before the other. With this prior knowledge, we can exclude certain causal relations within the model. The need for prior knowledge of certain problems is a limitation to our causal inference approach.

2. It is more realistic a certain amount of confounding bias is present within a process variable set. Events are likely influenced by unmeasured phenomena. Research could be done on (synthetic) job scheduling cases where unmeasured confounding variables had been present that caused bias in the process variable sets. Causal discovery methods like the FCI, modified PC algorithm, RFCI and more could then be applied. Consequently, the use of the rules of do-calculus could be expanded to predict the effects of interventions when unmeasured confounding variables are present. We haven't covered this part of do-calculus in this thesis.

3. The data sets and causal models used for causal inference are subjected to the Faithfulness assumption. When certain variable relations exist that can't be found in the observational data, the actual outcomes of interventions will likely differ from the predicted outcomes. Research could be done on a framework of how and when this constraint can be relaxed such that consistent predictions can still be made for when the data set does not satisfy this constraint. Having a framework like this and following it would increase the reliability of predictions based on real data sets.

4. We've computed the variances of the post-interventional job duration of job J_1 based on the post-interventional data sample that we generated with Rstudio's *sample* function.

Algorithms generate a *pseudo-random* set, which means the set is not entirely random. This is not an issue since we generated the nominal data in the same manner. However, if a real data set were to be used to generate a post-interventional data set with an algorithm, certain inaccuracies may exist in this post-interventional data. A slight bias will be introduced by the algorithm that does not exist in the real data set. This bias should be addressed as this bias will be incorporated in the optimization. Therefore, research could be done on how post-interventional variance values for discrete distributions can be accurately predicted, or generated with negligible bias compared to the real observational data.

Appendix A

Appendix

A-1 Probability calculus and graph theory fundamentals

Probabilistic inference is the process where probability distributions of variables given a model are determined.

Basic Probability theory

Probability theory is concerned with the probabilities of events occurring, where the occurrence of an event A is expressed as variables taking on a specific value A_i $(i = 1, ..., n)$ within set of all possible events Ω. An event can be defined as a light turning on or off, a system breakdown occurring, a specific input control input value or the outcome of an experiment, which could be the measured duration of a process. In Bayesian formulation the probability of an event occurring is said to be a degree of belief.

For a finite set of events Ω which is the sample space, the probability of an event A taking place is given as $P(A)$. For each event A $\in \Omega$, $P(A)$ should satisfy the following properties

$$P(A) \geq 0 \qquad \forall A \in \Omega \tag{A-1}$$

$$\sum_{A \in \Omega} P(A) = P(\Omega) = 1 \tag{A-2}$$

$$P(A_1, A_2, ..., A_n) = \sum_{n} P(A_i) \qquad i = 1, 2, .., n \tag{A-3}$$

Fundamental rule of probability calculus

Given two events A and B, the probability of both events occurring is given as

$$P(A, B) = P(B|A)P(A) = P(A|B)P(B). \tag{A-4}$$

This leads to the Bayes' rule which is fundamental to probabilistic inference. This rules holds as

Bayes' rule

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} = \frac{P(A,B)}{P(A)}. \tag{A-5}$$

The concept of Bayes' rule is that with having prior knowledge of event B happening, and event A occurring, both pieces of information can be used to update our expectation of event B happening. It can be seen that after the observation of A occurring and having prior knowledge of event B happening, the posterior probability distribution of event B happening under the condition of event A happening can be obtained. To obtain this posterior distribution, prior knowledge $P(B)$ is multiplied by the normalized likelihood of B given a, which is given as $\frac{P(A|B)}{P(A)}$.

Probabilistic networks

Probabilistic networks show conditional dependence and independence relationships within a set of variables and use probability as a unit of measurement for the strength of these (in)dependence relationships. Graphical probabilistic network models have been widely used to show relations between system variables. These models are a combination of graph theory and statistics, with graph theory being the mathematical language used to explain the structure of a graph. A graph consists of a set of nodes and edges connecting internally dependent node pairs subjected to an underlying joint probability distribution. These edges may be undirected or directed. For the undirected case, connected node pairs have a dependence relation and the model would be known as a Markov random field.

The graph in Figure A-1 represents an undirected model of variable set $V_p = \{v_1, v_2, v_3\}$. Node pairs $\{v_1, v_2\}$ and $\{v_2, v_3\}$ are connected by an edge internally, showing both node pairs dependencies. Since both nodes $\{v_1, v_3\}$ are given to be dependent on v_2, they indirectly influence each other. Note that in this literature survey node pairs $\{x, y\}$ may also be formulated as $(x, y)\ \forall\ x, y$.

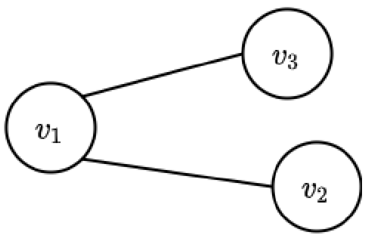

Figure A-1: Undirected graph of variable set V_p.

Bayesian networks

A Bayesian network is a probabilistic graphical model that represents the causal relations between the variables of interest. The directed edges represent causal variable relations in terms of conditional probabilities, where the node that has an edge directed towards its neighbour node is the direct cause of this neighbour node. In Figure A-2 a graph of variable set V_p is given, which represents a Bayesian network.

A graph consisting of directed edges where no path starting and ending at the same node is possible is known as a directed acyclic graph (DAG). In causal inference this type of graph is used. An example is given in Figure A-2

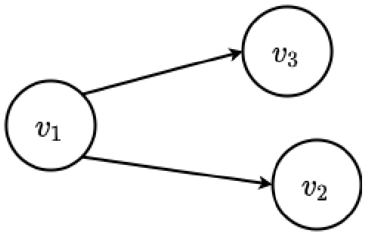

Figure A-2: Directed graph of variable set V_p.

Each subset in a directed acyclic graph is subjected to a probability distribution function. The product of each of these probability distribution functions in the set equals the joint probability distribution function of the graph. This probability distribution function is the product of the conditional probabilities of each node given its parent node(s). The joint probability distribution function of variable set V therefore satisfies

$$P(v_1, ..., v_n) = \prod_i P(v_i|pa(v_i)). \tag{A-6}$$

v	$Pa(v)$
v_1	$\varnothing$
v_2	v_1
v_3	v_1

Table A-1: Nodes of V_p and their parent nodes.

From Table A-1 or from Figure A-2 and Equation (A-10) it follows that

$$P(v_1, v_2, v_3) = P(v_3|v_1)P(v_2|v_1)P(v_1)$$

Independence relations

Two subsets X and Y are independent ($X \perp\!\!\!\perp Y$) if the probability of X and Y occurring together is equal to the probability of both X and Y occurring separately $\forall x, y \in \mathbb{R}$. This is given as

$$P(x, y) = P(x)P(y) \quad \forall x, y \in \mathbb{R} \tag{A-7}$$

Two subsets X and Y are conditionally independent if the probability of X occurring, knowing Z, is not affected by the occurrence of Y $\forall x, y, z \in \mathbb{R}$. Following Definition 1.1.2 [26], consider the subsets of variables X, Y and Z of finite set V with joint probability function P. The sets X and Y are conditionally independent given Z if

$$P(x|y, z) = P(x|z) \quad \text{whenever} \quad P(y, z) > 0, \tag{A-8}$$

which leads us to the definition

$$(X \perp\!\!\!\perp Y|Z) \quad \text{if and only if} \quad P(x|y,z) = P(x|z) \tag{A-9}$$

$\forall x, y, z \in \mathbb{R}$ such that $P(y, z) > 0$.

The strength of the causal dependence between a node and its parent node(s) is given as the conditional probability for this node occurring given its parent(s). The conditional probability of each node in the set given its parent(s) together forms a probabilistic model of a domain that must represent the joint distribution. Independence relations not only change the joint distribution but also lower the complexity of the model. This makes the identification of these relations crucial.

The convenient part of Bayesian network models is that their structure can be exploited to identify these independence relationships. In Figure A-2 it can be seen that nodes v_2 and v_3 are conditionally dependent on node v_1. If the value of node v_2 changes, then the value of node v_3 has also been influenced change as both nodes share the same single parent node which causes the change in variable value. The corresponding joint probability distribution of the node set is therefore given as

$$P(v_2, v_3|v_1) = P(v_2|v_1)P(v_3|v_1). \tag{A-10}$$

For the directed graph of V_p' with new edge directions shown in Figure A-3, it can be seen that node v_2 and node v_3 are both parent nodes of node v_1.

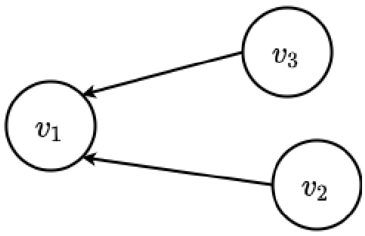

Figure A-3: Directed graph of variable set V_p'.

As these parent nodes have no incoming edges they are independent. This means that the probability of node v_2 and node v_3 happening is given as

$$P(v_2, v_3) = P(v_2)P(v_3). \tag{A-11}$$

Nevertheless, a change in the value of variable v_1 means that either a single parent node or both parent nodes caused this change in variable value. This makes both parent nodes dependent when conditioning on node v_1. The corresponding joint probability distribution function is then given as

$$P(v_1, v_2, v_3) = P(v_2)P(v_3)P(v_1|v_2, v_3). \tag{A-12}$$

A-2 PC algorithm

Algorithm 1 PC algorithm edge removal part

Data: Variable set V
Result: Skeleton of graph G
Initialization of edge removal part
start with connected undirected graph G' and set n=0
repeat
    for *each adjacent node pair* $(v_i, v_j) \in V$ **do**
        for *each subset* $\subseteq$ *set* $Y_i \backslash \{v_j\}$ *with cardinality* n **do**
            Perform a conditional independence test **if** $(v_i \perp\!\!\!\perp v_k | v_j)$ **then**
                Remove edge between v_i and v_k Add v_k to $S_{i,k}$ and $S_{k,i}$
            end
        end
        Update sets $Y_i \subseteq V$
    end
    n = n+1
until *each subset* $\subseteq$ *set* $Y_i \backslash \{v_j\}$ *is of cardinality less than or equal to* n ;

Algorithm 2 PC algorithm edge direction assignment part

Data: Variable set V, edge set E, separation set S
Result: Markov equivalence class of DAG G
initialization of edge direction assignment part
repeat
    for *each node triplet* (v_i, v_j, v_k) *where node pairs* (v_i, v_j) *and* (v_j, v_k) *are adjacent but not* (v_i, v_k) **do**
        assign the edge direction of both node pairs (v_i, v_j) and (v_j, v_k) towards node v_j if and only if v_j is not contained in $S_{i,k}$
    end
until *no triplets eligible for this edge direction orientation are left*;
repeat
    for *each node triplet* (v_i, v_j, v_k) *where* $v_i \rightarrow v_j - v_k$ *and node pair* (v_i, v_k) *not adjacent* **do**
        assign a directed edge such that $v_j \rightarrow v_k$ if this does not create a cycle or collider (node with only inwards directed edges)
    end
until *no triplets eligible for this edge direction orientation are left*;

A-3 Synthetic data table

v_1	5	7.5	10	12.5	15	17.5	20	22.5
n	2012	2019	1511	1422	1368	1035	537	96
v_2	1.5	1.75	2	2.25	2.5	2.75	3	3.25
n	958	3017	549	2417	1016	1025	481	483
v_3	12	13	14	15	16	17	18	19
n	2746	1151	1170	1351	1096	1109	883	494
v_4	60	62.5	65	67.5	70	72.5	75	77.5
n	1929	1654	1510	977	1346	1028	873	683
v_5	60	62.5	65	67.5	70	72.5	75	77.5
n	1982	1700	1423	1036	1474	1055	776	554
v_6	14	15	16	17	18	20	23	24
n	2439	1573	1376	808	711	704	1311	1078
v_7	825	850	875	900	925	950	975	1000
n	477	511	994	1474	2060	483	2529	1472

Table A-2: Observed V_1 variable values and their count n.

v_6 is given to be the observed job duration value. The j-th sample ($j = 1, 2, ..., N$) is noted as $v_{6,j}$. Job duration mean μ_1 is given as

$$\mu_1 = \frac{\sum_{j=1}^{N} v_{6,j}}{N}. \tag{A-13}$$

The job duration value of the j-th sample of the post-interventional data set for the k-th intervention ($k = 1, 2, ..., dim(V_{1,I})$) is given as $v_{i,jk}$.

$$\mu_{1k}(V_1) = \frac{\sum_{j=1}^{N} v_{6,jk}}{N}, \qquad k = 1, 2, ..., dim(V_{1,I}). \tag{A-14}$$

A-4 Synthetic data figure

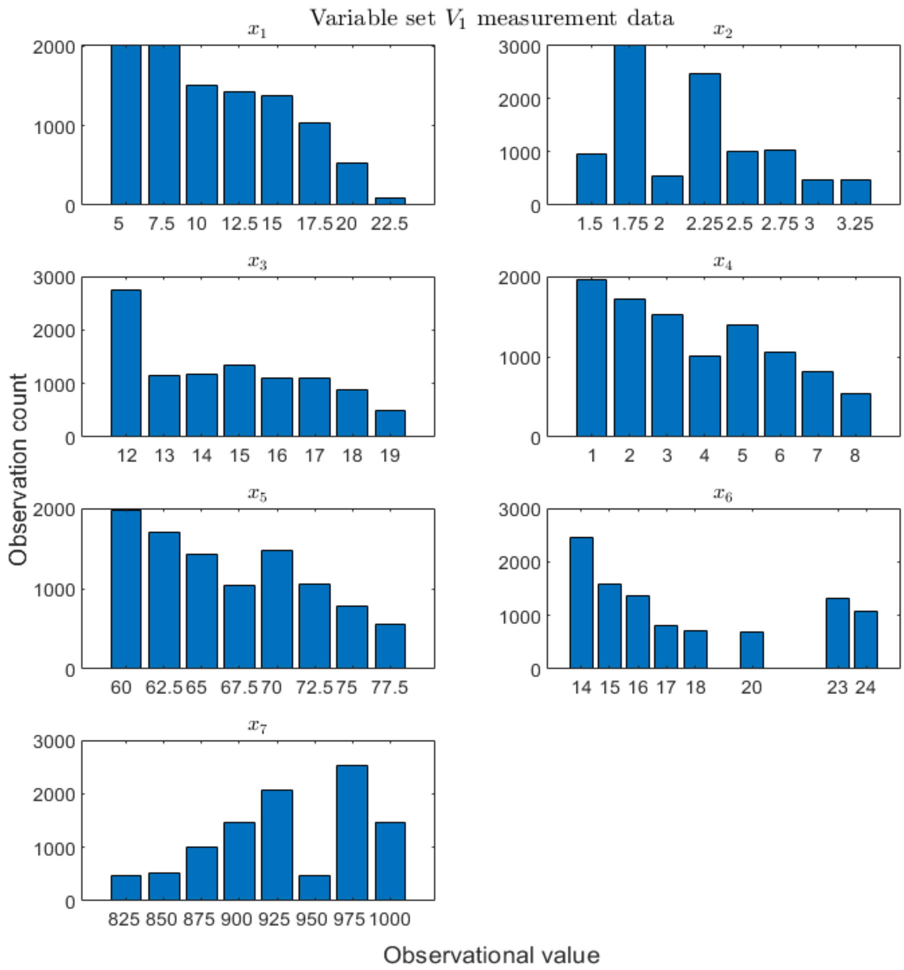

Figure A-4: Distribution of variables in V_1. $x_i{=}v_i$ for $i = 1, 2, ..., 7$.

A-5 Post-interventional probability distributions of v_6

$v_4 \in V_1$	v_4	$do(v_{41})$	$do(v_{42})$	$do(v_{43})$	$do(v_{44})$	$do(v_{45})$	$do(v_{46})$	$do(v_{47})$	$do(v_{48})$
$P(v_{61}\|v_4)$	0.244	0.276	0.253	0.254	0.260	0.250	0.200	0.200	0.194
$P(v_{62}\|v_4)$	0.157	0.143	0.145	0.149	0.173	0.166	0.170	0.182	0.186
$P(v_{63}\|v_4)$	0.138	0.108	0.117	0.150	0.139	0.141	0.175	0.154	0.169
$P(v_{64}\|v_4)$	0.081	0.084	0.093	0.076	0.083	0.075	0.079	0.080	0.068
$P(v_{65}\|v_4)$	0.071	0.105	0.085	0.075	0.068	0.066	0.051	0.038	0.014
$P(v_{66}\|v_4)$	0.070	0.084	0.093	0.061	0.062	0.055	0.073	0.047	0.060
$P(v_{67}\|v_4)$	0.131	0.117	0.123	0.138	0.108	0.134	0.135	0.163	0.141
$P(v_{68}\|v_4)$	0.108	0.083	0.091	0.098	0.107	0.114	0.118	0.137	0.168
$v_5 \in V_1$	v_5	$do(v_{51})$	$do(v_{52})$	$do(v_{53})$	$do(v_{54})$	$do(v_{55})$	$do(v_{56})$	$do(v_{57})$	$do(v_{58})$
$P(v_{61}\|v_5)$	0.244	0.270	0.263	0.244	0.258	0.228	0.206	0.235	0.189
$P(v_{62}\|v_5)$	0.157	0.209	0.202	0.164	0.146	0.138	0.132	0.088	0.070
$P(v_{63}\|v_5)$	0.138	0.138	0.144	0.153	0.146	0.132	0.118	0.130	0.136
$P(v_{64}\|v_5)$	0.081	0.096	0.084	0.074	0.080	0.080	0.077	0.069	0.066
$P(v_{65}\|v_5)$	0.071	0.072	0.064	0.071	0.066	0.072	0.091	0.058	0.069
$P(v_{66}\|v_5)$	0.070	0.066	0.058	0.077	0.075	0.073	0.068	0.081	0.074
$P(v_{67}\|v_5)$	0.131	0.083	0.097	0.117	0.138	0.144	0.169	0.194	0.218
$P(v_{68}\|v_5)$	0.108	0.066	0.089	0.099	0.092	0.132	0.140	0.146	0.1794

Table A-3: The probability distribution of v_6 for the nominal scenario and after interventions on v_4 and v_5

Bibliography

[1] R. L. Daniels and J. E. Carrillo, "β-robust scheduling for single-machine systems with uncertain processing times," *IIE transactions*, vol. 29, no. 11, pp. 977–985, 1997.

[2] C. W. Wu, K. N. Brown, and J. C. Beck, "Scheduling with uncertain durations: Modeling β-robust scheduling with constraints," *Computers & Operations Research*, vol. 36, no. 8, pp. 2348–2356, 2009.

[3] S. Ullah, Q. Liu, C. Zhang, and Y. R. Awan, "Scheduling with uncertain processsing times: Applying β-robust schedule on two-machine flow-shop with constraints," in *2009 IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 1946–1950, IEEE, 2009.

[4] S. Alimoradi, M. Hematian, and G. Moslehi, "Robust scheduling of parallel machines considering total flow time," *Computers & Industrial Engineering*, vol. 93, pp. 152–161, 2016.

[5] S.-M. Khatamia, M. Ranjbara, and M. Davari, "Maximizing service level in a β-robust job shop scheduling model," *Journal of Industrial and Systems Engineering*, vol. 8, no. 4, pp. 59–71, 2015.

[6] B. Wang, X. Xia, H. Meng, and T. Li, "Bad-scenario-set robust optimization framework with two objectives for uncertain scheduling systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 1, pp. 143–153, 2017.

[7] V. Jorge Leon, S. David Wu, and R. H. Storer, "Robustness measures and robust scheduling for job shops," *IIE transactions*, vol. 26, no. 5, pp. 32–43, 1994.

[8] M. Sevaux and S. Dauzère-Pérès, "Genetic algorithms to minimize the weighted number of late jobs on a single machine," *European journal of operational research*, vol. 151, no. 2, pp. 296–306, 2003.

[9] M. T. Jensen, "Generating robust and flexible job shop schedules using genetic algorithms," *IEEE Transactions on evolutionary computation*, vol. 7, no. 3, pp. 275–288, 2003.

[10] M. Sevaux and K. Sörensen, "A genetic algorithm for robust schedules in a one-machine environment with ready times and due dates," *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, vol. 2, no. 2, pp. 129–147, 2004.

[11] P. J. Van Laarhoven, E. H. Aarts, and J. K. Lenstra, "Job shop scheduling by simulated annealing," *Operations research*, vol. 40, no. 1, pp. 113–125, 1992.

[12] M. Dell'Amico and M. Trubian, "Applying tabu search to the job-shop scheduling problem," *Annals of Operations research*, vol. 41, no. 3, pp. 231–252, 1993.

[13] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, vol. 81. 01 1993.

[14] D. Colombo, M. H. Maathuis, *et al.*, "Order-independent constraint-based causal structure learning.," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3741–3782, 2014.

[15] T. Le, T. Hoang, J. Li, L. Liu, H. Liu, and S. Hu, "A fast pc algorithm for high dimensional causal discovery with multi-core pcs," *IEEE/ACM transactions on computational biology and bioinformatics*, 2016.

[16] R. Tu, C. Zhang, P. Ackermann, K. Mohan, H. Kjellström, and K. Zhang, "Causal discovery in the presence of missing data," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1762–1770, PMLR, 2019.

[17] D. Colombo, M. H. Maathuis, M. Kalisch, and T. S. Richardson, "Learning high-dimensional directed acyclic graphs with latent and selection variables," *The Annals of Statistics*, pp. 294–321, 2012.

[18] S. Shimizu, P. O. Hoyer, A. Hyvärinen, A. Kerminen, and M. Jordan, "A linear non-gaussian acyclic model for causal discovery.," *Journal of Machine Learning Research*, vol. 7, no. 10, 2006.

[19] D. M. Chickering, "Optimal structure identification with greedy search," *Journal of machine learning research*, vol. 3, no. Nov, pp. 507–554, 2002.

[20] J. Pearl, "Causal diagrams for empirical research," *Biometrika*, vol. 82, no. 4, pp. 669–688, 1995.

[21] Y. Huang and M. Valtorta, "Pearl's calculus of intervention is complete," *arXiv preprint arXiv:1206.6831*, 2012.

[22] Y. N. Sotskov and N. Y. Sotskova, *Scheduling under uncertainty: theory and algorithms.* Belorusskaya Nauka, 2010.

[23] W. Hines, "Montgomery. dc. 1990, probability and statistics in engineering and management science."

[24] D. E. Holmes, *Innovations in Bayesian networks: theory and applications*, vol. 156. Springer, 2008.

[25] M. Kalisch and P. Bühlmann, "Estimating high-dimensional directed acyclic graphs with the pc-algorithm," *Journal of Machine Learning Research*, vol. 8, no. Mar, pp. 613–636, 2007.

[26] J. Pearl *et al.*, *Models, reasoning and inference.* 2000.

[27] Y. N. Sotskov, V. S. Tanaev, and F. Werner, "Stability radius of an optimal schedule: A survey and recent developments," in *Industrial applications of combinatorial optimization*, pp. 72–108, Springer, 1998.