Proactive Data-driven Scheduling of Business Processes

Francesca Meneghello[1,2] , **Arik Senderovich**[3] , **Massimiliano Ronzani**[1] , **Chiara Di Francescomarino**[4] , **Chiara Ghidini**[5]
[1]Fondazione Bruno Kessler, Via Sommarive, 18, POVO - 38123, Trento, Italy
[2]Sapienza University of Rome, Via Ariosto, 25, 00185, Rome, Italy
[3]York University, 4700 Keele St, Toronto, ON M3J 1P3, Canada
[4]University of Trento, Via Sommarive, 9, 38123 Trento, Italy
[5]Free University of Bozen-Bolzano, via Bruno Buozzi, 1 - 39100, Bozen-Bolzano, Italy
{fmeneghello, mronzani}@fbk.eu, sariks@yorku.ca, c.difrancescomarino@unitn.it, chiara.ghidini@unibz.it

Abstract

Proactive scheduling creates robust offline schedules that optimize resource utilization and minimize job flow times. This work addresses scheduling challenges in business processes, often encountered in service systems, which differ from traditional applications like manufacturing due to inherent uncertainties in activity durations, and human resource availability. We model the business process scheduling problem (BPSP) as a variation of stochastic resource-constrained multi-project scheduling (RCMPSP), and apply *process mining* to infer unknown parameter values from historical event data. To overcome the randomness in activity durations, we transform the problem into its deterministic counterpart, and prove that the latter provides a lower bound on the Makespan of the stochastic problem. Our approach integrates data-driven Monte Carlo simulation with constraint programming to generate proactive schedules. We evaluate our approach using synthetic datasets with varying levels of uncertainty and size. In addition, we apply the approach to a real-world dataset from an outpatient cancer hospital, demonstrating its effectiveness in optimizing the process Makespan by an average of 5% to 14%.

1 Introduction

We address the challenge of scheduling business processes in service domains such as finance (e.g., purchasing, order fulfillment), healthcare (e.g., patient flow management, appointment scheduling), and retail (e.g., order processing, supply chain coordination). Business processes are structured sets of activities that organizations execute to achieve various objectives, such as completing a sale, providing a service, or managing a supply chain [Dumas *et al.*, 2018]. Unlike processes in manufacturing facilities, which typically involve predictable durations, and stable resource availability, business processes exhibit significant uncertainty and variability [Shoush and Dumas, 2022; Xu *et al.*, 2016]. Specifically, scheduling business processes is notoriously complex due to the stochastic nature of activity durations caused by human behavior, as well as the time-varying availability of resources influenced by factors such as part-time schedules, overlapping responsibilities, and vacations. These challenges make traditional deterministic scheduling methods inadequate [Pinedo, 2012].

In response to this limitation, proactive scheduling techniques generate robust offline schedules [Beck and Wilson, 2007; Chaari *et al.*, 2014]. These methods explicitly account for the stochastic nature of activity durations and aim to compute a *proactive optimal solution* that achieves a predefined confidence level [Beck and Wilson, 2007]. For example, a solution with a confidence level of $1 - \alpha$ ensures that the returned Makespan remains below an optimal threshold in at least $(1 - \alpha)$ of cases. While proactive scheduling has made significant progress in addressing variability through probabilistic activity durations [Satic *et al.*, 2022; Liu and Xu, 2020; Hauder *et al.*, 2020; Chen *et al.*, 2019; Beck and Wilson, 2007], critical gaps remain. Notably, the challenge of *planned* resource unavailability is seldom addressed. Although studies incorporate the planned unavailability of resources, some overlook the uncertainty of duration [Kreter *et al.*, 2018], while others fail to handle the complexity and variability typical of business processes [Yang *et al.*, 2020; Winklehner and Hauder, 2022].

Beyond methodological gaps, many real-world business processes face an additional practical challenge: the lack of readily available data on key scheduling parameters. Information such as activity durations, their probabilistic distributions, and resource availability calendars is often incomplete or missing. Without access to this data, even advanced scheduling techniques struggle to perform effectively in real-world settings. To bridge this gap, we employ *process mining*, a data-driven approach designed to extract knowledge and insights from event logs [van der Aalst, 2016]. Event logs are datasets that document process executions, capturing details such as activity types, start and end times, and the resources assigned to each activity.

Building on this foundation, we formulate the Business Process Scheduling Problem (BPSP) and establish its relationship to the well-known Resource-Constrained Project Scheduling Problems (RCPSP). Our approach leverages both process mining and proactive scheduling methods to tackle both the inherent uncertainty in business processes, and lack of knowledge of the problem. Specifically, we develop a framework that combines data-driven simulation [Meneghello *et al.*, 2025], and constraint programming [Kreter *et al.*, 2018]: we construct a constraint programming model that generates deterministic schedules for the BPSP, verify them using the simulator, and select the best performing schedules. These schedules are evaluated and optimized to minimize the uncertain Makespan while achieving a pre-specified confidence level, ensuring robustness against variability and resource constraints.

We evaluate our approach in two phases to demonstrate both its effectiveness and applicability. In the first phase, we conduct experiments using synthetic data, allowing us to systematically test the framework's ability to handle varying levels of uncertainty while accounting for resource calendars. This controlled environment ensures that the method can adapt to diverse scheduling scenarios and reliably optimize outcomes under uncertainty. In the second phase, we validate the applicability of our approach by applying it to real-world event logs from a healthcare process, achieving an optimization of the process Makespan by an average of 5% to 14%. Using process mining techniques, we extract BPSP parameters from the logs and employ our framework to generate optimal schedules. This highlights the practical value of our method, showcasing its ability to address complex, data-driven scheduling challenges in a real-world domain.

2 Business Process Scheduling

In this part, we first motivate our work with a real-life hospital example. Then, we proceed to define the business process scheduling problem (BPSP), and lastly, discuss the use of process mining for inference of BPSP parameters from data.

2.1 Motivating Example

We are motivated by a hospital process that provides a sequence of treatments to cancer patients. The resources involved in the process, nurses (N), physicians (P), and infusion nurses (IN), are shared among patients and must be scheduled ahead of time. The duration of treatments varies significantly based on patient-specific context (e.g., patient complexity). Furthermore, the resources involved are not always available, as they follow specific shift schedules and may go on planned vacations. For instance, nurses work 8 hours a day, 5 days a week, either in the morning or in the afternoon. Physician calendars are highly variable, and depend on exogenous factors (e.g., physicians serving in multiple hospitals).

The hospital process is depicted in Figure 1a, which illustrates possible patient pathways. The activities are represented by white rectangles, and the time required to complete each activity is defined by probability distribution functions. The process begins with the Blood Draw activity, performed by a nurse (N), and continues with the parallel execution of the Vitals (for vital signs) and Examination activities, performed by a nurse and a physician (P), respectively. Finally, based on previous activities, the physician decides whether to proceed with Chemo. Infusion.

To complement the picture, Figure 1b illustrates resource shifts for a typical day. Nurses have various shifts, which overlap for only two hours, between 11 am and 2 pm, while the physician has a two-hour break during which no examination activity can be performed. The hospital processes approximately 1000 patients per day, involving over 5000 activities. The goal is to find a proactive schedule that probabilistically minimizes the global Makespan, e.g., the time that the last patient leaves cannot exceed 6PM with probability of at least 0.95.

2.2 Problem Definition

Referring to the definition in [Sánchez *et al.*, 2023], a case in a business process corresponds to a project, and the activities within the case map to the jobs that comprise the project. Business process scheduling problems (BPSPs) can therefore be represented using the following parameters[1]:

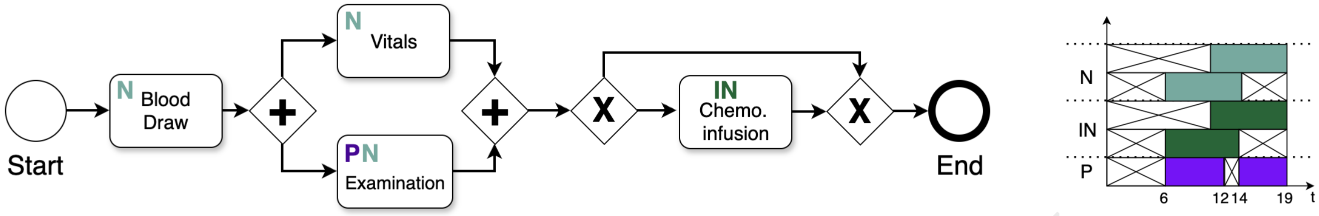
- $\mathcal{I}$ is the set of cases enumerated by $i \in \{1, 2, ...|\mathcal{I}|\}$,
- $A_i = \{a_1, \ldots, a_{|A_i|}\}$ is the set of activities comprising case i, where a_j, a_k may correspond to the same activity type (e.g., examination), and may repeat within a case,
- $\mathcal{A}$ is the set of all possible activity types, and ϕ is a mapping from activities to their types, $\phi(a_j) \in \mathcal{A}$,
- $\Pi_{i,a_j} \subseteq A_i$ represents activities that precede activity $a_j \in A_i$, i.e., their completion precedes the start of a_j,
- $\mathcal{R}$ is the set of resources involved in the process, with $r \in \{1, 2, ...|\mathcal{R}|\}$, and C_r being the capacity of r,
- $R_{i,a_j} \subseteq \mathcal{R}$ are the resources required to perform activity a_j in case i, and $\rho(a_j) = R_{i,a_j}$ is a function that returns the set of resources required by an activity,
- $\mathcal{T} = \{0, 1, \ldots, T\}$ is a set of time periods (e.g., minutes) during a scheduling horizon of length T,
- $D_{i,a_j} \sim \mathcal{P}_{i,\phi(a_j),\rho(a_j)}$ is the number of time periods required to perform activity a_j in case i, which follows the probability distribution $\mathcal{P}$ that depends on case i, activity type $\phi(a_j)$ and set of resources $\rho(a_j)$.[2]
- $v_{r,t} \in \{0, 1\}$ is the resource availability calendar, which is 1 if resource r is available at time period $t \in \mathcal{T}$.

It is worth mentioning that the BPSP is a variation of the RCPSP that, to the best of our knowledge, has not yet been addressed in the scheduling literature [Sánchez *et al.*, 2023].

Scheduling the business process is to assign each activity $a_j \in A_i,\ i = 1, \ldots, |\mathcal{I}|,\ j = 1, \ldots, |A_i|$ with a start time. Resource assignment is defined based on the activity requirement R_{i,a_j}, where non-interchangeable resources are treated as distinct. Optimal scheduling in our setting is to find a schedule such that the latest completion time among all cases, i.e., the global Makespan M, is minimized.

[1] RCMPSP notation taken from [Sánchez *et al.*, 2023].

[2] The only stochastic components of BPSP are activity durations.



(a) BPMN

(b) Resource calendars

Figure 1: Running example of hospital process

CaseId	Activity type	Start	Complete	Resources
1	Blood Draw	08:00	08:06	**N**
2	Blood Draw	08:01	08:07	**N**
3	Blood Draw	08:01	08:05	**N**
3	Vitals	08:05	08:11	**N**
2	Examination	08:07	08:14	**P**
2	Vitals	08:10	08:14	**N**
3	Examination	08:10	08:25	**P,N**
1	Vitals	08:11	08:14	**N**
1	Examination	08:11	08:30	**P,N**
2	Chemo. Infusion	08:20	09:10	**IN**
1	Chemo. Infusion	08:30	09:40	**IN**

Table 1: Example of a simple event log.

2.3 Learning BPSP Parameters From Event Logs

If all parameters of the BPSP are at hand, one can immediately switch to solving the underlying problem. However, if this is not the case, one can use historical data to infer these parameters with the use of process mining methods.

Process Mining [van der Aalst, 2016] integrates data science with business process analysis to extract insights from event logs recorded by enterprise information systems, such as EPIC in healthcare [Johnson, 2016]. Event logs (see Table 1) provide a valuable source of data for learning the parameters required for scheduling problems. An event log consists of multiple cases, where each case is a sequence of events corresponding to the execution of a case, e.g., a single journey of a patient in a hospital. Each event in the sequence is a measurement of an activity and its characteristics. Specifically, events capture the type of activity, the resources involved and the start and end timestamps that define the duration of the activity.

Learning Activities and Precedence. Activity types, which represent categories of activities performed in the process, are determined from the labels of activities recorded in the event log. Mapping these observed activities to predefined types involves grouping activities based on domain knowledge or textual similarity. The durations of activities can be derived from the start and end timestamps recorded in the event log. By analyzing these timestamps, probability distributions can be fitted to the observed durations of each activity type, such as normal or log-normal distributions, depending on the observed data, c.f., [Camargo *et al.*, 2020]. Precedence constraints can be inferred by examining the sequential order of activities in traces, identifying dependencies between activities where the completion of one activity consistently precedes the start of another [Senderovich *et al.*, 2019].

Inferring Resources and Capacities. Resource capacities can be inferred by analyzing the frequency of resource usage over time, leveraging approaches such as *Sched-Miner* [Senderovich *et al.*, 2015]. The effective capacity is estimated by considering the maximum resource utilization observed in the event log. This estimate often serves as a tight lower bound, as there remains a small probability that some resources were available but never utilized.

Resource availability calendars can be constructed by analyzing the distribution of timestamps associated with resource usage, identifying periodic patterns such as daily or weekly schedules, which can be assumed constant over extended periods. Resource requirements for activities can be extracted from the resource identifiers associated with events in the log.

Threats to Validity. The application of process mining techniques relies on several assumptions. It is assumed that logs are complete, capturing all resources and activities involved in the process, and that activities are non-preemptive, meaning they run to completion once started. Resource capacities are assumed to remain fixed over time, with resources becoming available immediately after completing an activity, unless they are absent according to their calendar. Additionally, activity durations are assumed to follow a known distribution, and resource calendars are considered periodic and consistent over the given time horizon. All of these assumptions may be violated in practice.

A first step in addressing threats to validity in process mining is to use representative event logs, i.e., logs that cover a time period meaningful to the process life-cycle and minimize the variability caused by seasonal behaviors. Variability in resource capacities and behaviors can be mitigated by estimating the capacity of pools of resources performing shared activities. Hence, resource pools reflect average performances and are not influenced by the behavior of individual resources. Finally, with respect to the time perspective of the process, many state-of-the-art approaches replace predefined distributions with advanced statistical or machine learning models to more accurately capture process dynamics.

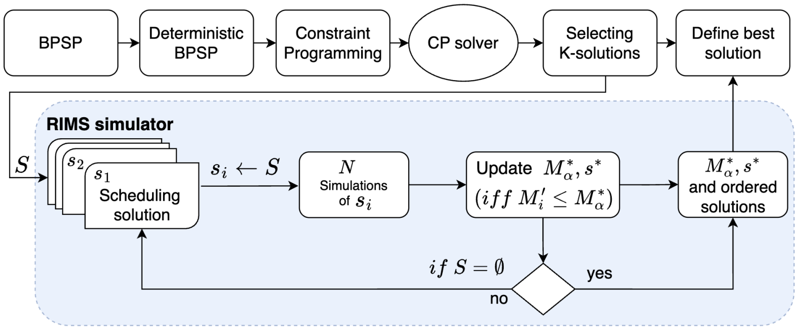


Figure 2: Our approach to Proactive Business Process Scheduling.

3 Proactive BPSP

Figure 2 illustrates the steps of our method. First, we define the BPSP based on either prior knowledge or by estimating its parameters from data. To manage the stochastic activity durations, we convert the stochastic BPSP into its deterministic counterpart by employing the transformation proposed in [Beck and Wilson, 2007], which is described in the next Section. Next, once the problem becomes deterministic, we construct a Constraint Programming (CP) model, and a CP solver is employed to obtain solutions that iteratively minimize the global Makespan M until either an optimal deterministic solution is found, or a time limit is reached. Once CP solutions are obtained, we first select the *top-*K solutions. Finally we employ Monte-Carlo simulation to identify the best solution. Below, we detail the different steps, and justify our design choices.

3.1 From Stochastic to Deterministic BPSP

The activity durations in a BPSP are random. Therefore, minimizing the Makespan implies finding the smallest possible value of M^*_α such that a solution exists with a random Makespan that, with high probability, $1-\alpha$ is less than M^*_α. We refer to M^*_α as the probabilistically minimum Makespan.

To transform the BPSP into a deterministic problem, we use the approach proposed by [Beck and Wilson, 2007], which associates a deterministic problem with the probabilistic one by transforming each random duration into the sum of its mean value and a buffer that depends on its standard deviation. More formally, for activity a_j in case i, the deterministic duration that is associated with it is,

$$d_{i,a_j} = \mu_{i,a_j} + q \cdot \sigma_{i,a_j}, \tag{1}$$

with μ_{i,a_j} being the mean duration of the activity, σ_{i,a_j} being its standard deviation and $q \geq 0$ being an uncertainty coefficient that provides a buffer for the mean value.[3] If $q = 0$, the mean value is used as activity duration across the board. All other components of the problem are deterministic, and remain unchanged. The challenge with the transformation defined in (1) is the identification of the value of a value of q such that the makespan M_q of the deterministic problem serves as a lower bound for the makespan M_α of the stochastic problem. In the remaining of the Section we provide some details on the solution proposed in [Beck and Wilson, 2007] and summarized the value of q defined in (2).

Lower-Bound Guarantees for BPSP

The work in [Beck and Wilson, 2007] shows that for Resource-Constrained Project Scheduling Problem (RCPSP), a lower bound for the probabilistic Makespan M^*_α is provided by the makespan M_U given by the solution of the deterministic problem obtained using the transformation (1) with the value of q set to

$$q_U = \frac{\Phi^{-1}(1-\alpha)}{\sqrt{U}}, \tag{2}$$

where Φ^{-1} is the inverse of the standard normal cumulative distribution function (CDF), and U represents an upper bound on the number of uncertain activities along the deterministic critical path (e.g., U can be set to the total number of activities). Furthermore, the authors in [Beck and Wilson, 2007] demonstrate that this lower bound can be tightened by selecting different values for q when additional assumptions are satisfied. The lower bound result is critical to efficiently select models using Monte Carlo simulation.

However, to apply the result in the BPSP case, we must overcome three violations of the classical RCPSP assumptions, namely multiple cases, potentially inter-dependent activity durations, and planned resource unavailability. The first two violations are straightforward to address. Since we consider the Makespan with respect to the latest activity regardless of the case, one can view BPSP as a single-case scheduling problem [Sánchez *et al.*, 2023]. Moreover, when conditioning on the case, the resources, and the activity being processed, activity durations become independent of each other. This property, referred to as conditional independence, ensures that the result from [Beck and Wilson, 2007] applies, as we assume the case identifier and activity information (i.e., its type and required resource set) are known.

The only significant limitation that remains unresolved relates to resource calendars. To prove that q_U indeed yields a lower-bound we must first define so-called *positive precedence expressions*. The set E is a set of precedence constraints between activities a_i and a_j specified as $\texttt{before}(i,j)$, i.e., the constraint that activity a_j cannot start

[3] Note that we assume the existence of the first two moments of the distribution for activity durations.

earlier than the completion of a_i. We are now ready to define positive precedence expressions (PPEs).

Definition 1 (Positive Precedence Expressions (PPE)). *The set E is referred to as a positive precedence expressions over a set of activities A if it is the smallest set satisfying:*

1. $\forall a_i, a_j, \mathtt{before}(i, j) \in E$.
2. *If $\phi, \psi \in E$, then $\phi \wedge \psi \in E$ and $\phi \vee \psi \in E$.*

In [Beck and Wilson, 2007] it is shown that for scheduling problems that can be written as a conjunction of PPEs, the proposed transformation that uses q_U (as above) provides a lower-bound on the stochastic problem. Therefore, what remains to be shown is that BPSP can be expressed by a conjunction of PPEs.

Proposition 1. *Selecting $q = q_U$ yields a lower-bound solution to the stochastic BPSP.*

Proof. We build the proof on the result of [Beck and Wilson, 2007], who established a lower bound for RCPSPs that do not account for resource unavailability. We will treat resource 'vacations' as dummy activities that we must schedule in addition to the other activities. The core argument relies on the ability to represent RCPSP constraints as Positive Precedence Expressions (PPEs). Specifically, the constraints of an RCPSP can be expressed as a conjunction of PPEs ([Beck and Wilson, 2007]):

1. **Precedence Constraints** (ϕ): Precedence relationships between activities a_i and a_j are represented as primitive precedence expressions $\mathtt{before}(i, j)$. Let ϕ denote the conjunction of all such precedence constraints.
2. **Resource Constraints** (ψ): Resource constraints can be represented using forbidden sets. A forbidden set $H \subseteq \mathcal{A}$ consists of activities whose simultaneous execution exceeds the capacity of a resource r, C_r. Resource constraints are satisfied if, for all $H \in F$, there exist two activities $a_i, a_j \in H$ such that $\mathtt{before}(i, j)$ holds. This is expressed as:

$$\psi = \bigwedge_{H \in F} \bigvee_{a_i, a_j \in H, i \neq j} \mathtt{before}(i, j).$$

Combining these, the constraints of an RCPSP are represented as $\phi \wedge \psi$, which is a conjunction of PPEs. In BPSP, resource unavailability introduces additional constraints: each resource r has an availability calendar $v_{r,t}$, where $v_{r,t} = 1$ if r is available at time t, and 0 otherwise. To incorporate resource unavailability, we redefine and extend the notion of forbidden sets:

1. Capacity forbidden sets H are defined as in RCPSPs, representing sets of activities whose simultaneous execution exceeds the capacity of a resource r.
2. For resource unavailability, we extend H with additional constraints for each resource r and time t where $v_{r,t} = 0$. The extended forbidden sets, denoted H', ensure that no activities requiring r overlap during unavailable periods.

The resource constraints with unavailability can then be expressed as:

$$\psi_u = \bigwedge_{H' \in F'} \bigvee_{a_i, a_j \in H', i \neq j} \mathtt{before}(i, j),$$

where F' is the extended set of forbidden sets that includes both capacity constraints and unavailability constraints.

The combined constraints for BPSP, including resource unavailability, can now be expressed as:

$$\phi \wedge \psi_u,$$

where ϕ represents precedence constraints, and ψ_u represents resource constraints, including unavailability. Each component is a conjunction of PPEs, ensuring that the overall problem retains the PPE structure.

Since BPSP constraints can be expressed as a conjunction of PPEs, the results from [Beck and Wilson, 2007] apply, and q_U provides a lower bound on the stochastic Makespan.

□

3.2 Solving Deterministic BPSP with CP

In this section, we present a CP model of the deterministic BPSP. We use the general CP model proposed by [Senderovich *et al.*, 2019], and extend it with the planned unavailability [Hauder *et al.*, 2020].

In particular, we employ the optional interval variable, var, to effectively represent the activities in our deterministic BPSP. The presence, start time, and duration are represented by Pres(var), Start(var) and Length(var), respectively, with Pres(var) $= 0$ indicating its absence in the constraint model. For each activity, $a_j \in A_i$, to be executed in a case i we define an interval variable x_{i,a_j} where Start(x_{i,a_j}) ≥ 0 and Pres(x_{i,a_j}) $= 1$. The precedence relations between the activities in i are guaranteed by the form EndBeforeStart (x_{i,a_k}, x_{i,a_j}) $\forall a_k \in \Pi_{i,a_j}$, ensuring that Start($x_{i,a_j}$) $\geq$ Start(x_{i,a_k}) $+$ Length(x_{i,a_k}). To represent the resources assignments we define,

$$X_{a_j} = \{\overline{x}_{i,a_j,R} : d_{i,a_j}, R \subseteq \mathcal{R}\},$$

as set of optional interval variables, where $\overline{x}_{i,a_j,R}$ represents the activity $a_j \in A_i$ assigned to a set of resources $R \subseteq \mathcal{R}$ with duration Length($\overline{x}_{i,a_j,R}$)$= d_{i,a_j}$ defined as (1).

Alternative(x_{i,a_j}, X_{a_j}) ensures that only one interval variable from X_{a_j} is present, and it starts and ends together with x_{i,a_j}. Unavailable($\overline{x}_{i,a_j,r}, v_{r,t}$) guarantees that the start and end times of the activity overlap with an available period in the resource calendar, $v_{r,t}$. Finally, to ensure that the total number of present and executing activity interval variables assigned to a resource does not exceed its capacity, we define Cumulative($\{\overline{x}_{i,a_j,r} : a_j \in A_i\}, C_r$), $\forall r \in \mathcal{R}$ and $\forall i \in \mathcal{I}$.

The goal is to minimize the global Makespan, i.e., the maximum end time across all activities.

$$\text{Makespan} = \max_{i \in \mathcal{I}, a_j \in A_i} \mathsf{Start}(x_{i,a_j}) + \mathsf{Length}(x_{i,a_j})$$

3.3 Solution Selection & Monte Carlo Simulation

In this phase, we apply a CP solver[4] to collect feasible solutions that iteratively minimize the global Makespan until either the optimal solution is found or a pre-defined time limit is reached. To evaluate the solutions returned by the CP solver and we employ Monte Carlo Simulation to select, M^*_α, the one that yields the probabilistic minimum Makespan. M^*_α is defined as the $M^*_\alpha = \min_{s_i \in S} M'_{s_i}$ where M'_{s_i} represents the $100(1-\alpha)$th percentile of Makespan values computed over N Monte Carlo simulations of a solution s_i returned by the CP solver. Each Makespan value, corresponds to the maximum end time across all activities in the simulation

Firstly, we select a subset of k-solutions from those returned by the solver, leveraging the well-established correlation between deterministic and probabilistic solutions [Bonfietti *et al.*, 2014].[5] Then, we iteratively evaluate selected solutions by performing a large number of independent simulations using the original stochastic activity durations to identify M^*_α.

4 Evaluation

In this section, we first outline the experimental settings, followed by a detailed description of the synthetic and real-world datasets used in our evaluation framework and their results. For synthetic experiments, we do not introduce other competitive methods, as the aim of these experiments is to test our framework under different uncertainty levels, problem sizes, and the presence or absence of resource calendars. For the real-world dataset, we do not compare our approach with other works due to their limitations, such as the absence of planned resource unavailability, the inability to account for uncertainty in activity durations, or scalability issues with large BPSP problems (see Section 5), which would lead to unfair comparisons.

4.1 Experimental Setting

Success Metrics. To assess the effectiveness of the proposed approach, we use two different metrics: the Normalized Probabilistic Makespan (NPM) and the Percentage of Improvement (PI). The NPM evaluates the ratio between the minimum probabilistic Makespan (M^*_α) and the minimum deterministic Makespan (M^*_C) obtained on synthetic experiments. It is computed as $NPM = M^*_\alpha / M^*_\text{C}$.

For the real-world dataset, we compute the PI metric to measure the improvement between the M^*_α obtained from the optimized schedule and M_act obtained from the actual schedule. It is calculated as $PI = ((M^*_\alpha - M_\text{act})/M_\text{act}) \times 100$.

Experimental Procedure. To transform the BPSP problems into deterministic ones, we need to define the q value for each of them. Instead of using q_U, we leverage the Monte Carlo simulation to obtain a more accurate estimation of the critical path compared to the one provided in the q_U definition (2). To do that we employ RIMS [Meneghello *et al.*, 2025], a state-of-the-art business process simulator capable to accurately represent the complexity of business processes. Specifically, we simulate the BPSP problem $1,000$ times using stochastic activity durations and identify the maximum critical path, π. In this way, we obtain a deterministic Makespan that is closer to the probabilistic one, as we will show in Figure 3. This higher correlation between deterministic and stochastic Makespan allows us to accelerate the convergence to the minimal probabilistic Makespan M^*_α.[6] The q_C is then defined as follows ([Beck and Wilson, 2007])

$$q_C = \frac{\Phi^{-1}(1-\alpha)}{\sqrt{|\pi|}} \frac{\sqrt{Mean\{\sigma^2_{i,j} : A_i \in \pi\}}}{Mean\{\sigma^2_{i,j} : A_i \in \pi\}},$$

where $|\pi|$ is the number of activities in π.

After collecting all the solutions returned by the solver, we select the top k-solutions to be evaluated with RIMS. These k-solutions are identified based on the last significant improvement[7], which tends to remain consistent in subsequent solutions. For instance, in Figure 3b, the last significant improvement is observed between solutions 42 and 43, marked by a notable step on the M_C line, followed by a plateau.

A 3-minute time limit is set for the CP solver, and the evaluation with RIMS, thanks to the k-solution selection, takes an average of 5 minutes, with a maximum of 15 minutes for the largest problem in the DayHospital experiment. The experiments are conducted on a PC with 16 GB of RAM and an M2 processor.

4.2 Synthetic Data Experiment

The Data. As synthetic data we use three problems from a set of publicly available JSP beanchmarks of different size small (10×10), medium (20×20), big (50×20) [Reijnen *et al.*, 2023][8], where with 10×10 we indicate a problem with 10 cases with 10 activities for each case. For each of the problems, we set three levels of uncertainty $u \in \{0.1, 0.5, 1\}$ to define the standard deviation of each activity $a_{i,j}$ as a random number within the interval $\sigma_{i,j} = [0, u * \mu_{i,j}]$. For each resource involved, we generate the corresponding calendars, assuming a working week of 5 days and 8 hours per day.

Results. Table 2 presents the normalized probabilistic Makespans (NPM), showing that in the first set of experiments, the presence or absence of calendars does not significantly impact performance, except for the Big size at uncertainty levels 0.1 and 1.0. From an empirical analysis of the experiments with the presence of resource calendars, we note

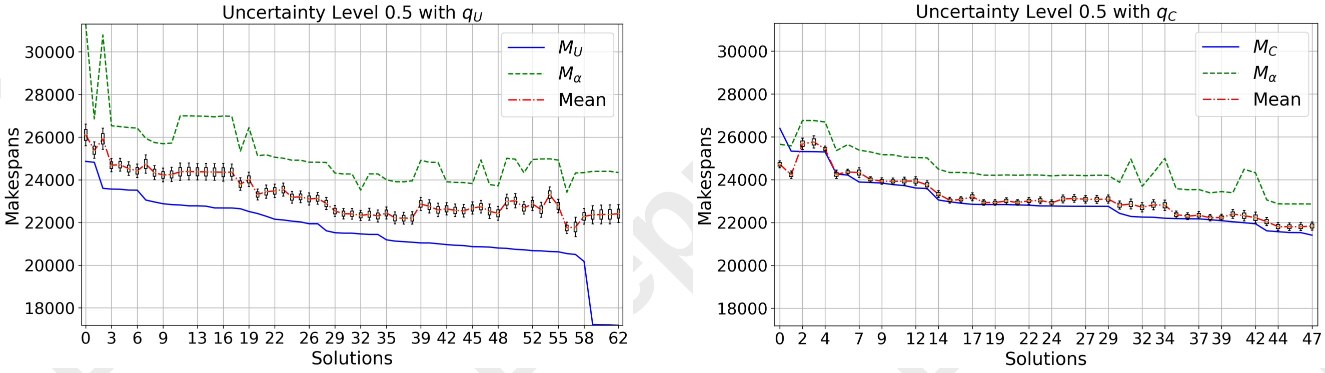
[4] We use the Python version of Google OR-Tools (v9.11.4210) as our CP solver.

[5] The evaluation section provides a detailed explanation of the k-solutions selection process.

[6] In the repository https://github.com/francescameneghello/IJCAI2025-Proactive-DataDriven-Scheduling-Business-Process provides further evidence of the differences between applying q_U and q_C. The latter allows finding M^*_α closer to M^*_C and, within the same solver time limit, achieves a smaller M^*_α with fewer solutions, as shown in Figure 3. The downside is that using q_C, we cannot guarantee the lower bound property proved in Proposition 1.

[7] A significant improvement in the ordered list of deterministic solutions is defined as a $\Delta_i = (M_{\text{C},j} - M_{\text{C},i})/M_{\text{C},j} \geq 0.20$ where solution j precedes solution i.

[8] In the repository https://github.com/francescameneghello/IJCAI2025-Proactive-DataDriven-Scheduling-Business-Process, three other problems of different sizes are reported.



(a) Definition of Deterministic BPSP with q_U

(b) Definition of Deterministic BPSP with q_C

Figure 3: Comparison of all M_C solutions returned by the CP solver, along with their corresponding M_α values and the *Mean* computed over 1,000 simulations, for the synthetic medium problem with calendars and with an uncertainty level of 0.5.

that a q_C with respect to q_U allows for representing activity durations closer to real times, enabling it to effectively leverage the planned availability and unavailability of resources to optimize the schedule. Figure 3b shows that the average Makespans overlap with M_C, and M_α is slightly higher. In contrast, q_U, as shown in Figure 3a, guarantees the lower bound (Section 3.1), but results in a worse solution compared to q_C. Variance in q may lead to a significant difference between the deterministic and the corresponding probabilistically minimal Makespans, even if their correlation remains.

4.3 DayHospital Experiment

The Data. In this experiment, we apply our approach to a real-world BPSP problem. Specifically, we use an entire year (2021) of data from DayHospital to learn the BPSP parameters and minimize the Makespan for the subsequent five months (January-May 2022), which serve as the test set. The event log includes timestamped activities along with their designated resources as shown in Table 1. This allows us to identify the activities, the precedence relationships, and the resources, including their capacities and calendars, to define the BPSP problem. Between 50 and 1,000 patients are treated every day in the hospital, depending on whether it is a weekday or a weekend, with over 1,600 activities taking place on the busiest days. To properly evaluate our approach, we compare the Makespan given from the simulation of the actual schedule and the M_α^* found with our approach.

Results. Table 3 reports the results achieved on the hospital dataset. We divided the days into small, medium, and big categories based on the number of activities treated in a day. For medium and big days, we observe, on average, a significant reduction in the global Makespan, up to 14%, which corresponds to a reduction of approximately 4 hours. In the case of small, the scope for improvement is limited, yet we still observe improvements, even if at a slightly lower percentage.

4.4 Discussion & Limitations

From the synthetic data evaluation, we verify the capability of our approach to optimize schedules for BPSP problems in the presence of various levels of uncertainty and planned resource unavailability. We define the k selection allowing us to save time and achieve improved M_α^* of the problem. As for resource calendars, we observe the importance of selecting a q_c that leads to producing M_α^* closer to the corresponding deterministic ones, thereby optimizing the utilization of available slots for the CP solver. The DayHospital experiment shows promising performance despite the limited clinical patient information available in the data (e.g., diagnosis, complexity, medical history). The use of RIMS [Meneghello *et al.*, 2025] can potentially leverage such information to significantly improve the estimation of activity durations.

Un. Level	0.1	0.1	0.5	0.5	1	1
Calendar	x	✓	x	✓	x	✓
Small	1.00	1.00	1.03	1.00	1.13	1.13
Medium	0.99	1.00	1.03	1.00	1.10	1.10
Big	1.09	1.00	1.07	1.07	1.12	1.18

Table 2: The NPM metrics are presented for each size, with and without the resource calendars.

Days	**Av. #Cases**	**Av. #Act.**	**Av. #Res.**	**Av. % Imp.**	**Max % Imp.**	**Av. Imp.**
Small	98	136	33	-1.2%	-6%	-14min
Medium	717	1242	237	-4.8%	-12%	-69min
Big	848	1505	266	-5.9%	-14%	-87min

Table 3: The days are divided into 3 groups based on 33rd and 67th percentiles.

Below, we list several additional limitations of our work. As mentioned earlier, several simplified assumptions were necessary to apply process mining techniques and infer the BPSP problem from event logs. Additionally, comparisons with other methods addressing job duration uncertainty, if extended to incorporate planned resource unavailability, remain unexplored. Alternative job scheduling solvers could also be applied after defining the deterministic BPSP problem, leveraging the proposed Monte Carlo simulation to achieve the minimum probabilistic Makespan. Lastly, our approach focuses solely on minimizing the global Makespan, while resource-based objectives – such as the minimization of costs or of idle resource times – could better address organizational needs in BPSP problems.

5 Related Work

Proactive and Data-Driven Scheduling. To the best of our knowledge, no other work combines together the unique features that characterize BPSP and that differentiate it from RCPSP, namely uncertain durations and resource unavailability. Several papers have studied the uncertainty in job durations such as [Satic *et al.*, 2022; Liu and Xu, 2020; Hauder *et al.*, 2020; Chen *et al.*, 2019; Beck and Wilson, 2007; Gómez *et al.*, 2023]; yet those works do not address resource unavailability.

In contrast, there are few papers on RCPSP constrained by resource time-window, mainly for human resources with working time constraints, resources with planned maintenance attributes [Ding *et al.*, 2023] or resource calendars [Kreter *et al.*, 2018]. [Tian *et al.*, 2018] address resource unavailability planning by defining randomly generated periods during which certain resources are unavailable. However, these works do not address duration uncertainty.

Lastly, the majority of studies test their solutions using generated instances, while only a small portion, about 11%, utilize real-world-based instances [Sánchez *et al.*, 2023]. The only exception is the work of [Senderovich *et al.*, 2019]. Our approach extends [Senderovich *et al.*, 2019] by incorporating stochastic activity durations, as well as by discovering duration distributions and resource calendars from event data.

Resource Allocation in Process Mining. Several works in process mining have addressed the problem of resource allocation, which aims to ensure that each activity in a process case is executed at the right time and with the right resources [Kumar *et al.*, 2002]. Much of the research in this area focuses on online resource allocation, involving reactive scheduling where tasks are dynamically assigned to resources at runtime. These strategies are particularly useful in settings where information becomes available over time, such as the appearance of unexpected cases or unforeseen resource unavailability. Various techniques have been proposed to address these problems, including batch allocation strategies [Delias *et al.*, 2011; Arias *et al.*, 2018; Zeng and Zhao, 2005], predictive allocation [Park and Song, 2019], reinforcement learning approaches [Huang *et al.*, 2011; Żbikowski *et al.*, 2023; Middelhuis *et al.*, 2025; Beerepoot *et al.*, 2023; Meneghello *et al.*, 2024], as well as formulations based on assignment problems or parallel machine scheduling [Kunkler and Rinderle-Ma, 2024].

Fewer works have explored offline scheduling, where an initial schedule or roster is required from the outset. For instance, in [Havur *et al.*, 2022], the problem is formalized as an Answer Set Programming (ASP) formulation, enabling an ASP solver to compute a schedule. Similarly, [Aalst, 1996; Doerner *et al.*, 2006] use a Petri net formalization of control flow and resource perspectives, solving the problem through the reachability graph. Notably, only [Doerner *et al.*, 2006] considers stochasticity in activity durations. These approaches lack mechanisms to ensure schedule robustness under temporal fluctuations. In this context, [Di Cunzolo *et al.*, 2024] is the only work that integrates operations research with PPM predictive models to produce robust schedules. Despite these advances, none of the existing methods considers planned resource unavailabilities in a proactive setting.

6 Conclusion

In this work, we introduced the Business Process Scheduling Problem (BPSP) to address the challenges of scheduling business processes with stochastic activity durations and planned resource unavailability. Our solution framework combines process mining for parameter inference, deterministic transformations for uncertainty modeling, constraint programming for robust scheduling, and Monte Carlo simulation for probabilistic evaluation. Through evaluation on synthetic datasets, we demonstrated the adaptability of our solution to varying uncertainty levels, problem sizes, and resource configurations, achieving effective optimization of the probabilistic Makespan. Real-world validation on hospital data showcased the utility of the approach, achieving up to 14% reductions in global Makespan.

Future work will focus on addressing the limitations identified in this study. Enhancing the scalability of the framework to handle larger, more complex business processes and integrating adaptive real-time rescheduling capabilities will extend its applicability to realistic environments. Exploring multi-objective optimization to balance trade-offs such as cost, resource utilization, and Makespan could further align the framework with organizational goals when scheduling business processes. Additionally, refining parameter inference methods to handle incomplete or noisy event logs and incorporating contextual data would significantly improve the practicality of the approach across diverse domains. Lastly, while we currently use Monte Carlo simulation to handle uncertainty by sampling from the full distribution of activity durations, we aim to explore stochastic programming in future work. Stochastic programming offers a more structured approach by optimizing over a finite set of scenarios with assigned probabilities, potentially reducing uncertainty more effectively. However, its main limitation lies in the need to predefine a manageable set of scenarios, which is impractical in our case due to the vast number of possible realizations. As we introduce new uncertainties—such as variability in activity sequences—scenario-based methods may become more tractable, making stochastic programming a promising direction for future exploration.

Acknowledgements

We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU, the HORIZON 2020 HumanE-AI project (Grant 952026), and the PRIN project

PINPOINT Prot. 2020FNEB27, CUP H23C22000280006 and H45E2100021000.

References

[Aalst, 1996] Wil van der Aalst. Petri net based scheduling. *Operations-Research-Spektrum*, 18:219–229, 12 1996.

[Arias *et al.*, 2018] Michael Arias, Jorge Munoz-Gama, Marcos Sepúlveda, and Juan Carlos Miranda. Human resource allocation or recommendation based on multi-factor criteria in on-demand and batch scenarios. *European Journal of Industrial Engineering*, 12:364–404, 2018.

[Beck and Wilson, 2007] J Christopher Beck and Nic Wilson. Proactive algorithms for job shop scheduling with probabilistic durations. *Journal of Artificial Intelligence Research*, 28:183–232, 2007.

[Beerepoot *et al.*, 2023] Iris Beerepoot, Claudio Di Ciccio, Hajo A. Reijers, Stefanie Rinderle-Ma, Wasana Bandara, Andrea Burattin, Diego Calvanese, Tianwa Chen, Izack Cohen, Benoît Depaire, Gemma Di Federico, Marlon Dumas, Christopher van Dun, Tobias Fehrer, Dominik A. Fischer, Avigdor Gal, Marta Indulska, Vatche Isahagian, Christopher Klinkmüller, Wolfgang Kratsch, Henrik Leopold, Amy Van Looy, Hugo Lopez, Sanja Lukumbuzya, Jan Mendling, Lara Meyers, Linda Moder, Marco Montali, Vinod Muthusamy, Manfred Reichert, Yara Rizk, Michael Rosemann, Maximilian Röglinger, Shazia Sadiq, Ronny Seiger, Tijs Slaats, Mantas Simkus, Ida Asadi Someh, Barbara Weber, Ingo Weber, Mathias Weske, and Francesca Zerbato. The biggest business process management problems to solve before we die. *Computers in Industry*, 146:103837, 2023.

[Bonfietti *et al.*, 2014] Alessio Bonfietti, Michele Lombardi, and Michela Milano. Disregarding duration uncertainty in partial order schedules? yes, we can! In Helmut Simonis, editor, *Integration of AI and OR Techniques in Constraint Programming - 11th International Conference, CPAIOR 2014, Cork, Ireland, May 19-23, 2014. Proceedings*, volume 8451 of *Lecture Notes in Computer Science*, pages 210–225. Springer, 2014.

[Camargo *et al.*, 2020] Manuel Camargo, Marlon Dumas, and Oscar González. Automated discovery of business process simulation models from event logs. *Decis. Support Syst.*, 134, 2020.

[Chaari *et al.*, 2014] Tarek Chaari, Sondes Chaabane, Nassima Aissani, and Damien Trentesaux. Scheduling under uncertainty: Survey and research directions. In *2014 International conference on advanced logistics and transport (ICALT)*, pages 229–234. IEEE, 2014.

[Chen *et al.*, 2019] HaoJie Chen, Guofu Ding, Jian Zhang, and Shengfeng Qin. Research on priority rules for the stochastic resource constrained multi-project scheduling problem with new project arrival. *Computers & Industrial Engineering*, 137:106060, 2019.

[Delias *et al.*, 2011] Pavlos Delias, Anastasios Doulamis, Nikolaos Doulamis, and Nikolaos Matsatsinis. Optimizing resource conflicts in workflow management systems. *IEEE Transactions on Knowledge and Data Engineering*, 23(3):417–432, 2011.

[Di Cunzolo *et al.*, 2024] Matteo Di Cunzolo, Massimiliano Ronzani, Roberto Aringhieri, Chiara Di Francescomarino, Chiara Ghidini, Alberto Guastalla, and Emilio Sulis. Robust solutions via optimisation and predictive process monitoring for the scheduling of the interventional radiology procedures. *International Transactions in Operational Research*, n/a(n/a), 2024.

[Ding *et al.*, 2023] Hongyan Ding, Cunbo Zhuang, and Jianhua Liu. Extensions of the resource-constrained project scheduling problem. *Automation in Construction*, 153:104958, 2023.

[Doerner *et al.*, 2006] Karl Doerner, Walter J. Gutjahr, Gabriele Kotsis, Martin Polaschek, and Christine Strauss. Enriched workflow modelling and stochastic branch-and-bound. *European Journal of Operational Research*, 175(3):1798–1817, 2006.

[Dumas *et al.*, 2018] Marlon Dumas, L Marcello Rosa, Jan Mendling, and A Hajo Reijers. *Fundamentals of business process management*. Springer, 2018.

[Gómez *et al.*, 2023] Mario Flores Gómez, Valeria Borodin, and Stéphane Dauzère-Pérès. Maximizing the service level on the makespan in the stochastic flexible job-shop scheduling problem. *Comput. Oper. Res.*, 157:106237, 2023.

[Hauder *et al.*, 2020] Viktoria A Hauder, Andreas Beham, Sebastian Raggl, Sophie N Parragh, and Michael Affenzeller. Resource-constrained multi-project scheduling with activity and time flexibility. *Computers & Industrial Engineering*, 150:106857, 2020.

[Havur *et al.*, 2022] Giray Havur, Cristina Cabanillas, and Axel Polleres. Benchmarking answer set programming systems for resource allocation in business processes. *Expert Systems with Applications*, 205:117599, 05 2022.

[Huang *et al.*, 2011] Zhengxing Huang, W.M.P. van der Aalst, Xudong Lu, and Huilong Duan. Reinforcement learning based resource allocation in business process management. *Data & Knowledge Engineering*, 70(1):127–145, 2011.

[Johnson, 2016] Ralph J Johnson. A comprehensive review of an electronic health record system soon to assume market ascendancy: Epic. *J Healthc Commun*, 1(4):36, 2016.

[Kreter *et al.*, 2018] Stefan Kreter, Andreas Schutt, Peter J Stuckey, and Jürgen Zimmermann. Mixed-integer linear programming and constraint programming formulations for solving resource availability cost problems. *European Journal of Operational Research*, 266(2):472–486, 2018.

[Kumar *et al.*, 2002] Akhil Kumar, Wil Aalst, and H. Verbeek. Dynamic work distribution in workflow management systems: How to balance quality and performance. *J. of Management Information Systems*, 18:157–194, 01 2002.

[Kunkler and Rinderle-Ma, 2024] Michel Kunkler and Stefanie Rinderle-Ma. Online resource allocation to process tasks under uncertain resource availabilities. In *2024 6th International Conference on Process Mining (ICPM)*, pages 137–144, 2024.

[Liu and Xu, 2020] Dongning Liu and Zhe Xu. A multi-pr heuristic for distributed multi-project scheduling with uncertain duration. *IEEE Access*, 8:227780–227792, 2020.

[Meneghello *et al.*, 2024] Francesca Meneghello, Jeroen Middelhuis, Laura Genga, Zaharah Bukhsh, Massimiliano Ronzani, Chiara Di Francescomarino, Chiara Ghidini, and Remco Dijkman. Optimizing resource allocation policies in real-world business processes using hybrid process simulation and deep reinforcement learning. In Andrea Marrella, Manuel Resinas, Mieke Jans, and Michael Rosemann, editors, *Business Process Management*, pages 167–184, Cham, 2024. Springer Nature Switzerland.

[Meneghello *et al.*, 2025] Francesca Meneghello, Chiara Di Francescomarino, Chiara Ghidini, and Massimiliano Ronzani. Runtime integration of machine learning and simulation for business processes: Time and decision mining predictions. *Inf. Syst.*, 128:102472, 2025.

[Middelhuis *et al.*, 2025] Jeroen Middelhuis, Riccardo Lo Bianco, Eliran Sherzer, Zaharah Bukhsh, Ivo Adan, and Remco Dijkman. Learning policies for resource allocation in business processes. *Information Systems*, 128:102492, 2025.

[Park and Song, 2019] Gyunam Park and Minseok Song. Prediction-based resource allocation using lstm and minimum cost and maximum flow algorithm. In *2019 International Conference on Process Mining (ICPM)*, pages 121–128, 2019.

[Pinedo, 2012] Michael L Pinedo. *Scheduling*, volume 29. Springer, 2012.

[Reijnen *et al.*, 2023] Robbert Reijnen, Kjell van Straaten, Zaharah Allah Bukhsh, and Yingqian Zhang. Job shop scheduling benchmark: Environments and instances for learning and non-learning methods. *CoRR*, abs/2308.12794, 2023.

[Sánchez *et al.*, 2023] Mariam Gómez Sánchez, Eduardo Lalla-Ruiz, Alejandro Fernández Gil, Carlos Castro, and Stefan Voß. Resource-constrained multi-project scheduling problem: A survey. *European Journal of Operational Research*, 309(3):958–976, 2023.

[Satic *et al.*, 2022] Ugur Satic, Peter Jacko, and Christopher Kirkbride. Performance evaluation of scheduling policies for the dynamic and stochastic resource-constrained multi-project scheduling problem. *International Journal of Production Research*, 60(4):1411–1423, 2022.

[Senderovich *et al.*, 2015] Arik Senderovich, Andreas Rogge-Solti, Avigdor Gal, Jan Mendling, Avishai Mandelbaum, Sarah Kadish, and Craig A. Bunnell. Data-driven performance analysis of scheduled processes. In Hamid Reza Motahari-Nezhad, Jan Recker, and Matthias Weidlich, editors, *Business Process Management - 13th International Conference, BPM 2015, Innsbruck, Austria, August 31 - September 3, 2015, Proceedings*, volume 9253 of *Lecture Notes in Computer Science*, pages 35–52. Springer, 2015.

[Senderovich *et al.*, 2019] Arik Senderovich, Kyle EC Booth, and J Christopher Beck. Learning scheduling models from event data. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, pages 401–409, 2019.

[Shoush and Dumas, 2022] Mahmoud Shoush and Marlon Dumas. When to intervene? prescriptive process monitoring under uncertainty and resource constraints. In *International Conference on Business Process Management*, pages 207–223. Springer, 2022.

[Tian *et al.*, 2018] Jinwen Tian, Xingye Dong, and Sheng Han. Optimizing for a resource-constrained multi-project scheduling problem with planned resource unavailability. In *2018 3rd International conference on modelling, simulation and applied mathematics (MSAM 2018)*, pages 243–248. Atlantis Press, 2018.

[van der Aalst, 2016] Wil van der Aalst. *Process Mining: Data Science in Action*. Springer Publishing Company, Incorporated, 2nd edition, 2016.

[Winklehner and Hauder, 2022] Philipp Winklehner and Viktoria A Hauder. Flexible job-shop scheduling with release dates, deadlines and sequence dependent setup times: A real-world case. *Procedia Computer Science*, 200:1654–1663, 2022.

[Xu *et al.*, 2016] Jiajie Xu, Chengfei Liu, Xiaohui Zhao, Sira Yongchareon, and Zhiming Ding. Resource management for business process scheduling in the presence of availability constraints. *ACM Transactions on Management Information Systems (TMIS)*, 7(3):1–26, 2016.

[Yang *et al.*, 2020] Dongsheng Yang, Xianyu Zhou, Zhile Yang, Qiangqiang Jiang, and Wei Feng. Multi-objective optimization model for flexible job shop scheduling problem considering transportation constraints: A comparative study. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2020.

[Żbikowski *et al.*, 2023] Kamil Żbikowski, Michał Ostapowicz, and Piotr Gawrysiak. Deep reinforcement learning for resource allocation in business processes. In Marco Montali, Arik Senderovich, and Matthias Weidlich, editors, *Process Mining Workshops*, pages 177–189, Cham, 2023. Springer Nature Switzerland.

[Zeng and Zhao, 2005] Daniel D. Zeng and J. Leon Zhao. Effective role resolution in workflow management. *INFORMS Journal on Computing*, 17(3):374–387, 2005.