Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 00 (2025) 000–000

www.elsevier.com/locate/procedia

19th CIRP Conference on Intelligent Computation in Manufacturing Engineering

CausalPlan: Integrating Causal Models to Estimate Processing Times for Production Plans

Marvin Matthes[a*], Martin Krockert[a], Torsten Munkelt[a]

[a]*Hochschule für Technik und Wirtschaft Dresden – University of Applied Sciences, Friedrich-List-Platz 1, 01069 Dresden, Germany*

* Corresponding author. Tel.: +49 351 462 3796. *E-mail address:* marvin.matthes@htw-dresden.de

Abstract

Production planning relies on realistic processing times, which depend on various influencing variables. To improve planning precision, we discover causal models from production data and apply these to generate precise processing times during production planning. For validation, we plan a production using fix processing times and then simulate the plan's execution, applying a given causal model to infer deviating processing times. From the simulation results, we rediscover the causal model and integrate it in production planning. Hence, we demonstrate how causal models improve planning accuracy and provide insight, how production variables relate to each other.

Peer-review under responsibility of the scientific committee of the 19th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 16-18 July, Gulf of Naples, Italy

Keywords: Causal Models; Production Planning; Prediction; Causal Bayesian Network

Nomenclature

O	is an Operation
$\mathcal{M}$	is an Machine
$\mathcal{J}$	is an Job, where each contains sequence of, Operations $(O_1, O_2, \cdots, O_{jm})$
$\mathcal{A}$	denote the set of jobs (production orders)

1. Introduction

Production scheduling is fundamentally dependent on realistic assumptions concerning the associated production plans. Particularly within job shop production environments, the definition of production plans – encompassing both the sequence of operations and the estimated processing times for each operation – often relies heavily on expert knowledge.

In today's high-tech manufacturing landscape, the costs associated with machinery make high machine utilization and throughput essential factors for maintaining market competitiveness. Production schedules are so sensitive that even minimal deviations in the processing time of a single operation can trigger a series of rescheduling efforts, impacting subsequent operations on the same machine and disrupting the entire production schedule. Since the plan made in advance often becomes the one that actually runs on the shop floor, it's essential that schedules not only optimize key objectives but also remain close to real execution. Accurate processing time estimates are therefore critical to ensure that the plan is both feasible and robust under real-world conditions.

Causal models offer a promising approach to address this challenge. As established in the literature [4, 6], causal models serve multiple purposes. Firstly, they provide a formal description of the joint observational distribution among variables. Secondly, and crucially for planning, they allow predictions under interventions and counterfactual conditions.

2212-8271
Peer-review under responsibility of the scientific committee of the 19th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 16-18 July, Gulf of Naples, Italy

The capability of causal models to generate context-dependent, probabilistic predictions, especially when considering potential interventions or changes in production parameters, can significantly aid in formulating more robust and precise production plans.

By integrating causal models into the planning process, we expect to achieve several key improvements:

- Develop more capable production plans that more accurately reflect operational realities.
- Reduce overall manufacturing times and lead times.
- Enhance the reliability of meeting production throughput times.

2. Related work

The prediction of processing times in manufacturing, particularly in job shop environments, has been explored using various machine learning and statistical techniques. For instance, Silva et al. [5] applied artificial neural networks for job shop prediction tasks. Addressing the challenge of unknown and complex processing times in real-world settings, Yamashiro [8] employed machine-learning models, specifically LightGBM and Ridge regression, to estimate these durations. Furthermore, Silva [5] introduced a framework integrating process mining with associational Bayesian Networks and predictive models. This approach aims to accurately estimate task completion times and identify performance-critical activities within manufacturing processes. In a comparative study, Sousa et al. [6] evaluated different AutoML models, including AutoGluon, H2O AutoML, rminer, and TPOT. Their findings suggest that incorporating data about individual manufacturing operations significantly improves the accuracy of predicting the total processing time for an entire production order. While these approaches demonstrate the utility of predictive modeling, our work focuses specifically on leveraging causal models. We claim that causal models capture the underlying cause-and-effect relations between variables, better than any predictive or associational model., and therefore enabling more robust predictions under intervention and providing deeper insights into the system dynamics influencing processing times.

3. Foundations of Probabilistic and Causal Graphical Models

Probabilistic Graphical Models (PGMs) offer a framework to represent and reason about uncertainty using the principles of probability theory and graph theory. By encoding only relevant conditional independencies through a structured graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$, PGMs enable the computation of joint, marginal, and conditional probabilities. This formulation supports inference and learning, making PGMs particularly suitable for high-dimensional problems where many variables interact. Probabilistic graphical models can be classified along several key dimensions: (i) directed vs. undirected models, depending on whether the edges represent directional (possibly causal) influence or symmetric dependencies; (ii) static vs. dynamic, depending on whether the model represents a snapshot at a single time point or evolves over time; and (iii) probabilistic vs. decisional, depending on whether the model includes only random variables or also incorporates decisions and utilities for reasoning under uncertainty. Each class offers distinct modeling capabilities depending on the domain and application [7].

However, standard PGMs do not distinguish between statistical association and causal influence. Two structurally distinct Bayesian networks, such as

$$A \rightarrow B \rightarrow C \text{ and } A \leftarrow B \leftarrow C \quad (1)$$

may encode identical conditional independencies but with different causal meanings if edges are interpreted as mechanisms. To address such limitations, causal graphical models extend PGMs by explicitly encoding cause-effect relationships and enabling new forms of reasoning.

Causal models, in particular Causal Bayesian Networks (CBNs), are well-suited to supporting both interventional and counterfactual queries. These capabilities are essential in domains where it is crucial to understand and manipulate system mechanisms. From a manipulationist perspective, causality is defined as a relationship in which manipulating a cause variable alters the probability distribution of its effect variable(s). Formally, causality is modeled as a binary relation $\rightarrow \subseteq \mathcal{F} \times \mathcal{F}$ on a finite probability space $(\Omega, \mathcal{F}, \mathsf{P})$, where $\mathsf{A} \rightarrow \mathsf{B}$ denotes that A causes B. This relation satisfies the following properties [7]:
transitivity:

$$\mathsf{A} \rightarrow \mathsf{B} \text{ and } \mathsf{B} \rightarrow \mathsf{C}, \text{ then } \mathsf{A} \rightarrow \mathsf{C} \quad \text{(i)}$$

irreflexivity:

$$\mathsf{A} \nrightarrow A \; for \; all \; \mathsf{A} \in \mathcal{F} \quad \text{(ii)}$$

antisymmetry:

$$\text{if } \mathsf{A} \rightarrow \mathsf{B}, \text{ then } \mathsf{B} \nrightarrow \mathsf{A} \text{ unless } \mathsf{A} = \mathsf{B} \quad \text{(iii)}$$

In this sense, a CBN is defined as a directed acyclic graph whose edges represent such causal relations, enabling the formal treatment of intervention (via *do*-calculus) and counterfactuals.

A key capability enabled by CBNs is reasoning about interventions. An intervention involves actively setting a variable X to a specific value x, denoted by the do(X=x) operator [1]. This is fundamentally different from observing X=x. The effect of an intervention is calculated using the do-calculus, a set of syntactic rules that allow us to transform expressions involving the do-operator

$$P\big(Y \mid do(X = x)\big) \quad (2)$$

the probability of Y given that we intervene to set $X = x$ into standard conditional probability expressions

$$P(Y \mid X = x, Z = z) \quad (3)$$

that can be estimated from observational data, provided certain graphical criteria are met. The do-calculus allows predicting the outcomes of actions or policy changes, even in the presence of unobserved confounding variables.

Furthermore, causal models facilitate counterfactual reasoning, which addresses "what if" questions about scenarios that did not actually occur (e.g., "What would the processing time have been if a different machine setting had been used, given the actual outcome?"). Counterfactuals require a fully specified causal model, often in the form of Structural Causal Models (SCMs), which define how each variable's value is determined by its direct causes and exogenous noise terms. They allow reasoning about alternative possibilities constrained by the observed facts [2, 3].

By combining the compactness and efficiency of PGMs with the interpretability and decision-making capabilities of causal reasoning, causal graphical models provide a powerful framework for both understanding complex systems and actively reasoning about change.

4. CausalPlan

As highlighted earlier, traditional production planning often relies on predetermined estimates for processing times of operations. However, these assumptions frequently diverge from the actual processing times observed during execution, leading to plan inaccuracies and inefficiencies. Our approach addresses this by learning a Causal Model from historical production feedback data and integrating its predictive capabilities directly into the planning process.

The core idea involves two main steps: first, learning the causal structure and parameters from data (Causal Discovery and Parameter Learning), and second, using the learned model to inform the planning process (Causal Inference for Planning).

Causal Discovery and Parameter Learning: The initial step is to identify the causal relationships between relevant production variables from observational data. This process, known as causal structure learning or causal discovery, aims to generate a CBN where the directed edges represent genuine causal influences, not just correlations. Numerous algorithms exist for this purpose, and libraries like pgmpy and gcastle provide implementations of various methods [1, 9]. While purely data-driven discovery is possible, practical applications often benefit from incorporating expert knowledge to guide the search, constrain possible relationships, or validate discovered edges. We anticipate an iterative process where initial structures derived from algorithms are refined with domain expertise to converge towards the ground truth graph.

Once a causal structure is established, the next step is parameter learning. This involves estimating the conditional probability distributions (CPDs) associated with each variable, given its direct causes (parents) in the graph. For instance, we estimate $P\left(p_O^{(m)} \mid state(m), D\right)$. This results in a fully parametrized CBN. We utilize frameworks like pgmpy [1] for parameter estimation from the available production data.

Causal Inference for Planning: With the parametrized CBN, we can replace static duration estimates with context-dependent predictions. Within our causal framework for production scheduling, we categorize variables into four main types:

1. Observable Variables: Factors directly measured or recorded (e.g., Material Type, Batch Size, Operator Skill Level, Cleaning performed, Tool Change occurred).
2. Unobservable Variables (Latent Variables): Underlying states influencing the system but not directly measured (e.g., internal Machine State encompassing wear, temperature drift).
3. Interventional Variables: Actions or settings actively controlled (e.g., deciding on a Tool Change, scheduling Cleaning, setting Machine Speed). Note the overlap: Tool Change can be observed and intervened upon.
4. Target Variable: The primary outcome to predict (e.g., processing time).

Consider our simplified example in Fig. 1: A planned Tool Change (Interventional/Observable) influences Machine State (Unobservable), which affects processing time as a deviation from the original processing time of the workplan (Target) and potentially the need for Cleaning (Observable/Interventional). The learned CBN captures these dependencies.

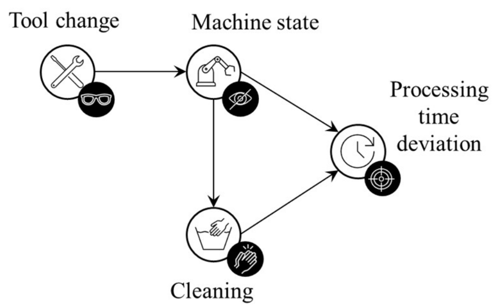

Fig. 1. Causal Model.

During the planning phase, instead of using a fixed processing time, we infer the expected processing time using the CBN. This inference considers the known or planned values of the operation's causal parents (e.g., planned tool change). The calculation leverages the learned CPDs within the CBN to provide a probabilistic estimate sensitive to the specific context of the operation being planned. The used algorithm is shown in Algorithm 1 employs a structural causal model to estimate the effects of different interventions on a target variable. By observing evidence variables and querying the model under various intervention scenarios, it computes expected outcomes and selects the intervention that optimizes the desired objective.

To cover dynamic processing times, based on influencing variables we integrate Causal Predictions and Interventions into the scheduling process. Therefor we employ an extended Giffler-Thompson algorithm, shown in Algorithm 2. This algorithm, suitable for job shop scheduling, allows us to incorporate the operation durations inferred from the causal model as we construct the production schedule step-by-step. This results in plans that are inherently more adaptive and potentially more accurate than those based on static time estimates.

Algorithm 1: Causal Interventional Model

$Input: evidence\ variable\ E, intervention\ variable\ D,$
$\quad target\ variable\ T, causal\ model\ C$
$Output: resultR*, inferencedvariables\ E, D, T$
$e \leftarrow observe(E)$
$q \leftarrow M.query(T \mid e, do(D \in \{0,1\}))$
$m \leftarrow mapping\ of\ T\ to\ values$
$v \leftarrow q \times m$
$d* \leftarrow argmin(v)$
$q* \leftarrow q[d*]$
$i* \leftarrow sample(q*)$
$R* \leftarrow adjust(m[i*])$
return $(R*, \{E: e, D: d*, T: m[i*]\})$

Algorithm 2: Extended Giffler Thomson for FJSSP

$\mathcal{A} \leftarrow \{O_{j1} \mid J_j \in \mathcal{J}\}$
$S \leftarrow \emptyset$
$avail(m) \leftarrow 0\ for\ all\ m \in \mathcal{M}$
while $\mathcal{A} \neq \emptyset$ **do**
  foreach $O \in \mathcal{A}, m \in \mathcal{M}_O$ **do**
    $\text{state}_m \leftarrow$ current state of machine m
    $p_O^{(m)} \leftarrow f_{\text{dur}}(O, m, \text{state}_m)$
    $ES_O^{(m)} \leftarrow \max(\text{ready}(O), avail(m))$
  $(O^*, m^*) \leftarrow \arg\min_{O \in \mathcal{A}, m \in \mathcal{M}_O} \left(ES_O^{(m)} + p_O^{(m)}\right)$
  $\mathcal{C} \leftarrow \{O \in \mathcal{A} \mid m^* \in \mathcal{M}_O \wedge ES_O^{(m^*)} < ES_{O^*}^{(m^*)} + p_{O^*}^{(m^*)}\}$
  $\tilde{O} \leftarrow \arg\min_{O \in \mathcal{C}} p_O^{(m^*)}$
  Schedule $\tilde{O}$ on m^* at time $ES_{\tilde{O}}^{(m^*)}$
  $avail(m^*) \leftarrow ES_{\tilde{O}}^{(m^*)} + p_{\tilde{O}}^{(m^*)}$
  Update $\text{state}(m^*) \leftarrow$ new machine state after $\tilde{O}$
  $S \leftarrow S \cup \{\tilde{O}\}$
  $\mathcal{A} \leftarrow \mathcal{A} \setminus \{\tilde{O}\}$
  foreach $successor\ O'\ of\ \tilde{O}$ **do**
    if all predecessors of O' are in S
      $\mathcal{A} \leftarrow \mathcal{A} \cup \{O'\}$
return schedule S

5. Evaluations

In order to evaluate our causal model approach, we designed an experiment — summarized in Fig. 2 — built around a discrete - event simulator that generates both ground - truth data and production plan schedules. First, we encode the *true* causal relationships among factors influencing processing times as an expert Bayesian network (Fig. 1) and embed this network in a SimPy - based Job Shop Scheduling Problem (JSSP) simulator.

In each simulation run, 100 jobs drawn equally from two product types; for each job, the number of sequential operations is sampled from a normal distribution $\mathcal{N}(\mu_o = 5, (0.2\,\mu_o)^2)$ and each operation's expected processing time is drawn from $\mathcal{N}(\mu_{dur} = 10, (0.2\,\mu_{dur})^2)$ to capture intrinsic variability. Operations follow sequential routings through three machines with two available tools and are dispatched via the Giffler–Thompson priority rule to produce a *ground-truth* schedule.

Once this simulated factory floor has produced both operation-level duration samples and the corresponding ground-truth schedule, we train five alternative predictive models on one batch of 100 jobs and then use those models to predict durations for a fresh batch of 100 jobs. The simplest baseline is a static *mean-duration estimator* that predicts each operation's duration by the historical sample mean stratified by product type and operation index. Next, we fit two *causal models*: (1) a purely observational causal-inference model that learns the conditional dependencies in the causal model to estimate duration distributions under observational conditions, and (2) an interventional causal model that additionally simulates the effects of hypothetical interventions by incorporate a do-calculus operation on upstream covariates (by enforcing a cleaning) when predicting durations. Finally, we consider two *distribution-based* approaches: a parametric log-normal model—fitting a log-normal distribution to each operation's durations—and a nonparametric empirical model that samples from histograms of historical durations per operation.

For each predictive model, we rerun the extended Giffler–Thompson scheduler using the model's predicted durations to produce a model-based production plan schedule.

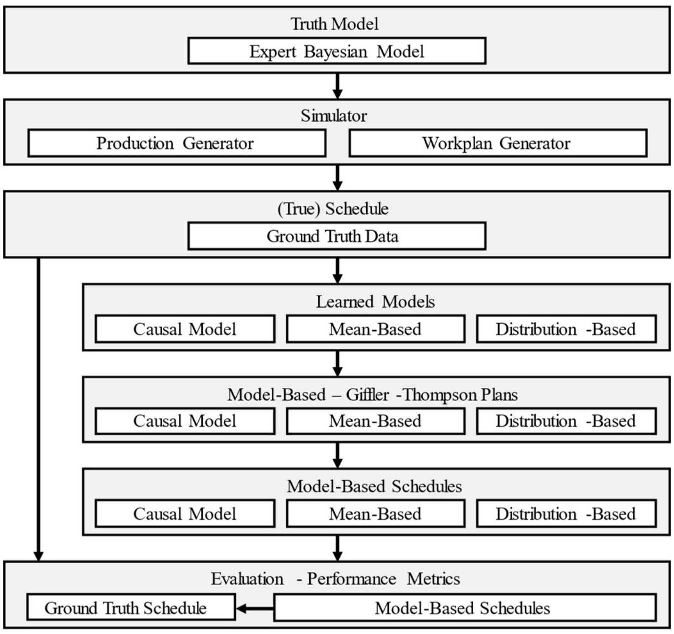

Fig. 2. Experimental design of CausalPlan

As illustrated in Table 2, the degree to which each predictive approach accurately recreates the ground truth schedule is averaged over 20 independent simulations. The evaluation of each model is conducted through the utilization of five distinct metrics: (6) the change in makespan, (7) the change in throughput, (8) the average start-time shift, and (9) the sequence similarity. The determination of the "best" model is based on the minimization of deviations from zero, that is, the extent to which the simulator's schedule is accurately recovered. The following formulae are employed in the calculation of our metrics:

Let $\mathcal{A}$ denote the set of jobs (production orders), where each job $\mathcal{J} \in \mathcal{A}$ consists of a subset of operations $0 \subseteq \mathcal{S}$. Each operation O has a scheduled start time $start(O)$ and an end time $end(O)$. The start and completion time of a job $\mathcal{J}$ are defined as:

$$start_{\mathcal{S}}(J) = \min_{O \in J}\big(start(O)\big) \quad (4)$$

$$start_{\mathcal{S}}(J) = \min_{O \in J}\big(end(O)\big) \quad (5)$$

Diff. of the makespan between two schedules $\mathcal{S}$ and $\mathcal{S}'$:

$$\Delta makespan(\mathcal{S}, \mathcal{S}') = \max_{O \in \mathcal{S}}\big(end(O)\big) - \max_{O' \in \mathcal{S}'}\big(end(O')\big) \quad (6)$$

The average throughput of job $\mathcal{J}$ in Schedule $\mathcal{S}$ is then defined as:

$$throughput(\mathcal{S}, \mathcal{J}) = \frac{|\mathcal{J}|}{\max_{O \in \mathcal{J}}(end_{\mathcal{S}}(O)) - \max_{O \in \mathcal{J}}(end_{\mathcal{S}}(O))} \quad (7)$$

Mean operation start shift based on the operations in $\mathcal{S}$

$$shift(\mathcal{S}, \mathcal{S}') = \frac{1}{|O|}\sum_{O \in \mathcal{S}}\big(start_{\mathcal{S}}(O) - start_{\mathcal{S}'}(O)\big) \quad (8)$$

The operation position sequence similarity is denoted on an injective encoding of a machine sequence, defined as $\phi: A \cup B \rightarrow \Sigma$, *with:*

$$enc(A) = \big(\phi(a_1), \ldots, \phi(a_{|A|})\big),$$

$$enc(B) = \big(\phi(b_1), \ldots, \phi(b_{|B|})\big).$$

$$seqsim(A, B) = Lev\ (enc(A), enc(B)) \quad (9)$$

where $Lev(\cdot\,,\,\cdot)$ denotes the Levenshtein distance between two strings.

Fig. 3 plots the distribution of makespan deviations (Δ Makespan) for each predictive approach relative to the ground - truth simulator. The purely observational Causal-Inference model attains the small bias, with a mean Δ of - 0.55 time units. Both distribution-based approaches, the Nonparametric and Log-Normal estimators, achieve comparable accuracy. As expected, the Interventional Causal (Do) model systematically implies interventions to create shorter schedules, achieving the shortest makespan (mean Δ = –8.35). Finally, the baselines approaches “Mean-Duration” and “Static-Duration” show large positive biases (+41.9 and +60.8, respectively), underscoring their inability to capture either the inherent stochasticity of operation times or the underlying causal dependencies.

As demonstrated in Figure 4, the observational Causal-Inference model achieves the closest replicates the simulator's throughput, exhibiting a mean deviation of approximately -2.4 time units. The interventional Causal (Do) model, developed with the specific objective of minimizing makespan, reliably yields the lowest throughput, as intended.

Table 2. Experimental results from calculating the average for each Δ comparing the production plan schedule S for each approach with the simulated production schedule S’

Approach	Make-span	Through-put	shift	μ_{dur}	seqsim
Mean Duration	41.90	16.9	101.8	0.76	41.05
Causal (Do)	-8.35	26.63	94.02	0.92	40.72
Causal (Inference)	-0.55	-2.42	94.40	0.94	41.00
Nonparametric-Distribution	*-0.50*	-5.95	93.36	0.91	40.96
Log-Normal-Distribution	0.55	-3.84	93.36	0.91	40.96
Static Duration	60.8	26.63	106.1	0.79	40.7

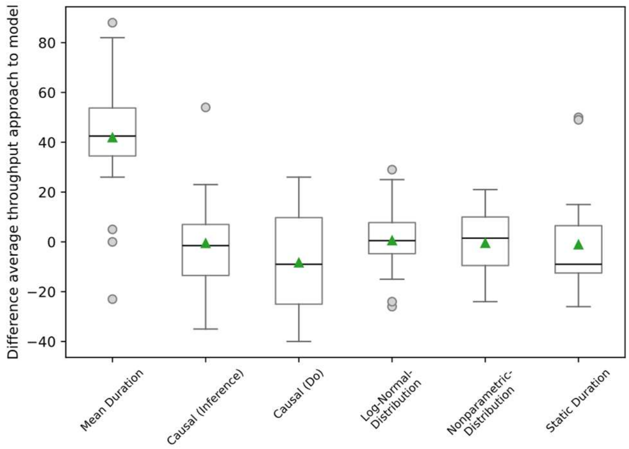

Figure 3: Distance of makespan between the approach and the actual production execution based on the ground truth

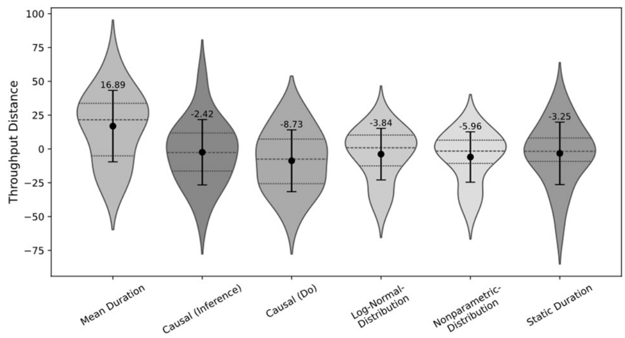

Figure 4: Throughput distance

In terms of schedule stability, all of the learned models—whether causal-inference, interventional causal, or distribution-based—maintain remarkably consistent operation start times, with average shifts clustering tightly around 93–94 time units (Table 2). By comparison, the static-duration baseline shifts operations by roughly 106 units and the mean-duration estimator by about 102 units, revealing that naïve strategies introduce considerably more schedule drift. Machine sequence similarity, measured via the Levenshtein distance, further confirms this pattern: all approaches achieve distances in the 40–41 range, with the causal models marginally edging out the others. Even as the causal-inference model delivers near-perfect makespan and throughput predictions, it does so

without compromise in stability.
From a computational standpoint, the observational causal-inference approach also offers a favorable trade-off between accuracy and runtime. Once the Bayesian network structure and conditional distributions are learned, sampling processing times from known conditionals even work in high-throughput scheduling contexts. The interventional (Do) model must repeatedly sample under each hypothetical intervention, which can become costly in real-time or large-scale settings. Moreover, if the underlying Causal Bayesian Network is not expert-provided, learning its structure from data can entail substantial computational cost and data requirements.

4. Conclusion and Outlook

Our experiments demonstrate that embedding causal models into production-planning models yields both accuracy and decision transparency. In particular, the purely observational causal-inference approach reproduces the simulator's own makespan and throughput almost perfectly, outperforming naïve mean-duration baselines and purely distributional estimators. By explicitly modeling the conditional dependencies encoded in a provided Bayesian network, this method not only tightens its prediction intervals but also avoids the systematic over- and under-estimation that plagues alternative approaches.
Beyond accuracy, the causal framework delivers powerful explainability, as it can visualize how observational and interventional factors influence each operation to affect processing times. Because the structure of the Bayesian network is both graphical and modular, the same approach can be readily adapted to a wide variety of production environments. Whether applied to batch processes, assembly lines, or hybrid flow shops, one need only adjust the graph topology and conditional distributions to reflect domain-specific causal relationships.
However, several limitations must be acknowledged. First, learning a high-dimensional causal graph from data remains an open challenge: existing structure-learning algorithms often demand extensive data or heavy expert involvement. Second, the interventional (Do) variant—while valuable for "what if" analyses—incurs extra runtime to resample covariates under each hypothetical scenario, making it less suitable for latency-sensitive, real-time planning. Finally, like any stochastic model, even the best causal-inference approach guarantees only statistical accuracy; individual production runs may still deviate notably from expectations in highly variable or rapidly changing environments.
Future work will focus on (i) automated structure learning to reduce reliance on expert‑provided graphs, (ii) approximate inference techniques (e.g., variational Bayes) to speed up interventional sampling, and (iii) robust scheduling under distributional uncertainty — leveraging the BN to generate probabilistic schedules with explicit confidence bounds. Such extensions promise to make causal‑modeling approaches not only accurate and explainable but also practical for real‑world, high‑throughput manufacturing systems.

Acknowledgements

This research was funded by the Federal Ministry for Economic Affairs and Climate Protection on the basis of a resolution of the Deutscher Bundestag with funding code 13IK020F. Funded by the European Union.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used DeepL Write and GPT chat bots in order to rephrasing and finding synonyms. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

[1] Ankan A and Textor J. "pgmpy: A Python Toolkit for Bayesian Networks". In: Journal of Machine Learning Research 25.265 (2024), pp. 1–8. url: http://jmlr.org/papers/v25/23-0487.html.
[2] Pearl J. "Causal inference in statistics: An overview". In: Statistics Surveys 3 (none Jan. 1, 2009). issn: 1935-7516. doi: 10 . 1214 / 09 - SS057. url: https://projecteuclid.org/journals/statistics-surveys/volume-3/issue-none/Causal-inference-in-statistics-An-overview/10.1214/09-SS057.full (visited on 04/02/2025).
[3] Peters J, Janzing D, and Schölkopf B. Elements of causal inference: foundations and learning algorithms. Adaptive computation and machine learning series. Cambridge, Massachuestts: The MIT Press, 2017. 265 pp. isbn: 978-0-262-03731-0.
[4] Ruschel E, De Freitas Rocha Loures E, and Santos. "Performance analysis and time prediction in manufacturing systems". In: Computers & Industrial Engineering 151 (Jan. 2021), p. 106972. issn: 03608352. doi:10.1016/j.cie.2020.106972
[5] Silva et al. "Job Shop Flow Time Prediction using Neural Networks". In: Procedia Manufacturing 11 (2017), pp. 1767–1773. issn: 23519789. doi: 10.1016/j.promfg.2017.07.309. url: https://linkinghub .elsevier.com/retrieve/pii/S2351978917305176.
[6] Sousa et al. "Production Time Prediction for Contract Manufacturing Industries Using Automated Machine Learning". In: Artificial Intelligence Applications and Innovations. Ed. by Ilias Maglogiannis et al. Vol. 647. Series Title: IFIP Advances in Information and Communication Technology. Cham: Springer International Publishing, 2022, pp. 262–273. isbn: 978-3-031-08336-5978-3-031-08337-2. doi: 10.1007/978-3-031-08337-2_22. url: https://link.springer.com/10.1007 /978-3-031-08337-2_22.
[7] Sucar E. Probabilistic Graphical Models: Principles and Applications. Advances in Computer Vision and Pattern Recognition. Cham: Springer International Publishing, 2021. isbn: 978-3-030-61942-8 978-3-030-61943-5. doi: 10.1007/978-3-030-61943-5. url: http:// link.springer.com/10.1007/978-3-030-61943-5 (visited on 03/04/2025).
[8] Yamashiro H and Nonaka H. "Estimation of processing time using machine learning and real factory data for optimization of parallel machine scheduling problem". In: Operations Research Perspectives 8 (2021), p. 100196. issn: 22147160. doi:10.1016/j.orp.2021.100196. url: https://linkinghub.elsevier.com /retrieve/pii/S2214716021000178.
[9] Zhang et al. gCastle: A Python Toolbox for Causal Discovery. Version Number:1.2021. doi:10.48550/ARXIV.2111.15155. url: https://arxiv.org /abs/2111.15155.