DATA-DRIVEN CAUSAL DECISION SUPPORT FOR BUSINESS PROCESS MANAGEMENT

ALI JANDAGHI ALAEE

SUPERVISOR:
PROF. ARIK SENDEROVICH

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTERS OF ARTS

GRADUATE PROGRAM IN INFORMATION SYSTEMS AND TECHNOLOGY
YORK UNIVERSITY
TORONTO, ONTARIO

APRIL 2024

© ALI JANDAGHI ALAEE, 2024

Abstract

Control-flow and resource assignment decisions influence business processes. Recorded process data can be used to identify which decisions are informed by data to predict their outcome, and to guide interventions as part of a what-if analysis. The latter requires causal models that explain decisions. Yet, existing methods are limited: they focus on control-flow decisions only, ignore potential confounders, and use ad-hoc methods to resolve causal conflicts. We fill this gap, by introducing a causal decision modeling framework which uncovers confounding effects, and captures resource decisions. Moreover, we provide a process-aware causal discovery algorithm that takes process precedence into account. In addition, we employ domain knowledge to include unobserved factors. We address the problem of identification, conduct interventional outcome prediction and improve decision-making by acquiring unavailable data to maximize the utility of interventions. We demonstrate the feasibility of our approach through a set of experiments on synthetically generated and real-world datasets.

Acknowledgements

First of all, I acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).
This research was also supported by York University's Catalyzing Interdisciplinary Research Clusters (CIRC) program for the Research Cluster on Data Economy.

I sincerely appreciate my wonderful supervisor, Prof. Arik Senderovich, whose positive outlook and unwavering support from the very beginning boosted my motivation and made this journey an amazing experience, full of personal and academic growth.

Additionally, I would like to thank Prof. Matthias Weidlich for his insightful contributions to this project and his warm hospitality during my visit to Humboldt University of Berlin.

Finally, I am deeply grateful to my lovely family, who always encouraged me to pursue my professional endeavors.

Table of Contents

Abstract ii

Acknowledgements iii

Table of Contents iv

List of Tables vi

List of Figures vii

1 Introduction 1

2 Literature Review 5
- 2.1 Causal Inference . . . 5
- 2.2 Decision Support and Causal Analysis in Business Processes . . . 7

3 Preliminaries 10
- 3.1 Causal Modeling and Reasoning . . . 10
- 3.2 Constraint-Based Causal Discovery . . . 12
- 3.3 The Knapsack Problem . . . 13

4 A Framework for Causal Decision Support in Processes 15
- 4.1 Modeling Causal Decision Structures . . . 15
- 4.2 Flow Analysis for What-If Interventions via Causal Decisions . . . 16
- 4.3 Process-Aware Causal Discovery . . . 17
  - 4.3.1 Step 1: Mining Temporal Decision Tiers . . . 18
  - 4.3.2 Step 2: Adding Forbidden Edges . . . 19
  - 4.3.3 Step 3: Incorporating Contextual Variables into Tiers . . . 19
  - 4.3.4 Step 4: Constraint-based Discovery and Post-processing . . . 20
- 4.4 Evaluation . . . 21
  - 4.4.1 Synthetic Data for Running Example . . . 21
  - 4.4.2 Real-world Data . . . 22
- 4.5 Discussion . . . 23

5 Unidentifiable Causal Queries in Business Processes **24**
5.1 Identifiability in a Motivating Example 24
5.2 Latent Variables in Business Process Causal Models 26
5.2.1 Identifiability and Estimation of Causal Decision Queries 26
5.3 DAP: Data Acquisition for Unidentified Queries 29
5.3.1 Problem Definition 29
5.3.2 Applying DAP in Practice 31
5.4 Evaluation 32
5.4.1 Implementation 32
5.4.2 The Case of ED-Anonymous 32
5.5 Discussion 34

6 Discussion and Limitations **35**

Bibliography **38**

Appendix **43**

List of Tables

2.1 Previous work on data-driven decision support (≈ denotes partial support). *Temporal:* Utilizes temporal information. *Vars:* Incorporates contextual variables. *Ident:* Addresses the issue of identification. 7

4.1 Tables showing generative probabilities and the final tiers. 21
4.2 Causal effect table. 22
4.3 Distribution of D_1 after intervening on discrtized_amount_req 23

List of Figures

1.1 Running example: a BPMN model of a car purchase and delivery process. . . 3

3.1 The DAG of the causal model representing the relations between variables of the example given in Fig. 1.1; variable V denotes the price of the ordered car. 11

4.1 Overview of our approach to process-aware causal discovery. 18

5.1 Three causal graphs that could account for the loan request decision D. . . 26
5.2 Identifiability and estimation. 26
5.3 Data acquisition problem overview. 29
5.4 Left: the causal graph of ED-Anonymous; contextual variables: Age (AG), Gender (GE), Arrival Mode (AM), Medical History (MH), Physical Assessment (PA); intervention variable Triage Code (TC); decision variable: Is Consult (IC). Right: The mutilated causal graph after an intervention on TC and acquisition of PA. 34

6.1 BPMN model obtained by running inductive miner on BPIC 2012 dataset. . 43

Chapter 1

Introduction

Decisions play a central role in business processes [1]. They influence the outcome of individual cases and, in the larger context, implement organizational policies. Methods to improve decision-making, therefore, are vital to the success of organizations [2]. With the ample availability of process-related data, the opportunity to develop data-driven decision support has been widely recognized and methods to learn, predict, and prescribe decisions have been developed [3, 4, 5, 6, 7, 8, 9].

Traditional approaches to data-driven decision support, however, are not grounded in causal models. As a consequence, they cannot support accurate 'what-if?' analysis and also lack means to quantify the uncertainty associated with decisions. Rooted in the deciphering of cause-and-effect dynamics from observational or experimental data, causal inference equips researchers with the tools necessary for informed decision-making. To measure the true effect of variables on an outcome (that could be a decision), the first solution is to run Randomized Controlled Trials (RCTs). These experiments involve randomly assigning participants to different groups, with one group receiving the treatment being studied and another group serving as a control. By controlling for extraneous variables and using randomization, researchers can confidently attribute any observed differences between the groups to the effect of the treatment [10].

However, conducting Randomized Controlled Trials (RCTs) can be costly, time-consuming, and sometimes impractical. Therefore, there is a need to leverage causal inference techniques that can utilize observational data to infer intervention outcomes without the need for RCTs. This underscores the importance of having a true causal graph that accurately models the environment. Therefore, discovering such a comprehensive graph for business processes from an event log, and utilizing it for analysis purposes, especially for decision support is crucial.

Only recently, first approaches to overcome these limitations have been proposed, such as [11]. However, these existing techniques exhibit substantial drawbacks due to not taking advantage of event logs, as the main input, to its full capacity; To begin with, the impact of possible confounders is disregarded and the models are mostly limited to the control-flow, which neglects decisions related to other process perspectives, such as resource assignments. Additionally, most of the previous works neglect the temporal instinct of an event log as an

auxiliary tool in the causal discovery phase.

To illustrate the importance of the above aspects, Fig. 1.1 gives a BPMN model of a process at a car dealership. Business Process Model and Notation (BPMN) is a graphical representation standard for depicting business processes in a clear and consistent manner. It provides a visual language composed of symbols and shapes that enables organizations to map out their workflows, identify inefficiencies, and optimize operations. BPMN diagrams offer a common understanding among stakeholders by illustrating the sequence of activities, decision points, and interactions within a process. In a BPMN model, rectangle shapes represent an activity in a process which usually leaves a trace in the event log after execution, and arrows demonstrate the workflow of the process and structurize the possible paths for cases. To demonstrate a workflow choice, an "$\times$" symbol, divides an incoming arc to two different paths. Additionally, a "+" symbol is used when two branches of activities can be executed concurrently.

The process in Fig. 1.1 starts with an order (containing a price V) and ends with either the rejection of the order or the delivery of the car. The process features several control-flow decisions: D_1 on the type of the customer document assessment; D_2 on the rejection of the order; D_3 on the type of pre-delivery inspection, and D_4 on the need for a technical fix. Also, the process involves decisions on the assignment of resources, exemplified as D_5 and D_6 for the feature inspection and technical inspection, respectively.

For this process, decisions D_3 and D_4 have a large impact on the cycle time, due to the duration of a complete inspection, in particular when involving a technical fix. Support for decision-making shall therefore explore the factors that influence these decisions. Yet, existing techniques to construct a respective causal model [11] neglect confounders such as contextual variables and those that do, neglect the temporal order of events in a process, and therefore, may not identify the true causal graph among multiple candidates.

In the given example, decision D_1 on the document assessment influences decision D_2 on the rejection, but not the decision D_3 on the inspection. However, the order price V is a confounder, influencing all three decisions. Existing work would ignore it and, hence, identify a spurious causal effect of D_1 on D_3. Similarly, the resource assignment decisions D_5 on D_6 influence decision D_4. Existing work cannot capture their effect, since the decisions do not materialize as control-flow choices in the data. Both limitations severely bias the estimation of causal effects, which leads to misleading analyses, as we later illustrate empirically.

Even if the most accurate causal model is obtained from data, still its validity is not guaranteed. Certain hidden or unmeasured variables may still be overlooked in the resulting causal graph, as we thoroughly elaborate in Section 5.1. The absence of these variables in the event log may stem from their impracticality to measure or the high costs associated with gathering them within the business process context. Nevertheless, these unobserved factors could potentially be discerned through domain knowledge and subsequently integrated into the established causal graph. The presence of latent variables in a causal graph presents the challenge of *identification* of causal queries, which involves determining the computability of a causal query given the current availability of data. None of the previous works have addressed this potential scenario, and as a result, they do not propose a consistent procedure

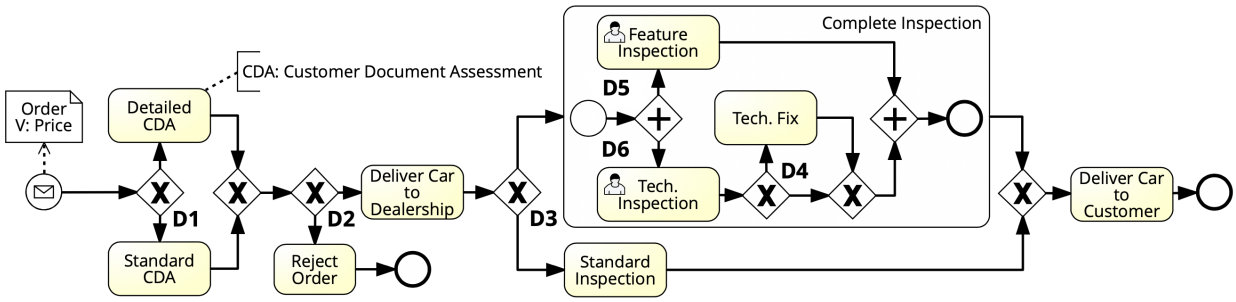


Figure 1.1: Running example: a BPMN model of a car purchase and delivery process.

to utilize domain knowledge in their causal models, or address the subsequent issues that arise alongside it.

Based on the aforementioned gaps, in this thesis, we propose a unified framework for probabilistic data-driven decision support that enables modeling of causal structures to analyse *what-if?* questions based there on. This framework is equipped with a novel method for causal discovery tailored for business processes that leverages the temporal nature of recorded events to discover a faithful causal structure containing all types of variables and decisions. We also allow for manually incorporating latent variables, and *identifying* whether, due to the inability to measure some of the features, data cannot fully support decision-making. If this is not the case, our framework supports *predicting* the outcome under the current running policy, or gaining insights about scenarios that are not observed in data, *estimating* the effect of potential interventions by manipulating variables, that might lead to improved decisions. Furthermore, we show how to analyse queries that turn out to be unidentifiable, due to unavailable data for latent features. For such cases, we formulate a *data acquisition problem* (DAP) and optimize it mathematically such that balances the cost of acquiring data and the utility gained by it.

We summarize our contributions as follows:

(1) We introduce a model to facilitate probabilistic and causal reasoning for process decision support. It goes beyond the state of the art by integrating potential confounders, namely contextual variables, and enabling the inclusion of various types of decisions.

(2) We show how to discover instances of our model from event data on process executions. Unlike existing approaches that separate the identification of process structures and causal structures, we adapt a tier-based causal discovery algorithm for event data, thereby seamlessly combining process mining and causal reasoning.

(3) We enable inclusion of hidden factors by domain knowledge in our framework, and investigate cases where unavailable data for variables in a causal structure results in unidentifiable queries. We propose a new technique for data acquisition to reach the most utility gained from transforming unidentifiable queries into identifiable while considering the costs and utilities associated to variables and queries.

We demonstrate the feasibility of our proposed main framework through a proof-of-concept implementation. We then, establish the improvement achieved by our modeling over the state

of the art with a controlled experimental setup and illustrate scalability using a real-world dataset. Furthermore, we implemented an extension to handle cases where hidden variables exist, and then experimented it alongside solving the Data Acquisition Problem (DAP) on a real-world dataset.

In the remainder, we first review the literature about causality and its applications in business processes (Chapter 2). Based thereon, we carve out the research gaps and characterize the problem of causal decision support for business processes. Following this, we formally define the fundamentals in Chapter 3. Subsequently, we aim to define our causal decision support framework including a novel discovery method from event log in Chapter 4. In Chapter 5, we delve into the discourse surrounding unmeasured variables and propose an optimization problem aimed at addressing the identification challenge. We then consolidate all our contributions, evaluate their limitations, and conclude our study by offering insights and suggestions for future research directions in Chapter 6.

Chapter 2

Literature Review

2.1 Causal Inference

In the realm of scientific inquiry, establishing causation represents a pivotal challenge, and scholars have long grappled with the complexities inherent in drawing meaningful causal inferences from observational data. Here we provide a brief review of works accomplished in this domain which has guided this thesis.

Directed acyclic graphs (DAGs) have become widely studied and applied in causal modeling and discovery [12, 13, 14]; their simple directed structure provides an interpretable representation for causality and facilitates systematic learning procedures. Indeed, given an infinite amount of observational data from a system of variables, there exist algorithms capable of learning the set of DAGs equivalent to the true causal DAG [13, 15]. Accordingly, any edge in common among the learned DAGs may be interpreted causally in the large sample limit. This guarantee comes at the cost of several assumptions, including the standard causal Markov, faithfulness, and the no unmeasured common causes—namely no latent confounding assumption. The latter is generally referred to as causal sufficiency.

Causal structure discovery. The causal discovery problem consists in selecting a causal graph as a possible explanation for a given data set [16]. More specifically, these algorithms map the implications of a graphical separation criterion (d-separation) on DAGs to conditional independence in the data [17]. Unfortunately, this process is underdetermined in the sense that multiple DAGs may be consistent with the patterns of conditional independence in the data. Therefore, these algorithms learn a set of equivalent DAGs, more precisely a *Markov equivalence class (MEC)* and label any edge that varies between two or more of the learned DAGs as causally ambiguous.

Discovery approaches are classified into two major groups. The first one, named as constraint-based algorithms try to recover the causal graph by exploiting a set of conditional independence statements (CISs) obtained from a sequence of statistical tests. This class of methods translates conditional probability independence into graphical separation by assuming faithfulness of the underlying distribution. Faithfulness suggests that the statistical dependencies observed in the data are consistent with the causal relationships present in

the true causal graph. One of the most studied algorithm that leverages the CISs is the Peter-Clark (PC) algorithm [13] with its variants [18]. A first extension of the PC algorithm to the causal insufficient setting is represented by the Fast Causal Inference (FCI) algorithm [19]. Specifically, the FCI algorithm relaxes both the assumption of no latent confounding and no selection bias in the observational setting, pushing the causal discovery problem a step closer to real-world scenarios.

Score-based algorithms, the second class of discovery methods, are usually structured around the maximization of a measure of fitness of a graph through a space of possible graphs for the observed samples. The main difference between these two classes of learning is that constraint-based algorithms tend to start with a fully connected graph and use conditional independence tests to eliminate and orientate some of those edges. On the other hand, score-based algorithms which include the well-established Greedy Equivalence Search (GES) [20], Hill Climbing (HC) and TABU search, typically start from an empty graph and iteratively modify the graph to increase its objective score. Hybrid algorithms, which combine constraint-based and score-based methodologies, are also prevalent for mitigating their individual drawbacks. Adaptively Restricted GES (ARGES) and Greedy FCI (GFCI) are examples of hybrid methods [21].

Several recently published surveys have thoroughly summarized and classified causal discovery methods and their properties, such as [21, 22].

Prior knowledge in causal discovery. In the field of causal machine learning, many structure learning algorithms can be modified to account for causal knowledge that guides search towards a preferred causal structure, or restricts the search space of graphs that can be explored to those in which graphs contain or do not contain certain relationships. A knowledge approach that guides, rather than restricts, structure learning is often referred to as an approach that introduces soft constraints, whereas an approach that restricts learning towards specific graphs tends to be referred to as an approach that introduces hard constraints. Prior knowledge in any form could help resolving uncertainties about causal relationships, thereby increasing the number of identifiable causal relationships. For instance, in [23], variable types are used to determine the validity of causal relationships. Furthermore, [24] shows that FCI, a causal discovery algorithm that can model latent confounding, is sound and complete when given tiered background knowledge. This survey [25] introduces some novel and also describes some existing, knowledge-based approaches that enables combining structure learning with knowledge obtained from heterogeneous sources.

Table 2.1: Previous work on data-driven decision support (≈ denotes partial support). *Temporal:* Utilizes temporal information. *Vars:* Incorporates contextual variables. *Ident:* Addresses the issue of identification.

Methods	Temporal	Causal	Vars.	Ident.
Decision mining [3, 5]	-	-	+	-
Resource Assignment Mining [28]	-	-	≈	-
Decision Modeling & Notation (DMN) [6, 7]	-	≈	-	-
Fuzzy Decision Req. Diagrams (DRD) [8]	-	+	-	-
Decision Requirements Graph (DRG) [9]	-	+	-	-
Influence Diagrams (IDs) [29]	-	≈	≈	-
Causal control-flow mining [11]	+	+	-	-
Causal Rules from Event Logs [30]	-	+	+	-
Root cause using structural eq. models [31]	-	+	+	-
Framework proposed in this thesis	+	+	+	+

Another line of research has delved into examining the consequences of actions based on causal assumptions represented in a Directed Acyclic Graph (DAG) and statistical knowledge articulated through a probability distribution. Specifically, the focus lies on predicting distributions for outcomes following actions, taking into account a set of measurements. Pearl introduced the do-calculus as a comprehensive set of rules aimed at determining the answerability or *identifiability* of post-action outcomes given current settings [17]. More specifically, he argued that hidden, unobserved or unmeasured variables in a DAG could lead to unidentifiable causal queries, where the true effect of an intervention may not be correctly estimated with current probability distributions. Moreover, in the same vein, [26, 27] put forth robust algorithms, namely *ID* and *IDC*, designed to estimate the effects of interventions using a DAG as a causal graph and data in both unconditioned and conditioned modes.

2.2 Decision Support and Causal Analysis in Business Processes

Our work relates to data-driven decision support (e.g., decision modeling and mining) and prescriptive process monitoring. Below, we provide an overview of the respective literature, and then in the next section we will outline the research gaps that our approach aims to address (summarized in Table 2.1).

Decision support for business processes received much attention recently, fueled by novel formalisms for decision modeling, such as Decision Modeling and Notation (DMN) [1], and advancements in process mining [32].

Early work in the area discovers decision trees from event data to predict the next activity and explain historical control-flow choices [3, 5]. Yet, the model does not include causal

structures, and focuses on the prediction task, using only features available in the data. Other work considers decisions related to resource assignments in processes [28]. Albeit probabilistic, the approach disregards hidden variables, and does not consider interventions.

Models of business process may be refined with models of their decision logic (via DMN) that are derived from historical data [7, 6]. These ideas have later been refined to incorporate uncertainty related to decision outcomes through the discovery of fuzzy decision models [8, 9]. Yet, latent variables and, hence, identifiability of interventions are ignored.

To improve decision making, one may also adopt specialized Bayesian networks, known as influence diagrams, and treat them as causal models [29]. Such an approach highly depend on domain knowledge, assumes deterministic decisions, and does not take advantage of the temporal order of event in the discovery task.

Recently, first approaches that aim at learning causal dependencies between variables that may influence decisions have been proposed [11]. They align an Upper Bound Causal Graph (UBCG) which is obtained from the process map with the outcome of a simple causal discovery (PC) algorithm to reach a DAG. Although their causal structure is confined with the temporal constraints extracted from the process, the framework ignores contextual variables, which we demonstrate to be crucial in analyzing decisions. Moreover, their solution only focuses on control-flow decisions, and overlook resource assignments, which are naturally embedded into our approach. Additionally, the existence of possible latent variables is ignored, and therefore, conclusions on identifiability cannot be reached with this framework.

Considering the broader context, beyond decision support, causal reasoning was employed to facilitate root-cause analysis for business processes [30, 31]. The study by Bozorgi et al. [30] demonstrated how a causal uplift tree can be mined to suggest recommendations on potential controllable treatments to achieve a desirable outcome within sub populations. However, their work primarily falls under rule-mining efforts rather than learning a causal graph that comprehensively accounts for the relationships among variables. Qafari and van der Aalst [31] presented a technique to discover eqational causal models. Although they included contextual variables, they did not utilize the aid of temporal precedence and therefore, require domain knowledge to select the true graph among the class of Markov equivalent graphs. Furthermore, since they accept continuous variables, they rely on additional assumptions of the additive noise. Moreover, prescriptive analytics that strives for interventions to improve process outcomes adopts causal models [33, 34, 35, 36]. Here, causal dependencies are leveraged to answer prescriptive questions, e.g., on the next action to minimize cycle times [33, 34]. However, in addition to their focus on different use cases, we note that the models adopted for root-cause analysis and prescriptive analytics do not support conclusions on the identifiability of the respective queries.

Research Gaps. Table 2.1 summarizes the methods close to our work, both from decision support and causal inference. Based thereon, we identify the following research gaps:

- Many approaches in decision mining do not adopt *causal modeling*, thereby rendering any estimation related to interventions potentially inaccurate.
- Only [11] has aimed to incorporate temporal information to discover causal graphs.

- Certain studies neglect to incorporate contextual variables into decision analysis, while some others that do include them lack means for probabilistic modeling. Instead, they assume that one configuration of contextual variables leads to a single decision outcome.
- All existing models, fail to consider latent variables and the problem of *identifiability*.

To address these gaps, in Chapter 4, we present a comprehensive framework for modeling decisions in business process management, drawing upon causal theory [14]. Furthermore, we expand upon this framework in Chapter 5, incorporating the analysis pertaining to unobserved attributes.

Chapter 3

Preliminaries

In this chapter, we formally define the fundamentals required throughout this thesis. More specifically, We begin by presenting crucial formalizations concerning causal modeling, intervention, average treatment effect, and identification (Section 3.1), before delving into the discovery of causal models (Section 3.2). Throughout this discussion, we underscore the significance of employing a causal model to analyze decisions within business processes utilizing the example depicted in Fig. 1.1. We also dedicate Section 3.3 to discussing the knapsack problem for our data acquisition problem.

3.1 Causal Modeling and Reasoning

Causal Models. A causal model captures relations between variables and a joint probability distribution over their values. We recall basic definitions using [14].

Definition 1 (Causal Model)**.** *A causal model is a pair* $(\mathcal{G}, P)$ *with:*

- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *being a directed acyclic graph (DAG) over a set of variables* $\mathcal{V}$*. Every vertex is a variable in* $\mathcal{V}$*, and there exists an edge* $(V_i, V_j) \in \mathcal{E}$*, iff* $V_i \in pa(V_j)$*, with pa being a function that returns the parents of a node* $V \in \mathcal{V}$*; the parents are interpreted as the* direct causes *of* V*.*
- P *being a joint probability distribution, such that*

$$P(v_1, \ldots, v_n) = \prod_{i=1}^{n} P(v_i \mid pa(v_i)), \tag{3.1}$$

which is a standard Bayesian network decomposition of the joint distribution.

The structure of the causal model of our running example from Fig. 1.1 is illustrated in Fig. 3.1. The price of the order, variable V influences the choices on the customer document assessment (D_1), on the rejection of the order (D_2), and on the inspection (D_3). The document assessment (D_1) also directly influences the order acceptance (D_2). Moreover, the resource assignment for the technical inspection (D_6) influences the decision on the need for a technical fix (D_4).

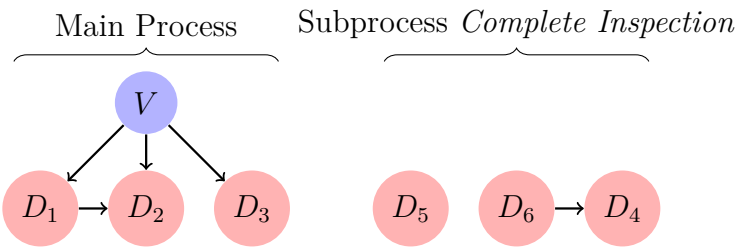


Figure 3.1: The DAG of the causal model representing the relations between variables of the example given in Fig. 1.1; variable V denotes the price of the ordered car.

We write $P(v)$ to denote the probability that a random variable V takes a value v, i.e., $P(v)$ is defined as $P(V = v)$. For a set of random variables $\mathcal{V}$, aka a configuration, we write $P(\mathcal{V} = v)$ to refer to $P(V_1 = v_1, \ldots, V_n = v_n)$, for all $V_i \in \mathcal{V}$. Considering the example in Fig. 5.1 and focusing solely on the variables of the main process, we can write the joint probability as:

$$P\left(d_1, d_2, d_3, v\right) = P(d_3 \mid v)P(d_2 \mid v, d_1)P(d_1 \mid v)P(v). \tag{3.2}$$

Using historical data, one can estimate each of the quantities and answer *statistical* (i.e., predictive) queries about the decisions in Fig. 3.1. However, interventional queries ('what would have been the outcome of decision D_2, if the price changes?') cannot be answered only using historical data and the decomposition in Eq. 3.2. Instead, one would require the following notion of interventions.

Interventions and Causal Queries. An intervention is a deliberate change of the values of some variables, usually to establish the causal effect of one variable on other variables. It is represented by a *do* operator, with $do(V = v)$ (or $do(\mathcal{V} = v)$ for a set of variables) denoting that the value of variable (or set thereof) has been set to v. We consider interventions as a modification of a causal graph $\mathcal{G}$ to a mutilated graph, $\mathcal{G}'$, such that all incoming edges to the intervened variables are eliminated, noting that the pre- and post-intervention distributions are different. For an intervention $do(V_i = v_i)$, the transformation corresponds to a truncated factorization formula:

$$P\left(v_1, \ldots, v_n \mid do(V_i = v_i')\right) = \begin{cases} \prod_{j \neq i} P\left(v_j \mid pa(v_j)\right) & \text{if } V_i = v_i' \\ 0 & \text{if } V_i \neq v_i' \end{cases}. \tag{3.3}$$

Eq. 3.3 reflects the removal of the term $P(v_i \mid pa(v_i))$ from the product of Eq. 3.1, since $pa(v_i)$ no longer influences V_i. We think of such an intervention as a causal query, formulated as follows, for which we receive probabilistic responses on the changes in the non-intervened variables:

$$P(T \mid do(I = i)), \tag{3.4}$$

with target $T \in \mathcal{V}$ and intervention $I \in \mathcal{V}$. As shorthand notation, we refer to a query as a doublet $q = (T, I)$, using the symbol $\perp$ to replace i in the absence of a context condition or an intervention, respectively. We write $\mathcal{Q}(\mathcal{V})$ for the set of all possible queries over $\mathcal{V}$.

Average Treatment Effect. An important quantity that measures the impact of an intervention is the average treatment effect (ATE), which is defined as:

$$\mathbb{E}[Y \mid do(T = 1)] - \mathbb{E}[Y \mid do(T = 0)] \tag{3.5}$$

where Y is an outcome, and T is a binary treatment (our intervention; it can be easily extended to non-binary treatments). For the example in Fig. 1.1, we may wish to assess the impact of setting $D_1 = 1$ (choosing the upper branch in the process model) on the expected outcome of decision D_2. Since D_2 is also binary (reject or not), we can write the ATE as:

$$\begin{aligned} \text{ATE} = &\sum_v P(D_2 = 1 | D_1 = 1, V = v) P(V = v) - \\ &- \sum_v P(D_2 = 1 | D_1 = 0, V = v) P(V = v). \end{aligned} \tag{3.6}$$

The two quantities (and difference) can then be estimated using historical data.

Identifiability. One may incorporate latent variables manually into the $\mathcal{G}$ in a causal model due to recommendation of a domain expert. Consequently, denoting the new variables as $\mathcal{V}_{um}$ and their corresponding new edges as $\mathcal{E}_{um}$, the new DAG would be defined as $\mathcal{G}_{dk} = (\mathcal{V} \cup \mathcal{V}_{um}, \mathcal{E} \cup \mathcal{E}_{um})$. Naturally, the probability distribution P associated to the causal model pertains to $\mathcal{V}$ and the probability distribution of $\mathcal{V}_{um}$ is unavailable. In such situations, the answerability of a causal query based on P is to be determined. We define the notion of identifiability of an intervention, as follows:

Definition 2 (Identifiable Causal Queries)**.** *A causal query q is identifiable based on a manipulated causal model $(\mathcal{G}_{dk} = (\mathcal{V} \cup \mathcal{V}_{um}, \mathcal{E} \cup \mathcal{E}_{um}), P(\mathcal{V}))$, if*

$$P(T \mid do(I = i)) = f(P(\mathcal{V})), \tag{3.7}$$

i.e., the probability function can be computed from the observed variables with f being some function over realized variables, $\mathcal{V}$.

To ensure identifiability, the observed variables, summarized by the probability function P over $\mathcal{V}$, need to be assessed in the context of the augmented causal graph $\mathcal{G}_{dk}$, which specifies (qualitatively) which variables participate in the determination of each variable in the domain [14]. That is, both the data (summarized by a probability distribution) and the causal graph are needed to decide whether a query can be answered.

3.2 Constraint-Based Causal Discovery

Given some observational data, causal discovery aims at uncovering the 'true' causal model that generated the data [14]. Approaches for causal discovery can be divided into three categories [37]: (1) constraint-based discovery, (2) scored-based discovery, and (3) learning functional causal models (FCMs).

FCM-based methods are mostly suitable for data with continuous variables, and thus are less suitable for discovery of decisions. Constraint-based methods conduct a series of conditional independence tests between variables in a fully connected causal model to eliminate (or allow) possible causal arcs. Score-based methods are a relaxation of constraint-based discovery, where independence tests are replaced with goodness-of-fit tests: Instead of searching for a graph that contains true independence relations, 'good enough' graphs are discovered.

As highlighted in Chapter 2, a few works have studied the discovery of a business process causal model [11, 31]. These endeavours predominantly leverage established discovery algorithms from the causality domain and apply them to datasets sourced from event logs. Despite the inclusion of recently developed advanced algorithms in these studies, the utilization of supplementary prior knowledge derived directly from the business process itself has been infrequent in supporting this task.

Without introducing any additional background knowledge, causal discovery can lead to ambiguity in the resulting structure since different causal models may manifest the same dataset. Therefore, we employ [25], an approach that allows to choose between constraint-based and score-based methods and combines historical data with background knowledge in multiple forms, e.g., temporal constraints over variables, and forbidden edges.

Tier-based causal discovery. Within the realm of knowledge-based causal discovery techniques, tier-based discovery involves utilizing a table that indicates the time order of variables, which can be partially specified, alongside the input dataset. The rationale behind this concept involves introducing constraints to discovery algorithms based on the time precedence of variables to streamline the search space. More precisely, it operates on the premise that variables set after others cannot serve as parents to those variables in a causal graph. It not only leads to a substantial reduction in computational costs but also results in the exclusion of numerous potential outputs that do not align with pre-established conditions.

3.3 The Knapsack Problem

As we shall see in Section 5.3, we will propose a unique variation of knapsack problem to optimize the Data Acquisition Problem (DAP) stemming from the identification problem in causal models. Thus, in this section we summarize works related to this problem. The Knapsack Problem (KP) has a rich historical lineage, with its earliest documented mention dating back to 1896 [38]. In the 1950s, Bellman introduced the first dynamic programming (DP) approach to tackle KP [39], marking a significant milestone in its algorithmic treatment. Since then, KP has been the subject of rigorous academic inquiry, with seminal contributions emerging, notably documented in works such as [40, 41]. The complexity and breadth of knapsack variants, along with the wealth of associated research contributions, have led to the publication of specialized literature dedicated to knapsack problems, exemplified by texts like [42, 43, 44]. The Set-Union knapsack Problem (SUKP), introduced in [45], is the closest variation of this problem to what we aim to propose later in Chapter 5.

The knapsack problem asks for the selection of items from a set N with n elements, every item $j \in N$ with a non-negative integer weight w_j associated with it [44]. A subset of items $S \subseteq N$ is called feasible when $\sum_{j \in S} w_j \leq c$ applies, where c is the capacity of a knapsack for carrying the items. Additionally, a profit is incurred by choosing a feasible subset of items and is measured through a properly defined function of N. The aim is to select a feasible $S \subseteq N$ which maximizes profit. The problem appears in distinct versions, each of them characterized by a different type of objective function.

Chapter 4

A Framework for Causal Decision Support in Processes

In this section, we introduce a data-driven framework for causal decision modeling in business processes. First, we introduce the formal model (Section 4.1), followed by an application of the model to flow analysis (Section 4.2). Subsequently, the novel process-aware causal discovery is proposed in Section 4.3 before the its evaluation on synthetic and real-world data in Section 4.4.

4.1 Modeling Causal Decision Structures

We adopt a causal model to capture the decision structures of a business process as follows. Following Def. 1, a causal model is a pair $(\mathcal{G}, P)$ of a causal graph and a probability distribution. Specifically, the causal graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined over the set of all possible business process variables $\mathcal{V}$ that influence the execution of the process, e.g., control-flow decision points, resource assignment decisions, and contextual variables (such as price in our running example). The set of edges, $\mathcal{E}$, represents direct causal effects between those variables.

While we address the task of discovering the graph, $\mathcal{G}$, later (Section 4.3), we note that the estimation of P without interventions is a purely statistical problem that does not require $\mathcal{G}$. Yet, to learn the behavior of P under a *do* intervention, one requires to obtain the causal graph first.

When modeling the decision structure of a business process with a causal graph, we distinguish between decision variables, and contextual variables, as they play different roles in any analysis. Specifically, decisions may disable other decisions, while variables cannot be disabled and cannot disable decisions. Consider decision D_2 in the running example. Choosing to reject the order will disable decision D_3. This difference between decisions and contextual variables is crucial as it calls for a different treatment and representation.

Decision nodes. Our aim is to model and analyze decisions from a given causal model. To this end, we assume the existence of m decision points that correspond to m variables, $\mathcal{D} = \{D_1, \ldots, D_m\} \subseteq \mathcal{V}$. Examples of decision points include the choice of the next activity

from a set of activities when routing a case, choosing a resource to perform an activity from a set of resources, etc. Decision points can be detected given a business process model e.g., via the method proposed in [6] for control-flow decisions. They can also be mined from an event log, via standard algorithms for process discovery for control-flow decisions [46] or the extraction of resource profiles to identify possible resource assignments [47].

A decision point $D_i \in \mathcal{D}$ takes values in a corresponding discrete bounded domain $\mathbf{D}_i$. For instance, if D_i is the decision on which resource will perform an activity (e.g., activity 'Feature Inspection' of our example), we can define the domain as $\mathbf{D}_i = \{R_1, \ldots, R_l\}$ with l being the number of resources eligible to perform the task. For control-flow decisions, the domain is given by an enumeration of the branches that are activated once the decision has been taken. In our running example, all control-flow decisions $D_1, \ldots, D_4$ happen to be binary, so that their domain is represented as $\mathbf{D}_i = \{0, 1\}, 1 \leq i \leq 4$.

Decision points may be evaluated more than once throughout the execution of the business process. Reasons for that may be control-flow loops or several resource allocations that happen as part of multiple-instance activities. In the remainder, we shall consider recurrent decisions to be distinct decision points. For example, if a loop may take the process back to a previous decision, we shall consider it to be a new decision point.

Contextual variable nodes. In addition to decision points, we also consider contextual variables that may be present and have an impact on the outcomes. We consider the existence of n contextual variables such that $\mathcal{C} = \{V_1, \ldots, V_n\} = \mathcal{V} - \mathcal{D}$. Contextual variables of a decision can include static case features that do not change during execution (e.g., the age of a patient checking in at a hospital), dynamic case features (e.g., body temperature measures of a patient), and inter-case features (e.g., the number of cases in the system). Each variable V_i takes a value from its domain $\mathbf{V}_i$, as we consider all of them to be discrete variables.

For dynamic variables, that potentially change with every event in the log, depending on the causal value they add, multiple instances of them corresponding to their measurements throughout the process must be incorporated into the causal model as separate nodes. The arrangement of these nodes in terms of causal edges is determined by the causal discovery algorithm explained in Section 4.3. In this case, for each dynamic variable V_x there are i measurements, $V_{x,1}, \ldots, V_{x,i}, 1 \leq i \leq k$ with k being the maximal number of measurements allowed for a variable.

In the dealership example, we focus on a single contextual variable, namely the price of the ordered vehicle. While this variable is naturally continuous, its exact value can be considered to be of minor importance for decision support. We therefore discretize it and assume its domain to be $\{1, 2, 3\}$, representing different price levels of the ordered car.

4.2 Flow Analysis for What-If Interventions via Causal Decisions

Next, we demonstrate the usage of our causal model for decision support when computing key performance indicators (KPIs), such as cycle time, and flow times between two events,

under various interventions.

Without interventions, computing the cycle time, for instance, can be performed via exact analysis by computing expectations of flow times using *flow analysis* [48], where a closed-form mathematical expression of the expected values of a KPI can be obtained. To this end, one would not require a causal model. However, if we are interested in an interventional analysis, e.g., a what-if analysis on the impact of various decisions on flow times, one must use a causal model in order to compute these KPIs.

Specifically, the computational formulae adopted in flow analysis will remain the same, but the expected values that one plugs in are different. In fact, they dependent on the estimated P of the causal model, in conjunction with the mutilated causal graph $\mathcal{G}'$ (see Chapter 3).

We illustrate that this approach with our running example. Let X_1 be the duration of activity 'Detailed CDA', and X_2 be the duration of activity 'Standard CDA', and X_3 be the duration of activity 'Deliver Car to Dealership'. Our KPI of interest is the expected flow time F between process start until the car is delivered or until the order is rejected. Without an intervention on D_1, the flow time is, given by the following flow analysis expression:

$$\mathbb{E}(F) = P(D_1 = 1)\mathbb{E}(X_1) + P(D_1 = 0)\mathbb{E}(X_2) + P(D_2 = 0)\mathbb{E}(X_3), \tag{4.1}$$

which are statistical quantities that can be easily estimated from historical data. Alternatively, if we intervene on D_1, say by setting it to 1 (i.e., choosing detailed CDA) and wish to compute the KPI again (thus measuring the impact of this decision), we get the following:

$$\mathbb{E}(F) = \mathbb{E}(X_1) + P(D_2 = 1 \mid do(D_1 = 1))\mathbb{E}(X_3), \tag{4.2}$$

which is a causal expression that cannot be computed without a causal model. However, if we obtain the true causal model (see Fig. 5.1), we can turn this causal expression into a statistical expression:

$$\mathbb{E}(F) = \mathbb{E}(X_1) + \sum_v P(D_2 = 1 | D_1 = 1, V = v)P(V = v)\mathbb{E}(X_3), \tag{4.3}$$

which can again be estimated from historical data. This simple example emphasizes the difference between causal quantities, and statistical quantities when computing the impact of decisions on performance measures.

Flow analysis of pre-interventional KPIs can be performed easily using historical data as it corresponds to the latter. However, post-interventional KPIs can only be computed using a causal model.

4.3 Process-Aware Causal Discovery

In this section, we introduce a novel process-aware causal discovery algorithm. To achieve this, we combine constraint-based discovery from [25] with temporal constraints that we mine from an event log. These temporal constraints are fed into the constraint-based algorithm as additional background knowledge. In this work, we consider two specific types of temporal

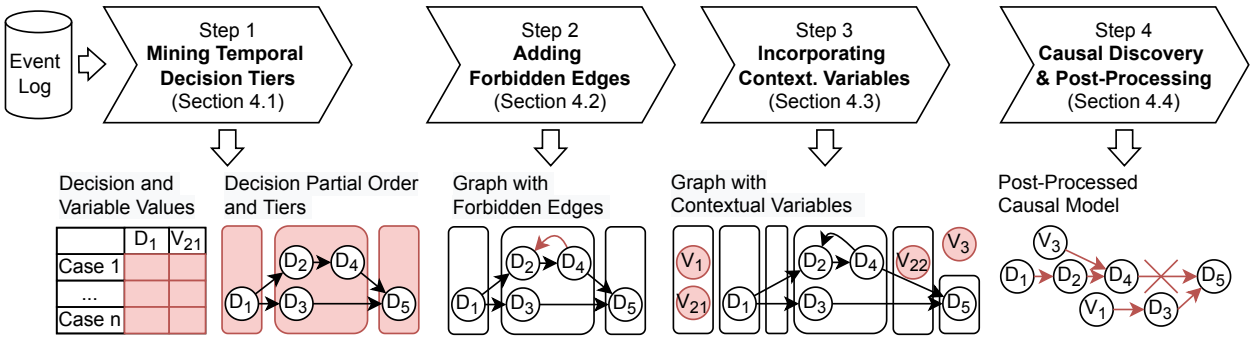


Figure 4.1: Overview of our approach to process-aware causal discovery.

constraints, namely temporal tiers, and forbidden edges. The former type of constraints corresponds to a set of time-ordered tiers indicate the ordering between the variables (decisions and contextual data). Forbidden edge constraints are employed to ensure that variables that always succeed others, yet do not fit into separate tiers, do not parent their predecessors.

Fig. 4.1 presents the steps of our discovery algorithm. Step 1 detects a partial order over decisions, and constructs temporal tiers, such that each tier contains a set of unrelated decisions. Step 2 adds forbidden edge constraints that enforce a temporal ordering within tiers. Step 3 adds a subset of contextual variables from the event log into temporal tiers, when the ordering between variables and decisions can be derived. All other contextual variables remain 'floating' without an assigned tier. Lastly, Step 4 employs the approach from [25] to discover a causal graph, and post-processes the result to remove disabling edges, i.e., edges that induce false causal relations between decisions. We shall now go over the four steps in detail.

4.3.1 Step 1: Mining Temporal Decision Tiers

We start by running a partial order discovery algorithm on the event log to obtain the partial order over all activities and decisions. Here, any algorithm to derive a partial order may be used, for instance the one presented by Kourani and van Zelst [49]. Then, we project the partial order on decision points, which yields a set of pairs of ordered decision points, PO, of the following structure:

$$PO = \{(D_i, D_j), (D_k, D_l), \ldots\} \tag{4.4}$$

where (D_i, D_j) represents a temporal precedence between decisions D_i and D_j.

Using PO, we encode the temporal ordering among decisions into a disjoint set of tiers. Denote by $T_D = \{t_1, \ldots, t_k\}$ the set of tiers, i.e., distinct sets of decisions, $t_i \subseteq \mathcal{D}, 1 \leq i \leq k$ with $t_i \cap t_j = \emptyset, 1 \leq i, j \leq k$ and $i \neq j$, that also respect the partial order:

$$\forall\ t_i, t_j : i < j \iff \forall\ D_x \in t_i\ D_y \in t_j :\ (D_x, D_y) \in PO. \tag{4.5}$$

We construct the tiers T_D by performing a topological sort over the DAG that corresponds to the partial order. The tiers are then constructed from the resulting strongly connected components.

4.3.2 Step 2: Adding Forbidden Edges

When constructing decision tiers some temporal constraints that may exist within the tier may be neglected. To illustrate this scenario, consider the 'Complete Inspection' subprocess of our example in Fig. 1.1, which includes two resource decisions, and a control-flow decision. The top concurrent branch represents the 'Feature Inspection' activity, which can be executed by multiple resources (D_5). Meanwhile, the bottom branch consists of the 'Technical Inspection' activity, also operated by multiple resources (D_6). After technical inspection, there is a decision point regarding whether a 'Technical Fix' is required (D_4). The concurrency between D_5 and $\{D_4, D_6\}$ leads to lack of ordering between the decisions when considering tiers. However, from the temporal order in the log, we can learn that D_4 will never precede D_6.

We incorporate such knowledge into our approach by assessing whether there exists an ordering between decisions within the same tier that is obeyed in all recorded process executions. If so, we inject the obtained knowledge by imposing constraints on forbidden edges that would violate the identified ordering. Specifically, we restrict the existence of edges with an opposite direction with respect to time. In our example, we inform the algorithm that the edge (D_4, D_6) is not permitted.

4.3.3 Step 3: Incorporating Contextual Variables into Tiers

Next, the ordered decision tiers T_D are expanded with contextual variables. Note that it is not required to assign a tier to every variable. Some contextual variables may remain unassigned.

Recall that $\mathcal{C}$ is the set of all contextual variables. We split this set into two disjoint sets corresponding to static and dynamic variables, which we denote by $\mathcal{C}_s$ and $\mathcal{C}_d$, respectively. To distinguish the two sets, we check the variable values throughout the recorded executions in the event log. If for each case in the log, all the measurements of V_x throughout the execution represent a single unchanged value, variable V_x is assigned to $\mathcal{C}_s$; otherwise, it is assigned to $\mathcal{C}_d$.

Having established the two sets, we aim to extend our decision tiers T_D to a set of tiers that would also include contextual variables. To this end, we create new variable tiers between two consecutive decision tiers, which is captured by an extended set $T_{D,V}$ of tiers:

$$T_{D,V} = \{\text{tier}(i) \mid i \in \mathbb{N}_0, i \leq 2k\} \text{ with } \text{tier}(i) = \begin{cases} \emptyset & \text{if } i\%2 = 0 \\ t_{\frac{i+1}{2}} & \text{otherwise.} \end{cases} \tag{4.6}$$

Based on Eq. 4.6, each decision tier (t_i) corresponds to a new tier in the final tier set, $t_i = \text{tier}(2i-1) \in T_{D,V}$. In addition, we create a tier between decision tiers (from T_D), and we can now assign variables from both $\mathcal{C}_s$ and $\mathcal{C}_d$ to tiers.

Packing tiers with static variables. Static variables are case attributes for which the value is set upon the start of case executions. Thus, they are placed into a variable tier that precedes all decision tiers. In other words, we allocate all variables in $\mathcal{C}_s$ to $\text{tier}(0) \in T_{D,V}$.

Packing tiers with dynamic variables. The dynamic variables are not as straightforward to assign. Consider a scenario where we wish to assign a dynamic variable that we observe changing z times throughout a case. We let $V_{x,1}, \ldots, V_{x,z}$ be those z measurements. The objective is to categorize each $V_{x,i}, i = 1, \ldots, z$ into a tier in $T_{D,V}$, necessitating an analysis of its temporal ordering as captured in the event log. This entails recording the timestamp $\tau_{V_{x,i}}$ corresponding to the measurement of variable $V_{x,i}$, as well as the timestamps of the decisions. For each decision tier, we ascertain the timestamps of the earliest and latest decision made by bounding the preceding and succeeding activities that appear before and after the decision, respectively. Subsequently, we evaluate $\tau_{V_{x,i}}$ against this temporal ordering. If it falls within the time range defined by the earliest and the latest decision timestamps for a specific t_i, then $V_{x,i}$ is allocated to tier$(2i-1)$ (a decision tier). Conversely, if it falls between the latest decision timestamp of t_i and the earliest decision timestamp of t_{i+1}, then $V_{x,i}$ is assigned to the variable tier identified by tier$(2i)$.

It is important to note that the tier assignment described above is specific to the variable within a single case. Therefore, there may be temporal inconsistencies when assigning variables when considering multiple cases. Heuristically, we can assure that for a majority of cases, the tier of a contextual variable is consistent. A solution to resolve these inconsistencies in a more systematic manner is left to future work, though.

4.3.4 Step 4: Constraint-based Discovery and Post-processing

Having obtained $T_{D,V}$ and the set of forbidden edges, we feed them along with the dataset containing all variables into a constraint-based algorithm for causal discovery. The result is a causal graph, that in conjunction with P (estimated from the data), can be used to analyze and support decisions.

In some cases, control-flow decisions disable parts of the process. For example, in Fig. 1.1, if decision D_2 selects the bottom branch, the respective process execution will not face decisions $D_3, D_4, \ldots$. This yields a deterministic disablement dependency between the values of D_2 and D_3, which in turn produces a spurious edge in the discovery phase between D_2 and D_3, which should not be there according to Fig. 5.1.[1] Thus, to obtain an accurate causal graph, we must pre-process the data table from Step 1 by adding *null* values to each of the decision domains for decisions that can be disabled. Here, the outcome *null* signifies that the decision was not faced during the execution of the process. Based thereon, we post-process the result of causal discovery that we get in the current step by removing spurious disablement edges.

To this end, we search for decisions that are always disabled whenever upstream decisions take on specific outcomes (e.g., the rejection of an order). Specifically, we consider a decision D_i that is always mutually exclusive with a decision D_j, which comes later in the ordering relation, i.e., $(D_i, D_j) \in PO$. Then, if $\exists\ o \in D_i : P(D_j = null \mid D_i = o) = 1$, we are guaranteed that a deterministic causal edge will appear between D_i and D_j. In that case,

[1]Note that this is a unique feature of the business processes domain that stems from the dependencies between a given choice and its down-stream decisions.

Table 4.1: Tables showing generative probabilities and the final tiers.

Expression	**Probability**
P_v	0.2
$P(d_1\|v)$	$\begin{pmatrix} 0.2 & 0.8 \end{pmatrix}$
$P(d_2\|v, d_1)$	$\begin{pmatrix} 0.3 & 0.8 \\ 0.8 & 0.2 \end{pmatrix}$
$P(d_3\|v)$	$\begin{pmatrix} 0.3 & 0.9 \end{pmatrix}$

(a) Probabilities used to generate data

tier(0)	**tier(1)**	**tier(2)**	**tier(3)**
V	D_1	D_2	D_3

(b) Final tiers

We collect (D_i, D_j) into a set E_{null}, and eventually remove E_{null} from the causal graph. Note that multiple edges leading into D_j may be removed, as multiple upstream decisions can disable D_j.

4.4 Evaluation

In this section, we first test the proposed approach on data generated from our running example (Section 4.4.1). Furthermore, we compare our framework against the method proposed by [11], to highlight the advantages of our causal model. Second, we apply our approach to a real-world dataset, and show that it is able to uncover a causal graph, as well as estimate the ATE for the corresponding decisions (Section 4.4.2).[2]

4.4.1 Synthetic Data for Running Example

First, we assigned conditional probabilities as in Table 4.1a, in correspondence with the model presented in Fig. 5.1 and generated a tabular dataset that consists of 10,000 samples across all variables. Then, we aimed to construct an event log that would correspond to the running example. To this end, we utilized PM4Py [50] to generate event data from BPMN models, using the process from the running example (Fig. 1.1). The choice probabilities in the generative model depended on the previous choices and variables in accordance to the tabular data, and the probabilities in Table 4.1a.

With a synthetic event log in hand, our objective was to execute our process-aware discovery algorithm. To achieve this, we ran the steps in Section 4.3 and obtained the final tier table presented in Table 4.1b.

For causal discovery, we used the *Bayesys Package*[3], which implements the algorithms in [25]. Our approach produced exactly the underlying causal graph depicted in Fig. 5.1 when

[2]An implementation of the proposed approach is available at: `https://github.com/aliash98/Causal-Discovery-for-Business-Processes`

[3]https://bayesian-ai.eecs.qmul.ac.uk/

Table 4.2: Causal effect table.

	Process-Aware Causal Discovery	**Approach from [11]**
ATE of D_1 on D_3	0.0	0.456
ATE of D_1 on D_2	-0.283	-0.165

provided with the dataset, the tier table, and the forbidden edges constraints. In addition, we removed the redundant edge between D_2 and D_3 (Step 4 in Section 4.3).

Utilizing this causal graph to estimate the Average Treatment Effect (ATE) of D_1 on D_3 resulted in an estimate of 0, indicating no causal path from the treatment to the outcome. However, the method proposed in [11], which assumes no additional confounders beyond decision variables reported an ATE of 0.456, demonstrating a significant bias. Specifically, the method in [11] failed to consider the absence of causal paths from D_1 to D_3; instead it produced an erroneous ATE based on correlation. Moreover, employing *CausalNex*[4], we provide the ATE of D_1 on D_2 in Table 4.2. Likewise, in this measurement the estimated ATE is significantly lower than the actual one, due to the ignorance of [11] to the confounding variable (V).

4.4.2 Real-world Data

To demonstrate the applicability of our method to practical problems, we conducted a test on a real-world dataset. Specifically, we selected the 'BPI Challenge 2012' event log, published by the IEEE Task Force on Process Mining, which pertains to a loan process [51]. Using the inductive miner, we obtained a process model depicted in Fig. 6.1 in the Appendix, yielding eight decision points. The decision data corresponding to each case was then collected into a tabular dataset. Additionally, we included the static contextual variable, namely 'Amount Requested', and discretized it into four labels along with the dataset.

Next, following the method described in Section 4.3, we extracted three tiers, with $\{D_3 \rightarrow D_2\}$ being a forbidden edge constraint. Again utilizing the Bayesys package, we identified two causal arcs, namely $discrtized_amount_req \rightarrow D_1$ and $D_2 \rightarrow D_3$.

From [51] and the description of the BPI Challenge 2012 reports, we know that D_1 plays a crucial role in determining which main branch the case will traverse. Returning to Fig. 6.1, branch C indicates approval of the loan request, while branch D represents a rejection thereof. We observe that as the requested amount increases, the likelihood of rejection decreases. This trend could be interpreted as the bank's inclination toward approving larger loan applications. In Table 4.3, we present a summary of the results of intervening on the discretized_amount_req variable and its impact on the distribution of D_1. Additionally, D_2 indicates whether another round of 'Assessing the application' is needed, while D_3 determines the same for 'Seeking additional information during the assessment phase'. The causal

[4]https://github.com/mckinsey/causalnex

Table 4.3: Distribution of D_1 after intervening on discrtized_amount_req

DISC_AMNT_REQ	Branch of D_1				
	A	B	C	D	E
0 (from 0 to 5500)	0.1426	0.0072	0.2412	0.6088	0.0003
1 (from 5500 to 9500)	0.2538	0.0039	0.3170	0.4253	0.0
2 (from 9500 to 19000)	0.2966	0.0043	0.3357	0.3634	0.0
3 (above 19000)	0.3199	0.0058	0.3762	0.2981	0.0

influence of D_2 on D_3, with an Average Treatment Effect (ATE) of 0.513, signifies that the repetition of the assessing task increases the likelihood of seeking additional information.

4.5 Discussion

In this chapter, we proposed a framework for causal decision support that overcomes substantial drawbacks of existing approaches (Section 4.1). Specifically, we showed how the framework enables causal reasoning beyond the state-of-the-art, e.g., in flow analysis (Section 4.2). We then introduced a novel end-to-end method for process-aware causal discovery to facilitate the identification of causal graphs from event data (Section 4.3). It benefits from the temporal order of events, includes both static and dynamic contextual variables, and is not limited to control-flow decisions. Furthermore, we showcased the superiority of our framework against the most sophisticated method currently available by experimenting the calculation of the Average Treatment effect—a pivotal measure in causal inference—on a synthetic data driven from a straight-forward process example. Finally, we demonstrated the application and scalability of our approach through an experiment on a real-world dataset.

Chapter 5

Unidentifiable Causal Queries in Business Processes

In this chapter, we commence by elucidating the significance of latent variables and the identification problem within business processes, using a motivating example (see Section 5.1). Following that, we aim to formally integrate latent variables in the business process causal model, and propose an algorithm that assess identifiability of a causal query (Section 5.2). Finally, we introduce the data acquisition problem, aimed at determining the optimal approach for acquiring data to maximize the utility derived from potential newly identified queries (Section 5.3).

5.1 Identifiability in a Motivating Example

Example 1 - The Loan Application Process. We consider a loan application process, where given a client's credit score and the requested amount, the bank decides to grant or reject the loan. Let D be the respective binary decision variable, with $D = 0$ denoting rejection. The decision is determined by the requested amount (V_1) and by the applicant's credit score (V_2). For simplicity, we assume binary variables, with $V_1 = 1$ indicating a loan request larger than $\$10,000$ and $V_2 = 1$ denoting a credit score greater than 600. In addition, there is an unobserved factor, the client's risk-taking profile, a binary variable with $R = 1$ being risk-tolerant, which influences the decision.

It is worth noting that the choice of variable types, discrete or continuous, does not impact the results in the remainder of this section.

By assuming Fig. 5.1(left) to be the causal graph in the motivating example, we observe that the requested amount (V_1) and the credit score (V_2) are two independent variables causing the decision (D). Thus, we can write:

$$P\left(d, v_1, v_2\right) = P(d \mid v1, v2)P(v1)P(v2). \tag{5.1}$$

As an alternative, consider a scenario in which the value of V_2 is the cause of V_1,

i.e., applicants typically know their credit score before applying for a loan and adjust their asking amount accordingly. Then, Fig. 5.1(middle) would represent the causal model of the decision, which leads to a different DAG:

$$P\left(d, v_1, v_2\right) = P(d \mid v1, v2)P(v1|v2)P(v2). \tag{5.2}$$

In a third scenario, we consider the unobserved variable R that captures the risk profile of the client. In Fig. 5.1(right), it is considered with two directed edges to V_1 and V_2, as a more risk-tolerant applicant would request a larger loan amount and would have a lower credit score. While this leads to the same probabilistic model as in Eq. 5.2, we shall see that the causal interpretation will be different when considering interventions on some of the observed variables.

In our example, using the model from Fig. 5.1(left), queries involving interventions on either V_1 or V_2 can be estimated via

$$P(D = 1 \mid do(V_i = 1)) = P(D = 1 \mid V_i = 1), i \in \{1, 2\}, \tag{5.3}$$

where one can easily estimate $P(D = 1 \mid V_i = 1)$ from historical measurements.

For the model in Fig. 5.1(middle), the estimation of $P(D = 1 \mid do(V_1 = v_1))$ yields a different term. The directed edge from V_2 to V_1 renders V_2 to be a confounder for the two other variables. Using Pearl's do-calculus rules, the non-causal path from V_1 and D is blocked by conditioning on V_2 [17], so that the query is identifiable and can be computed from measurements of D, V_1, and $V2$:

$$\begin{aligned} P(D = 1 \mid do(V_1 = 1)) = {} & P(D = 1 \mid V_1 = 1, V_2 = 1) \cdot P(V_2 = 1) \\ & + P(D = 1 \mid V_1 = 1, V_2 = 0) \cdot P(V_2 = 0), \end{aligned} \tag{5.4}$$

Yet, for the graph in Fig. 5.1(right), an intervention on V_2 is no longer identifiable, i.e., the query $P(D = 1 \mid do(V_2))$ cannot be answered, even with ample amount of data measuring D, V_1, V_2. Surprisingly, the intervention $P(D = 1 \mid do(V_1))$ can be proven to still be identifiable, despite the latent factor R, given by Eq. 5.4. This example emphasizes the need to identify queries prior to their estimation.

For a business manager monitoring processes, understanding the genuine impact of various factors on decision outcomes is of paramount importance. Our example serves to illustrate, firstly, the crucial role of a causal graph in capturing the intrinsic nature of the decision-making process when making inferences about interventions, such as the Average Treatment Effect. Secondly, it underscores the undeniable impact that hidden or unmeasured factors can have, emphasizing their significance in the overall analysis.

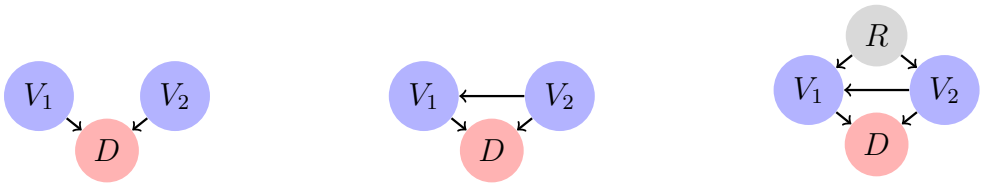


Figure 5.1: Three causal graphs that could account for the loan request decision D.

5.2 Latent Variables in Business Process Causal Models

Having established the significance of latent variables in causal inference within a business context, we aim to acknowledge unmeasured variables in the causal framework proposed in Section 4.1. Employing the domain knowledge, we augment the graph $\mathcal{G}$ in our causal model to include $\mathcal{V}_{um}$ and $\mathcal{E}_{um}$, the set of unmeasured variables and its corresponding edges needed to be incorporated into the business process causal graph. The new DAG accounting for the process would be denoted as $\mathcal{G}_{dk} = (\mathcal{V} \cup \mathcal{V}_{um}, \mathcal{E} \cup \mathcal{E}_{um})$.

In the motivating example proposed in Section 5.1, let us assume the Fig. 5.1 (middle) as the discovered graph $\mathcal{G}$ from the set of observed business process variables $\mathcal{V} = \{V_1, V_2, D\}$. Therefore, $\mathcal{G}$ along with P which summarizes the observational dataset form the business process causal model. Having the process reviewed by an expert, a set of unmeasured variables $\mathcal{V}_{um} = \{R\}$ along with the corresponding edges $\mathcal{E}_{um} = \{(R, V_1), (R, V_2)\}$ are incorporated to the initial $\mathcal{G}$ resulting an augmented graph $\mathcal{G}_{dk} = (V \cup V_{um}, \mathcal{E} \cup \mathcal{E}_{um})$, which is depicted in Fig. 5.1 (right).

As pointed out in Section 3.1, the problem of identification for causal queries arises when unmeasured variables are added to the analysis. In the following subsections we first propose an algorithm to deal with the identification problem. Then, we aim to estimate the causal query if the query turned out identifiable, and otherwise, we discover the possible options for acquiring data that makes the query identifiable.

5.2.1 Identifiability and Estimation of Causal Decision Queries

We aim to focus on causal queries, in which the target is a decision variable $D \in \mathcal{D}$, the intervention variable is not empty, i.e., $I \neq \perp$, and identifiability should be reviewed, i.e., unmeasured variables exist. We refer to such queries as *interventional decision queries* (IDQs,

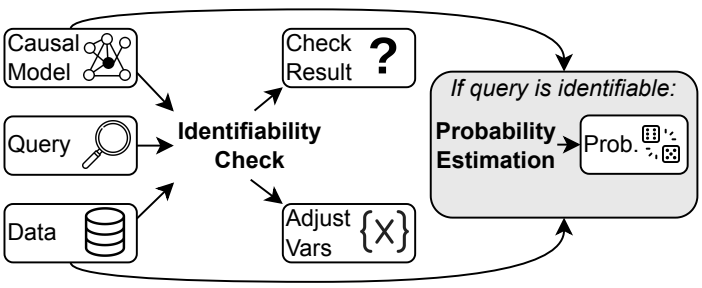


Figure 5.2: Identifiability and estimation.

Algorithm 1 Identifying a query from a dataset.

function IDENTIFY($q = (D, \mathcal{C}, I), \mathcal{G}_D, \mathcal{V}$)
  $\mathcal{G}_{D,SM} \leftarrow ProjectToSemiMarkovian(\mathcal{G}_D, \mathcal{V})$
  $\mathcal{Z}_q = \emptyset$
  if IDC($q, \mathcal{G}_{D,SM}$) is True **then**
    return True, $\mathcal{Z}_q$
  else
    for $\mathcal{A} \in \mathcal{P}(\mathcal{V}_{um,D})$ **do**
      $\mathcal{G}'_{D,SM} \leftarrow ProjectToSemiMarkovian(\mathcal{G}_D, \mathcal{V} \cup \mathcal{A})$
      if IDC($q, \mathcal{G}'_{D,SM}$) is True **then**
        $\mathcal{Z}_q \leftarrow \mathcal{Z}_q \cup \mathcal{A}$
      end if
    end for
    return False, $\{\mathcal{A} \in \mathcal{Z}_q \mid \nexists\, \mathcal{B} \in \mathcal{Z}_q : \mathcal{B} \subset \mathcal{A}\}$
  end if
end function

$q = (D, I)$). Note that if $I = \perp$, the query is predictive and not interventional, since it does not induce a change to the causal graph. Such predictive decision queries can be answered using standard predictive monitoring and decision mining methods, e.g., [52]. Since a historical dataset summarized by P corresponding to observed business variables $\mathcal{V}$ along with the set of unmeasured variables $\mathcal{V}_{um}$ is available, we can assess the identifiability of the query, as illustrated in Fig. 5.2. If the query turns out to be identifiable, we proceed with the probability estimation.

Identifiability. Alg. 1 outlines our approach to check whether the query is identifiable. In addition to checking the result, the algorithm returns a superset of variables that outlines different ways to resolve the issue and achieve identifiability. Specifically, the so-called adjustment superset, $\mathcal{Z}_q$, is a collection of $n(q)$ variable sets $\mathcal{Z}_q = \{\mathcal{Z}_{q,1}, \ldots, \mathcal{Z}_{q,n(q)}\}$ and at least one of the sets in $\mathcal{Z}_q$ must be added to the dataset in order to identify the query. Each set in the adjustment set is minimal, i.e., a set $\mathcal{Z}_{q,i} \in \mathcal{Z}_q$ is not a subset of any other set $\mathcal{Z}_{q,j} \in \mathcal{Z}_q, j \neq i$.

The algorithm employs two methods from causal theory: (i) a *semi-Markovian projection* of a causal graph, referred to as *ProjectToSemiMarkovian* in Alg. 1; it takes a causal graph and a set of observed variables as input, and removes the unobserved variables from the causal graph, replacing them with bi-directional arcs [53], and (ii) *IDC*, an algorithm that identifies causal queries based on semi-Markovian causal graphs [26].

Here, for convenience, we extract the sub-graph including all the influential factors on D from $\mathcal{G}_{dk}$ as follows:

Definition 3 (Contextual Projection)**.** *Given a decision D, the projection $\pi(\mathcal{G}_{dk}, D)$ returns a subgraph $\mathcal{G}_D = (\mathcal{V}_D, \mathcal{E}_D)$, where $\mathcal{V}_D$ contains all ancestors of D, and the set $\mathcal{E}_D$ contains only edges from $\mathcal{E}$ that connect these ancestors.*

We refer to the resulting graph as the *business process causal decision model* (BP-CDM) and to $\mathcal{V}_D$ as the context of D.

The following lemma ensures that the BP-CDM remains Markovian, i.e., the result of the projection is a DAG.

Lemma 1. *If $\mathcal{G}$ is a DAG, the subgraph $\mathcal{G}_D = \pi(\mathcal{G}, D)$ is also a DAG.*

Proof. By way of contradiction, suppose to the contrary that the directed graph $\mathcal{G}_D$ contains a cycle of length $k-1$, $(V_1, V_2), (V_2, V_3), \ldots, (V_{k-1}, V_k)$. Since π does not add new edges to $\mathcal{E}_D$, this cycle must have existed in $\mathcal{G}$. Yet, $\mathcal{G}$ is acyclic. □ □

In line 1 of Alg. 1, we project the original graph ($\mathcal{G}_D$) into a semi-Markovian graph ($\mathcal{G}_{D,SM}$) using only the observed variables, $\mathcal{V}$. Then, we employ the IDC algorithm to scan the semi-Markovian projection, $\mathcal{G}_{D,SM}$, and determine whether q is identifiable (line 3). If the query is identifiable, IDC returns true, making the output of Alg. 1 true, and the adjustment superset empty $\mathcal{Z}_q = \emptyset$ (line 4).

If IDC returns false, we consider every element $\mathcal{A}$ in the power set of all unobserved variables, $\mathcal{P}(\mathcal{V}_{um,D})$ and test whether the addition of $\mathcal{A}$ would make q identifiable (lines 7-11). To this end, we construct the corresponding semi-Markovian graph with the addition of $\mathcal{A}$ (line 8) and re-run IDC (line 9). Subsequently, in line 10, we collect the elements of the power set for which IDC returns True into the adjustment superset $\mathcal{Z}_q$. After iterating over the power set, we obtain the adjustment superset $\mathcal{Z}_q$, which may not be minimal. In line 13, we return False (as the original query q is not identifiable), together with the minimal adjustment superset. Here, we enforce minimality of each set in $\mathcal{Z}_q$.

Running Alg. 1 on $q = (D, do(V_2 = 1))$ from Fig. 5.1 yields a False response, and an adjustment superset, $\mathcal{Z}_q = \{\{R\}\}$, since there is only one possible adjustment set.

Estimating the effect of interventions on decisions. Having established the identifiability of interventional decision queries, we now turn to their estimation using historical data. Given an identifiable query $q = (D, I)$, we estimate its probability distribution as follows:

$$\hat{P}(D \mid do(I = i)) = \hat{f}(z, i)), \tag{5.5}$$

with $\mathcal{Z}_{q,i} = z$ being one minimal set of variables on which we must condition to 'remove' the do-operator and turn the causal quantity into an observational one.

In the motivating example, the query $P(D = 1 \mid do(V_1 = 1))$ can be estimated by estimating each of the components of Eq. 5.4, e.g., using logistic regression. The latter can be replaced by any other supervised learning approach.

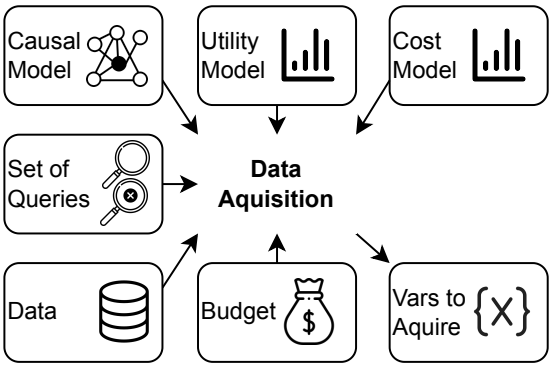


Figure 5.3: Data acquisition problem overview.

5.3 DAP: Data Acquisition for Unidentified Queries

In this part, we propose a data acquisition method as sketched in Fig. 5.3, which relies on the business process causal model equipped with unmeasured variables. As elucidated earlier, among all queries one can formulate given a causal model, depending on the existence of hidden variables, some queries might end up unidentifiable. Therefore, we wish to acquire additional hidden variables that would expand the set of queries that can be identified. In the acquisition process, our goal is to balance between the utility we gain from identifying additional queries, and the costs associated with measuring new variables.

We formulate the data acquisition problem (DAP) in the context of interventional decision support and show that it can be viewed as a generalization of the set-union knapsack problem (SUKP) [54], and as a new variation of the well-known knapsack problem. We also aim to prove that DAP is $\mathcal{NP}$-hard. Additionally, we aim to consider practical applicability of DAP to real-world settings.

5.3.1 Problem Definition

Prior to defining DAP, we provide useful notation and problem primitives.

Assume that we are provided with the following three parameters: (i) a real-valued cost mapping of acquiring a variable V, $c(V) \in \mathbb{R}^+$; (ii) a data acquisition budget $B \in \mathbb{R}^+$ (that limits the total acquisition costs); and (iii) a real-valued utility function $u(q) \in \mathbb{R}^+$ that maps interventional queries to the expected benefit from their identification.[5] Let $\mathcal{Q}^-$ be the subset of interventional queries that cannot be identified from $\mathcal{V}$. Thus, we aim to acquire a set of unmeasured variables $\mathcal{V}_a \subseteq \mathcal{V}_{um}$, while limiting the total acquisition cost by B. Let $\mathcal{Q}(\mathcal{V}_a)$ denote the set of queries from $\mathcal{Q}^-$ that can be identified by acquiring $\mathcal{V}_a$, i.e.,

$$\mathcal{Q}(\mathcal{V}_a) = \{q \in \mathcal{Q}^- \mid \exists \mathcal{Z} \in \mathcal{Z}_q : \mathcal{Z} \subseteq \mathcal{V}_a\}. \tag{5.6}$$

In other words, every query in $\mathcal{Q}(\mathcal{V}_a)$ is 'adjusted' by $\mathcal{V}_a$. Moreover, the total utility of

[5]We discuss how one can obtain or compute these parameters in Section 5.3.2.

identifying a subset of queries, $\mathcal{Q}$, is defined as

$$U(\mathcal{Q}) = \sum_{q \in \mathcal{Q}} u(q). \tag{5.7}$$

With this notation, we are ready to define the data acquisition problem (DAP).

Problem 1 (DAP)**.** *Find a subset $\mathcal{V}_a \subseteq \mathcal{V}_{um}$ such that the set of queries identified, $\mathcal{Q}(\mathcal{V}_a)$ maximizes the total utility $U(\mathcal{Q}(\mathcal{V}_a))$, while remaining within budget, i.e., $\sum_{V \in \mathcal{V}_a} c(V) \leq B$.*

Below, we prove the computational complexity of DAP, and then, provide an integer linear programming formulation that solves the problem.

Theorem 1. *The computational complexity of DAP is $\mathcal{NP}$-hard.*

Proof. We only need to show that a relaxed version of the DAP is $\mathcal{NP}$-hard, i.e., the DAP is at least $\mathcal{NP}$-hard. If every query q has exactly one adjustment set (special case of DAP), $\mathcal{Z}_q = \{\mathcal{V}_{a,q}\}$, the problem reduces to the set-union knapsack problem (SUKP), which is known to be $\mathcal{NP}$-hard [54]. Since a simplified case of DAP is SUKP, it must be at least as hard as SUKP. To show that DAP is SUKP for single adjustment set per query, we consider queries to be items to be collected into a knapsack. Each query q is viewed as an item that has some utility ($u(q)$), and is associated with a set of elements (the adjustment set of q) that must be packed. Each element $V \in \mathcal{V}_{a,q}$ has a weight ($c(V)$). SUKP aims to maximize the total utility from collecting the items, while remaining within budget.

Integer linear programming formulation. In order to solve the problem optimally, we provide an integer linear programming formulation of the DAP. It is worth noting that the ILP problem is known to be $\mathcal{NP}$-hard in the general case [55]. Here, for short-hand notation, we use lowercase letters for sets. Let $x_v, v \subseteq \mathcal{V}_{um}$ be $0-1$ decision variables that equal 1 if a subset v is acquired. Note that we can acquire variables up to budget B, and we will acquire only a single variable set. We can precompute an indicator $a_{q,v} = 1$ if query q is adjusted by v (i.e., $\exists\ \mathcal{Z} \in \mathcal{Z}_q : \mathcal{Z} \subseteq v$). The ILP is given by,

$$\begin{aligned}
\underset{\mathbf{x}}{\text{Maximize}} \quad & \sum_{q \in \mathcal{Q}^-} u(q) \sum_{v \subseteq \mathcal{V}_{um}} a_{q,v} x_v \\
\text{subject to:} \quad & \sum_{v \subseteq \mathcal{V}_{um}} x_v \sum_{V \in v} c(V) \leq B \\
& \sum_{v \subseteq \mathcal{V}_{um}} x_v = 1 \\
& x_v \in 0, 1 \quad \forall\ v \subseteq \mathcal{V}_{um}
\end{aligned}$$

□

□

5.3.2 Applying DAP in Practice

Next, we discuss two practical considerations when solving DAP. We start by introducing a greedy heuristic that overcomes the computational complexity of the underlying knapsack problem, and conclude the section by considering methods for estimating DAP's parameters in real applications.

Greedy heuristic. We propose a heuristic solution for DAP inspired by the greedy heuristic for SUKP [54]. Let $\mathcal{Z}_{total} = \bigcup_{q \in \mathcal{Q}^-} \mathcal{Z}_q$ be the set of all adjustment sets. Moreover, let $f_u(\mathcal{Z})$ be the total utility an adjustment set, $\mathcal{Z} \in \mathcal{Z}_{total}$, yields by transforming a subset of queries identifiable, $U(\mathcal{Q}(\mathcal{Z}))$. More formally, and let $C(\mathcal{Z})$ be the total cost of the adjustment set. In each iteration we order members of $\mathcal{Z}_{total}$ in a descending order by $f_u(\mathcal{Z})/C(\mathcal{Z})$ and select the adjustment set corresponding to the highest value that is within our budget B. Denote this set by $\mathcal{Z}_{max}$.

First, we eliminate the adjustment sets that are associated with queries in $\mathcal{Q}(\mathcal{Z}_{max})$ from $\mathcal{Z}_{total}$. Also, all subsets of $\mathcal{Z}_{max}$ should be removed from $\mathcal{Z}_{total}$, and the new budget is the previous value subtracted from $C(\mathcal{Z})$. After each iteration, the size of $\mathcal{Z}_{total}$ reduces, and the computation of $f_u(\mathcal{Z})/C(\mathcal{Z})$ and the subsequent ordering must be accomplished again. We repeat these steps until the budget is exhausted or all queries become identifiable.

Estimating DAP parameters. DAP depends on three types of parameters: the costs, the budget, and utilities. Estimating the costs of data acquisition is straightforward. For each variable, there must be a data collection mechanism for which the induced cost can be estimated. If the data cannot be acquired, then the cost will be set to ∞. Similarly, the budget for acquiring new variables should be decided by the user of the approach. The most interesting and nontrivial question is how to compute the utility of identifying a causal query. The benefit from identifying an interventional decision query may vary and depend on the application and the impact of the decision on the overall business process. We propose to consider two sources of benefit from query identification:

- *Estimation using observational data.* Identified queries can be estimated using observational data. Specifically, it allows us to save a costly or infeasible controlled experiment to estimate the impact of $do(I = i)$ on D. For example, to measure the impact of changing credit scores on outcomes via a controlled experiment, we would need to divide the applying clients into two sets (at random), and intervene on only one of the sets. This can be a costly, and sometimes infeasible procedure. Alternatively, if we can identify the corresponding query from observational data by acquiring information regarding the risk-taking behavior of the user (e.g., via a survey), we may be able to estimate the effect without requiring a controlled experiment.

- *Effectiveness of decision-making.* Accurately estimating the effect of an intervention for improving our decision, we can save future costs. For example, if, according to our estimation, reducing the loan amount compared to the sum that is requested by a client can lead to the decision to grant the loan (as there is a higher chance of the client returning it).

Quantifying the two sources of utility requires a return-on-investment (ROI) analysis that would demonstrate the impact of being able to estimate the query.

5.4 Evaluation

In this part, we start by providing details on the implementation of the algorithm in Section 5.2, followed by an application of our approach to a real-world case study involving an emergency department in Toronto, Canada.

5.4.1 Implementation

An implementation of the proposed approach (in Python) is available in a Github repository.[6] Specifically, we implemented Alg. 1 as follows. The *ProjectToSemiMarkovian* function was implemented using the *NetworkX* library [56]. The IDC function was adapted from the *YLearn* package.[7] After feeding the code with a causal graph, for an unidentifiable query, our solution returns False together with the minimum adjustment superset required to obtain identifiability as output. Moreover, after the algorithm correctly evaluates a query as identifiable, it uses Bayesian network estimation as the corresponding statistical estimand. Lastly, given query utilities, variable costs, and a budget, we heuristically solve the data acquisition problem.

To validate the approach on a simple test case, we fed the algorithm with the causal graph in Fig. 5.1 (right). We assume discrete probability distributions for each node and generated the corresponding data accordingly ($10,000$ samples). Based on the statistical estimand returned by the IDC algorithm, we first estimated it by using data statistics without training a model (to be used as ground truth). Subsequently, we employed machine learning models such as linear regression, logistic regression, and a built-in estimator from the *DoWhy* package [57] to obtain estimates for the various queries in the running example.

5.4.2 The Case of ED-Anonymous

After validating our algorithms on our running example, we turned to apply it to a real-world scenario. ED-Anonymous is an emergency department of Anonymous hospital operating in the city of Toronto (Canada). Patients are triaged by a nurse into 5 acuity levels, and then move to an assessment by an attending ED physician. Based on a physical assessment, the outcome of triage and the medical history, the attending doctor decides whether they call for an additional examination by a consulting physician. The consultants work in different departments (psychiatry, cardiology, etc.), and visit the patient at the ED when they have time to do so.

[6] `https://github.com/aliash98/BPM_CausalReasoning`

[7] `https://github.com/DataCanvasIO/YLearn`

The ED manager informed us that patients who were seen by a consultant physicians had (often unnecessary) long cycle times (time between entry and exit), since consultants are often delayed in their own wards and do not arrive to the ED for an examination in a timely manner. In the first stage, we obtained an initial dataset that described patient visits to the ED and verified that indeed, consult patients are delayed longer (regardless of their triage code). Every patient visit in the data had event-related attributes, namely a unique visit id, a patient id, an entry timestamp and an exit timestamp. In addition, we obtained patient-related attributes, including the patient's age, gender, triage code (how severe the patient is on a scale of 1, most acute, to 5, most non-acute), the arrival mode (arrived by walking or by ambulance), and a boolean variable 'isConsult' that equals 1 when the patient required a consultant.

The intervention of interest that the ED manager proposed is to fix the triage score of walking young adult patients (for ages between 18 and 30) to 5, since they are typically not severe. The manager wished to measure the impact of the intervention on the probability to invite a consulting physician for such patients. Indirectly, this intervention can reduce the length-of-stay of patients in the ED, which is the key performance indicator measured by the government.

After interviewing the ED manager, we obtained the causal graph that best describes the underlying triage and assessment process (see Fig. 5.4). Gray nodes correspond to hidden variables (Medical History and Physical Assessment), blue nodes correspond to variables available in the dataset, and the orange node (Triage Score) corresponds to the intervention. The red node, IsConsult, is the decision variable. The query of interest is in essence,

$$P(IC = 1 \ \mid \ AM = `Walking' \ , \ 18 \leq AG < 30 \ , \ do(TC = 5)). \tag{5.8}$$

Running Alg. 1 on the initial data we discovered that the query is not identifiable, and that the two possibilities to identify it were to obtain the preliminary assessment data, or to acquire the patient's medical history. The outcome of the results of the preliminary assessment (which is mostly a physical assessment) are always recorded by the information system, yet, we were not given access to it in the initial dataset. Medical history is considered highly private information, which was not accessible as part of our project. Thus, setting $C(MH) = \infty$, we get that the only feasible acquisition is to ask a data analyst in the hospital to write an SQL query.

In the second stage, ED-Anonymous provided the preliminary assessment variable, and we were able to quantify the query. In fact, comparing the effect of fixing TC to 5 for all young adults arriving independently compared the current state of the ED can be computed as the difference between two probabilities (each can now be estimated from the observational data):

$$\begin{aligned} &P(IC = 1 \mid AM = `Walking' \ , \ 18 \leq AG < 30 \ , \ do(TC = 5)) \\ &- P(IC = 1 \mid AM = `Walking', 18 \leq AG < 30). \end{aligned} \tag{5.9}$$

Estimating the probabilities from data using maximum likelihood estimation we get that the interventional probability of calling a consult is 0.0322, while the latter is 0.0791, which signals

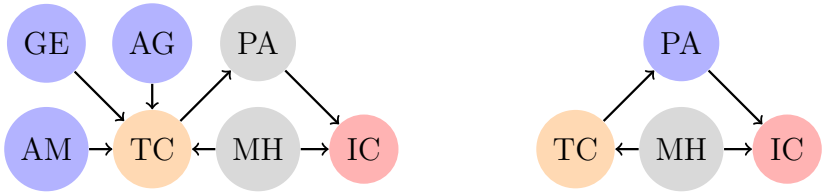


Figure 5.4: Left: the causal graph of ED-Anonymous; contextual variables: Age (AG), Gender (GE), Arrival Mode (AM), Medical History (MH), Physical Assessment (PA); intervention variable Triage Code (TC); decision variable: Is Consult (IC). Right: The mutilated causal graph after an intervention on TC and acquisition of PA.

a decrease in the need for a consultant in 4.7% of the times thus significantly decreasing the length of stay of these patients.

5.5 Discussion

Throughout this chapter, we argued that in some processes, certain variables are inherently hidden, rendering them impossible to record (Section 5.1). In some other cases, the absence of critical variables are simply due to the lack of measurement. Nonetheless, we extended our causal model framework to take additional latent variables, sourced from domain knowledge, into account. We then developed an algorithm to investigate the identifiability of causal queries in the presence of such variables. This marks the first instance where latent variables and the identification of causal queries—a critical subject with a high potential impact—are introduced to the community of business process management. Previous research in this field primarily was conducted under the assumption of *no hidden confounder*, a premise that this thesis convincingly demonstrates can lead to false results.

Following that, we proposed and implemented an algorithm to assess the identifiability of a given causal query, and to return possible adjustment sets if the query is not identifiable. We extensively discussed how our extended framework facilitates reasoning for identification, prediction, and estimation.

Furthermore, we demonstrated how to address situations where queries are unidentifiable due to unobserved data. For such cases, we formulated the Data Acquisition Problem (DAP) as an optimization problem, balancing the cost of acquiring data with the utility gained from it. We discovered that it represents a special variation of the knapsack problem that has not been previously studied. After analyzing the computational cost of optimally solving this problem, we proposed a heuristic to mitigate such costs in larger contexts.

Finally, we demonstrated the feasibility of the extension proposed in this chapter through implementation and experiments on synthetic and real-world datasets.

Chapter 6

Discussion and Limitations

Throughout this chapter, we establish the application and significance of the introduced methods put forth in previous chapters. We first review the framework as a whole, then, discuss each of its components to identify limitations and drawbacks and explore potential strategies to resolve them. In the end, we draw a comprehensive conclusion and include discussion of future work.

Framework overview. We introduced a business process causal model (Section 4.1), and proposed a novel process-aware method to obtain it from observations captured in event log (Section 4.3). It enables the cause-effect analysis of decisions, specifically when contextual variables are recorded. We illustrated a potential application for flow analysis through our running example when a decision directly influences a key performance indicator (KPI) (Section 4.2). Additionally, as an extension, we proposed a method to augment the causal graph with new variables not recorded in the dataset and developed an algorithm to address the identifiability problem and ultimately estimate the identifiable queries (Section 5.2). We also defined and analyzed the DAP, an optimization problem, to tackle cases where unidentifiable queries exist, and multiple options for data acquisition exist to resolve this issue (Section 5.3).

Business process causal model. The applications of a business process causal model are not limited to what discussed before, and can be extended to broader contexts. All discussions in Chapter 4 and Chapter 5, along with examples presented, focused solely on *atomic* causal queries—indicating the manipulation of a single variable. However, with the causal graph in hand, *non-atomic* interventions can also be accommodated. Furthermore, more sophisticated causal analyses, such as conditional intervention or Conditional Average Treatment Effect (CATE), may be facilitated by having a faithful causal graph extracted from the data. To conduct such analyses, one should consider counterfactual analysis, which relies on Structural Causal Models (SCMs) [10].

While variables related to Key Performance Indicators (KPIs) may not be directly included in the causal model, they can be integrated into the causal graph. For instance, one could calculate the cycle time associated with each trace and either include this data in the dataset or manually add it to the model. This integration facilitates the direct measurement of the

effect of interventions on KPIs.

Process-aware causal discovery. We leverage various types of information concerning decision variables and temporal data encoded in timestamps within event logs, in the discovery phase. By doing so, our objective is to reduce the uncertainty associated with the resulting causal model. Among the few previous works that have explored causal dependence discovery in business processes, none have capitalized on both contextual variables and temporal knowledge in causal graph discovery. Furthermore, this work is pioneering in its field by considering multiple measurements of dynamic variables within the causal model.

We initially limited our support to discrete variables as an assumption before conducting causal model discovery. This may be viewed as a limitation, particularly because business contextual variables, especially dynamic variables, may naturally be continuous (e.g., patient temperature in the emergency room). However, we made this modeling choice because causal discovery algorithms for continuous variables based on the additive noise assumption come with certain drawbacks. Most notably, without specific assumptions on the noise distribution, particularly non-Gaussian distributions, the algorithm may struggle to distinguish among a set of Markov equivalent graphs [10, Chapter 4]. Conversely, if all variables are discrete, according to [58], the additive noise setting requires no additional assumption to indicate the causal direction between each pair of variables, and thus, a more robust conclusion could be driven. Nevertheless, one can deal with continuous variables by discretizing them in the data set while extracting them from event log.

Given that the discovery phase heavily relies on the data collected in the event log, incomplete event logs—those with missing measurements of contextual variables, for instance—may lead to inaccurate causal models. As our framework does not support missing values in the dataset, we recommend a preprocessing step on the dataset derived from the event log to handle missing values using data imputation methods [59].

DAP. One limitation of the Data Acquisition Problem (DAP) is the potential challenge of modeling utility functions for different settings. A previous work [60] has explored deriving a measure for stakeholder utility that can be applied in a broad range of settings. Alternatively, one could attempt to model utility functions based on the relationship between causal queries and Key Performance Indicators (KPIs), as discussed in Section 4.2. Similarly, despite the solutions suggested in Section 5.3.2, modeling the costs of acquiring variables with respect to each other may still present difficulties. This problem has been widely studied in similar settings; For example, [61] proposes formulations based on Support Vector Machines for simultaneous classification and feature selection that explicitly incorporate attribute acquisition costs.

Conclusion. In this paper, we argued that improving decision-making is essential for the success of organizations and proposed a framework for causal decision support that overcomes substantial drawbacks of existing approaches. Specifically, we showed how the framework enables causal reasoning beyond the state of the art, e.g., in flow analysis. We also introduced a new method for process-aware causal discovery to facilitate the discovery of causal graphs from event data. We then, included the possibility of unmeasured variables into

our causal model, handling the identification problem rising as a consequence. Furthermore, we introduced a new technique for data acquisition to facilitate the estimation of the effect of interventions for unidentifiable decisions. Finally, we demonstrated the feasibility of our approach through an implementation, synthetic experiments, and a real-world dataset application.

Future work. We intend to expand our framework by explicitly incorporating objective functions and utility variables that can then be optimized. Additionally, we aim at adding declarative temporal constraints into constraint-based discovery to improve the results of the causal discovery approach. We also wish to test the approach on a real-world dataset where the ground-truth causal graph and the true BPMN model are known. This would demonstrate the accuracy and usefulness both of the proposed framework and of the discovery algorithm. Additionally, we plan to move away from the current greedy approach used for solving DAP, and instead implement a general dynamic programming solution and assess its usefulness in real-world scenarios.

Bibliography

[1] Ekaterina Bazhenova et al. “From BPMN process models to DMN decision models”. In: *Information Systems* 83 (2019), pp. 69–88.

[2] Shaofeng Liu et al. “Integration of decision support systems to improve decision support performance”. In: *Knowledge and Information Systems* 22 (2010), pp. 261–286.

[3] Anne Rozinat and Wil MP van der Aalst. “Decision mining in ProM”. In: *Business process management* 4102 (2006), pp. 420–425.

[4] Frada Burstein, Clyde W Holsapple, and Wil MP van der Aalst. “Decision support based on process mining”. In: *Handbook on Decision Support Systems 1: Basic Themes* (2008), pp. 637–657.

[5] Felix Mannhardt et al. “Decision mining revisited-discovering overlapping rules”. In: *Advanced Information Systems Engineering: 28th International Conference, CAiSE 2016, Ljubljana, Slovenia, June 13-17, 2016. Proceedings 28*. Springer. 2016, pp. 377–392.

[6] Ekaterina Bazhenova and Mathias Weske. “Deriving decision models from process models by enhanced decision mining”. In: *Business Process Management Workshops: BPM 2015, 13th International Workshops, Innsbruck, Austria, August 31–September 3, 2015, Revised Papers 14*. Springer. 2016, pp. 444–457.

[7] Ekaterina Bazhenova, Susanne Buelow, and Mathias Weske. “Discovering decision models from event logs”. In: *Business Information Systems: 19th International Conference, BIS 2016, Leipzig, Germany, July, 6-8, 2016, Proceedings 19*. Springer. 2016, pp. 237–251.

[8] Ekaterina Bazhenova et al. “Discovery of fuzzy DMN decision models from event logs”. In: *Advanced Information Systems Engineering: 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings 29*. Springer. 2017, pp. 629–647.

[9] Johannes De Smedt et al. “Holistic discovery of decision models from process execution data”. In: *Knowledge-Based Systems* 183 (2019), p. 104866.

[10] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. Cambridge, Massachusetts: The MIT Press, 2017.

[11] Sander JJ Leemans and Niek Tax. "Causal reasoning over control-flow decisions in process models". In: *Advanced Information Systems Engineering: 34th International Conference, CAiSE 2022, Leuven, Belgium, June 6–10, 2022, Proceedings*. Springer. 2022, pp. 183–200.

[12] Gregory F. Cooper and Clark Glymour, eds. *Computation, Causation, and Discovery*. 570 pp., 6 x 9 in. Menlo Park, CA: MIT Press, 1999. ISBN: 9780262571241.

[13] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, prediction, and search*. Jan. 2001. DOI: 10.7551/mitpress/1754.001.0001. URL: https://doi.org/10.7551/mitpress/1754.001.0001.

[14] Judea Pearl. *Causality: Models, reasoning, and inference*. Cambridge University Press, 2009.

[15] Diego Colombo and Marloes H. Maathuis. "Order-Independent Constraint-Based Causal Structure Learning". In: *Journal of Machine Learning Research* 15 (2014), pp. 3741–3782.

[16] Peter Spirtes and Kun Zhang. "Causal Discovery and Inference: Concepts and Recent Methodological Advances". In: *Applied Informatics*. Vol. 3. SpringerOpen, 2016, pp. 1–28.

[17] Judea Pearl. "The do-calculus revisited". In: *arXiv preprint arXiv:1210.4852* (2012).

[18] Diego Colombo and Marloes H. Maathuis. "Order-independent Constraint-based Causal Structure Learning". In: (2013).

[19] Peter L. Spirtes, Christopher Meek, and Thomas S. Richardson. "Causal Inference in the Presence of Latent Variables and Selection Bias". In: (2013).

[20] Juan I Alonso-Barba, Jose A Gámez, Jose M Puerta, et al. "Scaling up the Greedy Equivalence Search Algorithm by Constraining the Search Space of Equivalence Classes". In: *International Journal of Approximate Reasoning* 54.4 (2013), pp. 429–451.

[21] Alessio Zanga and Fabio Stella. "A Survey on Causal Discovery: Theory and Practice". In: (2023). arXiv: 2305.10032 [cs.AI].

[22] Ana Rita Nogueira et al. "Methods and Tools for Causal Discovery and Causal Inference". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* e1449 (2022), pp. 1–28.

[23] Philippe Brouillard et al. "Typing assumptions improve identification in causal discovery". In: *arXiv (Cornell University)* (July 2021). URL: https://arxiv.org/pdf/2107.10703.pdf.

[24] Bryan Andrews, Peter Spirtes, and Gregory F. Cooper. "On the Completeness of Causal Discovery in the Presence of Latent Confounding with Tiered Background Knowledge". In: *International Conference on Artificial Intelligence and Statistics* (June 2020), pp. 4002–4011. URL: http://proceedings.mlr.press/v108/andrews20a/andrews20a.pdf.

[25] Anthony C. Constantinou, Zhiwei Guo, and Neville K. Kitson. "The impact of prior knowledge on causal structure learning". In: *Knowledge and Information Systems* 65.8 (Apr. 2023), pp. 3385–3434. DOI: 10.1007/s10115-023-01858-x. URL: https://doi.org/10.1007/s10115-023-01858-x.

[26] Ilya Shpitser and Judea Pearl. "Complete identification methods for the causal hierarchy". In: *Journal of Machine Learning Research* 9 (2008), pp. 1941–1979.

[27] Jin Tian. "Identifying conditional causal effects". In: *arXiv (Cornell University)* (July 2004), pp. 561–568. DOI: 10.5555/1036843.1036911. URL: https://arxiv.org/pdf/1207.4161.

[28] Arik Senderovich et al. "Mining resource scheduling protocols". In: *Business Process Management: 12th International Conference, BPM 2014, Haifa, Israel, September 7-11, 2014. Proceedings 12*. Springer. 2014, pp. 200–216.

[29] Kimon Batoulis et al. "Enabling dynamic decision making in business processes with DMN". In: *Business Process Management Workshops: BPM 2015, 13th International Workshops, Innsbruck, Austria, August 31–September 3, 2015, Revised Papers 13*. Springer. 2016, pp. 418–431.

[30] Zahra Dasht Bozorgi et al. "Process mining meets causal machine learning: Discovering causal rules from event logs". In: *2020 2nd International Conference on Process Mining (ICPM)*. IEEE. 2020, pp. 129–136.

[31] Mahnaz Sadat Qafari and Wil van der Aalst. "Root cause analysis in process mining using structural equation models". In: *Business Process Management Workshops: BPM 2020 International Workshops, Seville, Spain, September 13–18, 2020, Revised Selected Papers 18*. Springer. 2020, pp. 155–167.

[32] Wil Van Der Aalst. *Process mining: data science in action*. Vol. 2. Springer, 2016.

[33] Zahra Dasht Bozorgi et al. "Prescriptive process monitoring for cost-aware cycle time reduction". In: *2021 3rd international conference on process mining (ICPM)*. IEEE. 2021, pp. 96–103.

[34] Zahra Dasht Bozorgi et al. "Prescriptive process monitoring based on causal effect estimation". In: *Information Systems* (2023), p. 102198.

[35] Sven Weinzierl et al. "Prescriptive business process monitoring for recommending next best actions". In: *Business Process Management Forum: BPM Forum 2020, Seville, Spain, September 13–18, 2020, Proceedings 18*. Springer. 2020, pp. 193–209.

[36] Irene Teinemaa et al. "Alarm-based prescriptive process monitoring". In: *Business Process Management Forum: BPM Forum 2018, Sydney, NSW, Australia, September 9-14, 2018, Proceedings 16*. Springer. 2018, pp. 91–107.

[37] Ruocheng Guo et al. "A survey of learning causality with data: Problems and methods". In: *ACM Computing Surveys (CSUR)* 53.4 (2020), pp. 1–37.

[38] G. B. Mathews. “On the partition of numbers”. In: *Proceedings of the London Mathematical Society* s1-28.1 (Nov. 1896), pp. 486–490.

[39] Richard Bellman. *Dynamic Programming*. Princeton, NJ, USA: Princeton University Press, 1957.

[40] P. C. Gilmore and R. E. Gomory. “The theory and computation of knapsack functions”. In: *Operations Research* 14.6 (1966), pp. 1045–1074.

[41] G. L. Nemhauser and Z. Ullmann. “Discrete dynamic programming and capital allocation”. In: *Management Science* 15.9 (1969), pp. 494–505.

[42] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. Berlin and Heidelberg: Springer, 2004.

[43] Silvano Martello and Paolo Toth. *Knapsack Problems: Algorithms and Computer Implementations*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Chichester and New York: Wiley, 1990.

[44] Christophe Wilbaut et al. “The Knapsack Problem and Its Variants: Formulations and Solution Methods”. In: *The Palgrave Handbook of Operations Research*. Ed. by Saïd Salhi and John Boylan. Cham: Springer International Publishing, 2022, pp. 105–151.

[45] Olivier Goldschmidt, David Nehme, and Gang Yu. “Note: On the set-union knapsack problem”. In: *Naval Research Logistics (NRL)* 41.6 (1994), pp. 833–842.

[46] Adriano Augusto et al. “Automated Discovery of Process Models from Event Logs: Review and Benchmark”. In: *IEEE Trans. Knowl. Data Eng.* 31.4 (2019), pp. 686–705. DOI: 10.1109/TKDE.2018.2841877.

[47] Gerhardus A. W. M. van Hulzen et al. “Mining context-aware resource profiles in the presence of multitasking”. In: *Artif. Intell. Medicine* 134 (2022), p. 102434. DOI: 10.1016/J.ARTMED.2022.102434.

[48] Ilya Verenich et al. “White-box prediction of process performance indicators via flow analysis”. In: *Proceedings of the 2017 International Conference on Software and System Process*. 2017, pp. 85–94.

[49] Humam Kourani and Sebastiaan J van Zelst. “POWL: partially ordered workflow language”. In: *International Conference on Business Process Management*. Springer. 2023, pp. 92–108.

[50] Alessandro Berti, Sebastiaan J. van Zelst, and Daniel Schuster. “PM4Py: A process mining library for Python”. In: *Softw. Impacts* 17 (2023), p. 100556. DOI: 10.1016/J.SIMPA.2023.100556.

[51] Boudewijn van Dongen. *BPI Challenge 2012*. 2012. DOI: 10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f.

[52] Massimiliano De Leoni, Wil MP van der Aalst, and Marcus Dees. “A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs”. In: *Information Systems* 56 (2016), pp. 235–257.

[53] T. S. Verma. *Graphical aspects of causal models*. Technical Report R-191. Computer Science Department, University of California, Los Angeles, 1993.

[54] Ashwin Arulselvan. "A note on the set union knapsack problem". In: *Discrete Applied Mathematics* 169 (2014), pp. 214–218.

[55] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[56] Aric Hagberg, Pieter Swart, and Daniel S Chult. *Exploring network structure, dynamics, and function using NetworkX*. Tech. rep. Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[57] Amit Sharma and Emre Kiciman. "DoWhy: An End-to-End Library for Causal Inference". In: *arXiv preprint arXiv:2011.04216* (2020).

[58] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. "Causal inference on discrete data using additive noise models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.12 (2011), pp. 2436–2450.

[59] Dhanya Pramod Anil Jadhav and Krishnan Ramanathan. "Comparison of Performance of Data Imputation Methods for Numeric Dataset". In: *Applied Artificial Intelligence* 33.10 (2019), pp. 913–933.

[60] Mark Dukes. "Stakeholder Utility Measures for Declarative Processes and Their Use in Process Comparisons". In: *IEEE Transactions on Computational Social Systems* 9.2 (2022), pp. 546–558. DOI: 10.1109/TCSS.2021.3092285.

[61] Sebastián Maldonado, Juan Pérez, and Cristián Bravo. "Cost-based feature selection for Support Vector Machines: An application in credit scoring". In: *European Journal of Operational Research* 261.2 (2017), pp. 656–665. ISSN: 0377-2217.

Appendix

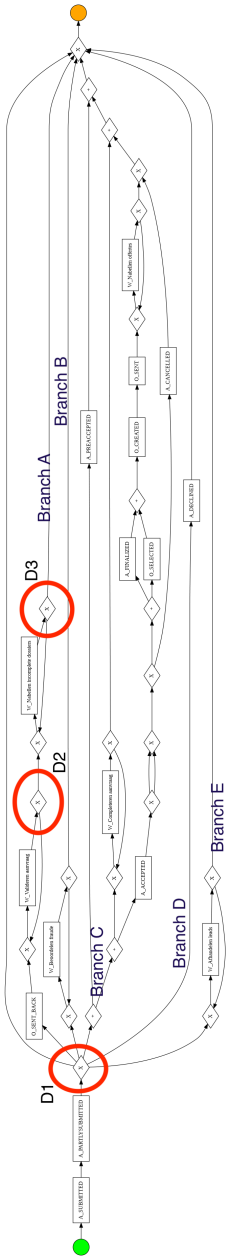


Figure 6.1: BPMN model obtained by running inductive miner on BPIC 2012 dataset.