Human Resource Allocation in Business Process Management

Bachelor Thesis

presented by
Marco Schulz
Matriculation Number 1691483

submitted to the
Data and Web Science Group
Prof. Dr. Han van der Aa
University of Mannheim

December 2022

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Optimal resource allocation will have a huge impact on process performance and total cost. So for a company, it is very important to use the available resources in an efficient way. Thereby, resources can be human or non-human agents like machines or software. Consider the assembly line work in a car factory, where several different human (employees) and non-human resources (machines, robots) work at manufacturing processes for different kind of automobiles at the same time. Each resource executes each task more or less efficient, especially for human resources the task performance fluctuates over time. In this case, it is necessary to find the best fitting resource for each task in such a way, that the overall process performance is improved and the total process cost is minimized. The main problem is to find the best performing resource for each task with the available knowledge of the process data. With the help of process analysis, it is possible to gain historic data from process-logs to find the best allocation strategy for each business process.

1.1 Problem Statement

Business Process Management (BPM) helps companies to manage, model and improve their business processes [13]. One main domain of BPM is Business Process Optimization (BPO), which goal is the election of the right process designs and the application of the most appropriate optimization techniques [10]. An important sub-discipline of BPO is the human resource allocation problem, due to the fact that the allocation of available human resources to process tasks has a significant impact on process performance, costs and the efficient usage of resources during the process execution [2]. Human resource allocation has to tackle with a high complexity because of multiple allocation goals, criteria, and contains like a different workload level or unpredictable task performance over time. In recent years, several papers were released that propose various techniques for human resource allocation based on different approaches [3], but there is no empirical comparison between all these techniques. This paper addresses this gap by 1) structuring the human resource allocation problem in different problem categories and 2) evaluating three allocation techniques against each other with real-world data within one category.

The remainder of this paper is organized as follows: First, section 1.2 provides the related work of three literature reviews for optimal resource allocation in BPM. Then, in chapter 2 an introduction in the theoretical framework of resource allocation and different allocation criteria is given. After that, chapter 3 explains at first the exact research methodology before establishing an overview over several human resource allocation techniques, which are structured in different problem categories. Chapter 4 evaluates and compares three different allocation approaches against each other, and finally in chapter 5 a summary of the paper and a view for future work is given.

1.2 Related Work

There was just one paper found that provides an overview of different techniques and compares them in terms of resource allocation in BPM. Luise Pufahl et al. [11] did a systematic literature survey in 2021 about automatic resource allocation in business processes, where they were analyzing various approaches regarding their allocation goals and capabilities, process models and process data, solution strategies and their maturity. They proposed, that the most techniques covers 1-to-1 allocation with process-oriented allocation goals and uses process models or process data to prepare the input. Furthermore, they found out that several different solution techniques for an optimal allocation are used, such as rule or logic programming, machine learning, heuristic methods and linear programming. Finally, the most papers offer an evaluation of their prototypes, but the provided prototypes are often only pseudocode or not accessible.

Nevertheless, two recent released literature reviews offer a similar course of action. On the one hand, Shyalika et al. [12] published a review in 2020 about several approaches for dynamic task scheduling, but they only focused on reinforcement learning strategies in a dynamic and uncertain environment. This literature review describes a few reinforcement learning approaches and frameworks and lists their merits and demerits.

On the other hand, Michael Arias et al. [3] put out a systematic mapping study in 2018 over all literature with respect to human resource allocation in BPM and process mining. This literature review gives a good classification over 95 papers that were published between 2005 and 2016. Their results confirm a growing interest in this research area in recent years, especially since 2011 in Asia and Europe, and point out that the most used venues are journals and conferences. Furthermore, Michael Arias et al. found out, that the most used research types are not the proposal of a new solution, but validation and evaluation research in which simulations and case studies are the mostly used evaluation methods. Here you can see that the approaches of human resource allocation are also validated with real-world data and not only with synthetic data or prototypes.

Chapter 2

Theoretical Framework

Before the different human resource allocation approaches and their problem categories are presented, the following chapter takes a look at the theoretical framework of human resource allocation in general. First, some preliminaries and important definitions for resource allocation are clarified in section 2.1. After that, section 2.2 describes the concept of resource allocation criteria.

2.1 Preliminaries of Resource Allocation

Resource allocation is an important issue in business process management. In companies each organization has a bunch of business processes to be executed. These business processes are often described with the help of process models, that contains the sequence of all tasks of the business process. The tasks of business processes are performed by several resources, which can be machines or software, but in the most cases they are humans. For an optimal business process management it is very important to assign these resources efficiently with the help of different allocation algorithms, which have to comply with diverse business constraints, such as time, cost or quality goals (as you can see in Figure 2.1).

In the last years, process data in the form of event-logs is becoming the typical way to gain historic information about business processes. From these event-logs it is possible to mine resource preferences, resource allocation rules or heuristics to develop different algorithmic resource allocation techniques. The techniques often follow a wide range of underlying programming approaches, such as: machine learning, linear programming, rule or logic programming. With the help of these allocation techniques, the performance of business processes is increasing, and they are becoming more efficient.

For resource allocation exist four different types regarding the resource alloca-

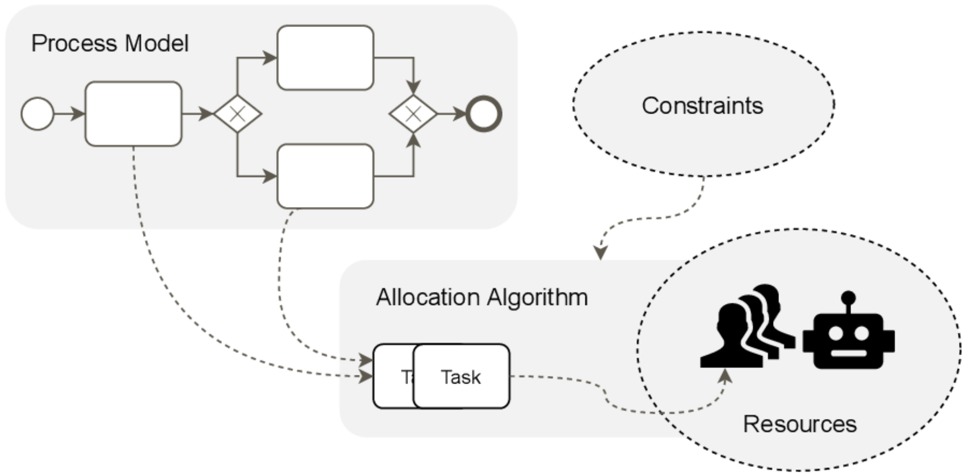

Figure 2.1: Resource Allocation from a given process model [11].

tion capability, which are defined by their relation of tasks to resources:

Definition 1 *1-to-1 allocation: one resource is allocated to one task*

Definition 2 *1-to-many allocation: a team of resources is allocated to one task*

Definition 3 *many-to-1 allocation: one resource is allocated to a set of tasks*

Definition 4 *many-to-many allocation: a team of resources is allocated to a set of tasks*

The work in this paper focuses on human resource allocation, where all resources that have to be assigned to tasks are humans. When resources are human, the complexity of optimal resource allocation increases, because each human has a different work load level and an unpredictable performance over time. Therefore, several allocation criteria exist to find the best fitting resource for a specific tasks. These criteria are described in the following section.

2.2 Allocation Criteria

Another important aspect of human resource allocation are the Allocation Criteria. The Allocation Criteria describe on the one hand the main goals of the different techniques, and on the other hand with the help of which resource characteristics

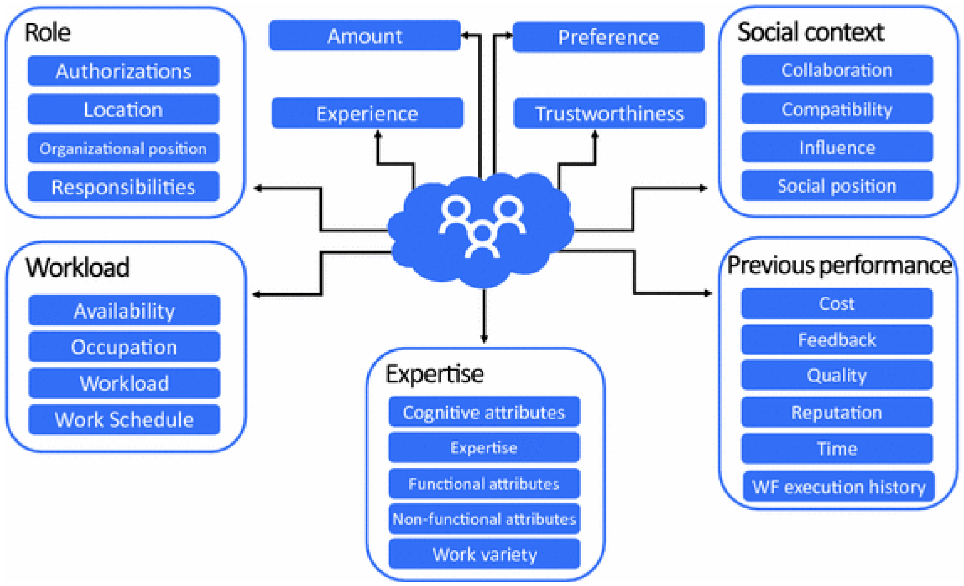

Figure 2.2: Taxonomy of Allocation Criteria [1].

try these techniques to reach their goals. Especially in Human Resource Allocation the different Allocation Criteria plays a huge roll, because if the resources are humans, there are much more different criteria to be considered as if the resources are non-human. The traditional way to find appropriate allocation criteria is by means of questionnaires or interviews of the involved staff. These techniques are very time- and cost-intense and also are rather subjective, so there is a need of providing rules for mining human allocation criteria from event logs by process mining.

In recent literature there are quite a few papers addressing this issue, so Michael Arias et al. [1] provide a taxonomy of human resource allocation criteria used in all human resource allocation approaches between 2005 and 2016. This taxonomy gives a good overview over several allocation criteria which are separated into five group criteria: Role, Workload, Expertise, Previous Performance and Social context. These group criterion has again several sub-criteria. The authors found out, that “Authorizations” from the Role-group, “Availability” from the Workload-group and “Expertise” from the Expertise-group are the most frequently used criteria. Furthermore, they supposed that there is a trend for multi-criteria approaches using distinct criteria in recent years. The full taxonomy of Michael Arias et al. can be seen in Figure 2.2.

To support selecting appropriate allocation criteria, Huang et al. [5] describe a resource allocation rules mining approach, which discovers interesting rules from event logs. This technique is able to extract more rules in an efficient and faster way for an optimal resource allocation in BPM. Another interesting approach is provided by Linh Tao Ly et al. [9], they describe a staff assignment mining technique to gather staff assignment rules using information from audit trail data and organizational information by using decision tree learning.

Chapter 3

Problem Structuring

The following chapter will first describe the exact research methodology of the literature review. Then, the human resource allocation problem is structured by providing a classification of different problem categories. After that, the different human resource allocation approaches are described precisely and are associated to a proper problem category.

3.1 Research Methodology

For the literature research of the several resource allocation techniques in chapter 3 only papers that were published in 2008 and later are considered. Given that most of them are not prior than 2015, you can say that this paper is a relatively up-to-date analysis of human resource allocation in BPM. To provide an overview of the topic, a list of relevant papers and articles has been generated by using the literature search engine "Google Scholar". Thereby, one of the keywords of *Human resource allocation, Resource allocation, Resource assignment, Staff assignment, Task scheduling* or *Task assignment* was combined with another keyword of *Business process management, Business process optimization* or *process mining*.

To list only the relevant papers, they were first filtered by the title. If the title is convenient with the keywords, the abstract and conclusion of this paper were skimmed. All articles that seemed valuable were added to the list. After that, the papers on the list have been read in detail and were put into order by their usability and how often they were cited by other papers for the topic of human resource allocation. To make it possible to evaluate the different approaches, it is important, that all papers provide a prototype or at least some pseudocode of the implementation for their techniques. So, all gathered papers which do not fulfill these requirements were not considered in the deeper analysis. After that, the remaining papers were classified into five problem categories, which are described in the following chapter. Finally, the most promising papers were considered, so that at least two approaches can be classified into each provided category.

3.2 Problem Categories

The first step is to figure out the different problem categories for Human Resource Allocation in BPM. Therefore, the provided allocation techniques are first analyzed by their point in time when they do the actual allocation. So, some techniques are **upfront planning** approaches, which means that the resource allocation has to be done before the process execution and on the other hand some techniques make the assignment **dynamically** or “on-the-fly”, by frequently updating their resource allocation planning during the process execution. If this differentiation is done, the approaches can be analyzed again by their main optimization goal. This can be either **minimizing the total process cycle time** or **minimizing the process total cost**, where both result in an optimization of the allocation schedule. These four problem categories focus mainly on 1-to-1 allocations, which means that one tasks has to be assigned to exact one resource. But there are a few techniques existing, that provide a 1-to-many allocation, where a team of resources is needed, to complete a task. So, this leads to an additional problem category, where the provided approaches focus especially on **team allocation**. A good overview of all presented human resource allocation categories can be seen in Table 3.1.

Upfront planning		**Dynamic allocation**	
Minimize process cycle time	**Minimize process cost**	**Minimize process cycle time**	**Minimize process cost**
Human Resource Allocation based on Process Mining [22]	Task Operation Model (TOM) + Ant Colony Optimization (ACO) [6]	Naive Bayes (NBSR) [14]	Reinforcement Learning (RLRAM) [7]
Answer Set Programming (ASP) + Dependencies [4]	Optimizing Resource Allocation by Business Process Improvement [18]	Resource Allocation by Minimizing Cycle Time [17]	Optimizing Cost and Maximum Throughput [16]
		Particel Swarm Optimization (PSO) [23]	

Table 3.1: The Human Resource Allocation Categories.

3.3 The Techniques

In this section, the different techniques of human resource allocation are presented and associated into the provided problem categories of section 3.2. So, for each approach, the *Allocation Time* and the regarding *Optimization Goal* shows in which category of Human Resource Allocation the technique can be classified. Then, the exact *Methodology* of the several techniques and their underlying approaches regarding the specific *Allocation Type* and *Allocation Criteria* are described. After that, the *Model Input*, which describes the input data of each technique, is specified. Finally, the generated model or framework is described in the *Model Output*. After all, a comparison of all provided techniques can be found in Table 3.2.

3.3.1 Upfront Planning - Minimize Process Cycle Time

Human Resource Allocation based on Process Mining [22]. This upfront-planning technique of Human Resource Allocation is based on Process Mining. The main *Optimization Goal* and *Allocation Criteria* is to minimize the total process execution time under three constraints: resource preference, resource availability and total cost. Moreover, the authors assume that the turnaround time, which is the time between the start and end of two neighboring activities during the process execution, has a large impact on the total process time. To minimize this turnaround time, the collaboration between resources should be maximized.

The *Model Input*-data can be got by mining event logs. For the *Model Output*, the approach provides a resource allocation model with a 1-to-1 allocation, where you can see each resource and their allocated activity, corresponding role and the start and end time of the activity. The authors demonstrate in their paper that the proposed approach has a better average process time over different log sizes than the Maximizing Compatibility [8] approach, which will be analyzed further on.

Answer-Set-Programming (ASP) with Dependencies [4]. The second approach in this category provides an upfront-planning conceptualization of resource allocation under real dependencies and potential conflicts. These dependencies can be resource requirements or temporal requirements, which can lead to working on one item blocks resources such that other work items cannot be worked on. Here the *Allocation Criteria* is to optimize the resource utilization which leads in a minimal execution time.

The approach finds the optimal schedule from the *Model Input* of a resource ontology in RDF schema (RDFS) by using Answer-Set-Programming. As *Model Output* the technique provides one or more resource sets with the set of resources that can potentially allocated to an activity in a 1-to-1 relationship between tasks

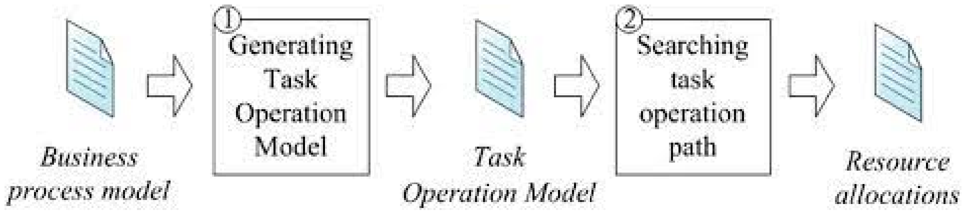

Figure 3.1: Methodology of the TOM + ACO technique. [6]

and resources. The authors of this technique, tested their solution against a Greedy approach and the results shows, that it has a 14% time usage improvement and an improvement of 7% in terms of average employee utilization compared to the Greedy approach.

3.3.2 Upfront Planning - Minimize Process Cost

Task Operation Model (TOM) + Ant-Colony-Optimization (ACO) [6]. This technique can also be categorized as an upfront-planning approach. Its *Allocation Criteria* is to improve the global utility of the process case by maximizing the availability and minimizing the cost of each task operation. The regarding methodology of the approach is shown in Fig. 2.

From the *Model Input* of a particular business process model, the approach automatically generates the associated Task Operation Model (TOM) and searches the optimal task operation path on the TOM by using an Ant Colony Optimization Approach (ACO). The authors of this approach could prove, that the accuracy of the Ant-Colony-based strategies are 100%, which indicates that the ACO can always find the optimal task operation path among the TOM. This path represents the *Model Output* of the technique with a 1-to-1 *Allocation Capability*. But the proposed approach can only consider one business case in isolation, and not multiple cases at one time.

Optimizing Resource Allocation by Business Process Improvement [18]. The upfront-planning technique provided in this paper focuses on Business Process Improvement. It optimizes the resource allocation planning by a smart change of the process structure before the execution of the business process. First, a basic allocation strategy is used to minimize the total expense. If the time constraint of the process is violated, an adjustment strategy is applied, to improve the execution time until the time limit is satisfied. So the *Allocation Criteria* and main *Optimization Goal* of this approach is to minimize the total process cost and to fulfill the time limit.

From the *Model Input* of a role-based business process model, the approaches will provide you as *Model Output* an optimal resource allocation table, where every resource is allocated to a specific task and role (1-to-1 allocation) with the regarding start and end time. Furthermore, you get the new process structure after the changes of the resource allocation planning. Thereby, all process constraints (time and cost) and dependencies pre-defined by the process structure are considered. Certainly, if it is however not possible to modify or change the process structure in your business process, then this technique is not very suitable.

3.3.3 Dynamic Allocation - Minimize Process Cycle Time

Naive Bayes (NBSR) [14]. In this approach, the resource allocation planning is done "on-the-fly" which means, that the planning will be frequently updated and executed during the execution time by considering recent human resource performance. The main *Optimization Goal* for this algorithm is to minimize the total process completion time.

For the *Model Input* you need a Naive-Bayes-Model (NBM) with one target node and five child nodes. So this approach try to predict the current human performance, which is the target node of the NBM, by considering the following five dependent factors as *Allocation Criteria* of the process execution, representing the five child nodes: the Performer, the Activity, the Queue in front of the activity, the Inter-arrival Rate of the system and finally the Daytime of the working shift. From that model, the Naive-Bayes-Selection-Rule algorithm (NBSR) gives you the *Model Output* as the appropriate human resource to perform a process instance in the activity a_i at a time t (1-to-1 allocation). The actual NBSR(a_i, $Q_a(t)$, R_a, BN, $D_a(t)$, $I(t)$) algorithm is shown in algorithm 1 with the parameters R_a is the set of all human resources employed in a_i, $Q_a(t)$ is the queue before a_i at a time t, BN is the NBM, $D_a(t)$ is the daytime at time t and $I(t)$ is the inter-arrival rate at time t. In addition, the authors proofed in [15] that the NBSR algorithm outperforms several static priority rules (like ORDER, LIDDLE, SIDDLE) in terms of completion time and waiting time.

Resource Allocation by Minimizing Cycle Time [17]. As the name of these techniques implies, it is a dynamic task assignment approach with the *Optimization Goal* of minimizing the cycle time of business processes. Firstly, this paper is one of the few approaches in this list, that considers also many-to-many relationships between resources by using an individual worklist model. Furthermore, it respects the different arrival rate of process instances and proposes a method based on stochastic- and queuing-theory, which can be applied when multiple identical

Algorithm 1

NBSR(a_i, $Q_a(t)$, R_a, BN, $D_a(t)$, $I(t)$)

BOOLEAN $loop \leftarrow$ TRUE
RESOURCE res
DOUBLE $tmp \leftarrow$ -9999
while $loop$ = TRUE **do**
  for INT $index \leftarrow$ 0 to SIZE(R_a) - 1 **do**
    $value \leftarrow$ DOINFERENCE(a_i, $Q_a(t)$, R_a, BN, $D_a(t)$, $I(t)$) $\triangleright$ *probability function in BN*
    if $tmp < value$ && r_{index} IS IDLE **then**
      $tmp \leftarrow value$
      $res \leftarrow r_{index}$
    end if
  end for
  if res != NIL **then**
    $loop \leftarrow$ FALSE
  end if
end while
return res

resources exist for estimating the mean cycle time of activities.

The *Model Input* of this approach is a formal business process model that considers the quantity of each resource as a new parameter. As *Model Output* it presents a table with the task assignment probabilities of each resource for minimizing the cycle time of the business process.

Particle-Swarm-Optimization (PSO) [23]. The last approach in this problem category proposes a dynamic resource allocation method, that is based on Particle-Swarm-Optimization (PSO) in multi-process instance environment, which are composed of multi-process tasks and different kinds of resources. So, running instances at the same time will lead to the problem, that resource conflicts can occur. PSO is a model based on swarm intelligence, which controls the behavior by updating the particles in the swarm, where each particle represents a possible solution to the problem. The main *Allocation Criteria* of this technique is to maximize the process performance, and it has 1-to-1 allocation capabilities.

The resource allocation *Model Input* proposed in this approach can fully consider the resource cost, time and other performance evaluation indicators. From that, the technique finds the global optimal solution of resource scheduling as *Model Output*.

3.3.4 Dynamic Allocation - Minimize Process Cost

Reinforcement Learning (RLRAM) [7]. This approach uses Reinforcement Learning to be able to perform a dynamic resource allocation based on the interactions with the environment. The *Allocation Criteria* for this approach are to minimize the long-term cost and to improve the performance of the business process execution.

As *Model Input* you need a Markov-Decision-Process-Model. The Reinforcement-Learning-Based-Resource-Allocation-Mechanism (RLRAM) assigns the work items of newly arrived process states to appropriate resources. Then, the algorithm uses a Q-learning method to evaluate these allocation policy by trying to minimize the cost-function of the business process environment. Thereby, it is possible to fine-tune the parameters of the Q-learning method. For the *Model Output* of this approach, you get an allocated ordered queue of work items of each resource. This technique outperforms a standard Greedy heuristic, but it is computationally demanding and therefore inappropriate for large decision problems.

Based on the RLRAM algorithm, Mehdi Yaghoubi and Morteza Zahedi [19] describe in their paper an enhanced version of the provided approach above. This approach try to minimize the entropy of work list for each resource, that is based on task similarities. So if the number of similar tasks in a work list increases, the entropy of work list decreases, because similar tasks have some common sub-tasks. The algorithm tries to inject similar tasks into the work list of each resource, which leads to a cycle time reduction. The combination of this technique and the RLRAM algorithm outperforms the normal RLRAM from Huang et al. [7] and the Naive Bayes [14] approach in terms of cycle time.

Resource Allocation by Optimizing Cost and Maximum Throughput [16]. The authors of this technique provide a dynamic resource allocation and task assignment approach for optimizing the cost and maximum throughput of business processes, which represents the *Allocation Criteria*. It considers different efficiency among resources in many-to-many relationships by using numerical analysis and genetic algorithms.

Therefore, it needs only some statistical data as *Model Input* and can find out the optimal solution as *Model Output* faster and with fewer data compared to non-numerical methods, which require detailed input data from event logs. However, in this approach, the resource costs are fixed and can not change during the process execution.

3.3.5 Team Allocation

Maximizing Compatibility [8]. For this upfront-planning approach, the main *Allocation Criteria* is to maximize the compatibility among actors in workflows for resource assignment, which means it is a 1-to-many allocation technique and can be classified as team allocation. The approach needs for the *Model Input* a compatibility matrix, which depicts the compatibility for each pair of actors in the workflow. As it could be problematic to create this matrix manually, the authors in this paper propose another approach, which can generate the compatibility matrix automatically from the execution log of the business process. After that, the Model for Optimal Work Assignment (OWA) gives you as *Model Output* the optimal assignment for each resource.

This technique is very quick for medium-seized problems, and even 20% faster than a normal Greedy approach. On the other side, is the complexity for creating the OWA-model is O(t^u), while t is the number of tasks and u is the number of resources per task. So, it is not very feasible for large allocation problems.

Team Faultlines [21]. The last approach focuses on team faultlines, which are dividing lines between a team of human resources (1-to-many allocation) that split them into two or more subgroups, which are formed by several individual characteristics. The assumption of the authors is, that humans in these subgroups have more collaboration, which leads to a higher process performance. So, the approach considers as *Allocation Criteria* several human characteristics from two perspectives. First, the demographic perspective related to human's personal background information, which includes such characteristics as gender, age, education, experience, etc. And second, the business process perspective, which are characteristics mined from process data e.g. activity frequency, business quality, interaction, etc. To select and weight key characteristics from these categories, information value is used. After that, the team faultlines are identified (based on the DBSCAN-clustering algorithm) and measured by a from the author's improved algorithm. Thereby, the allocation is done before the process execution, which makes it to another upfront-planning technique.

So from the *Model Input* of the resource data and event-logs, the approach creates as *Model Output* a resource allocation model, which consists of a performance prediction model (Fig. 3.2), that is generated by multi-layer perceptron and the allocation model based on the neural network. Thereby, the allocation model provides an ordered allocation list of all resource teams according to their performances. One major disadvantage of this approach is, that it could be problematic for a company to get all the demographic information with regard to data privacy of each human resource.

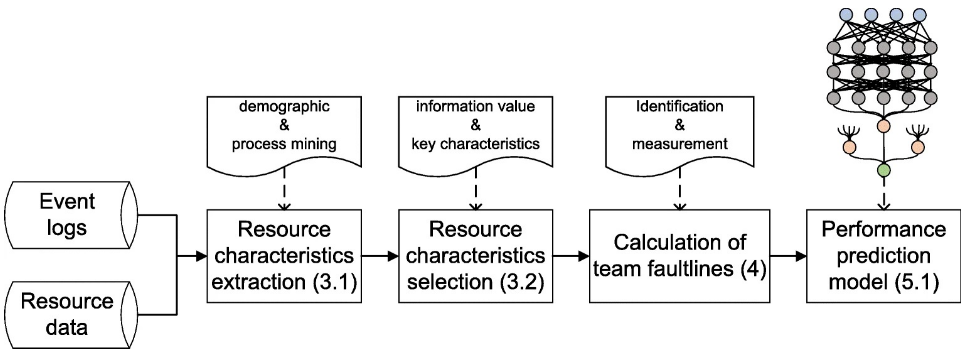

Figure 3.2: Performance prediction model of the Team Faultlines technique [21].

Technique	Allocation time	Optimization goal	Allocation type	Model Input	Model Output
Human Resource Allocation based on Process Mining	Upfront planning	Minimize cycle time	1-to-1	Event-log	Resource allocation model
Answer Set Programming (ASP) + Dependencies	Upfront planning	Minimize cycle time	1-to-1	Resource ontology	Set with potential resources
Task Operation Model (TOM) + Ant Colony Optimization (ACO)	Upfront planning	Minimize process cost	1-to-1	Business process model	TOM / optimal task operation path
Optimizing Resource Allocation by Business Process Improvement	Upfront planning	Minimize process cost	1-to-1	Role-based business process model	Allocation table / new process structure
Naive Bayes (NBSR)	Dynamic	Minimize cycle time	1-to-1	Naive-Bayes-Model	Appropriate human resource
Resource Allocation by Minimizing Cycle Time	Dynamic	Minimize cycle time	many-to-many	Business process model	Task assignment probabilities table
Particel Swarm Optimization (PSO)	Dynamic	Minimize cycle time	1-to-1	Resource allocation model	Optimal solution of resource scheduling
Reinforcement Learning (RLRAM)	Dynamic	Minimize process cost	1-to-1	Markov-Decision-Process-Model	Ordered queue of work items for each resource
Optimizing Cost and Maximum Throughput	Dynamic	Minimize process cost	many-to-many	Statistical data	Optimal resource allocation solution
Maximizing Compatibility	Upfront planning	Team allocation	1-to-many	Compatibility-Matrix	OWA-Model
Team Faultlines	Upfront planning	Team allocation	1-to-many	Resource data / event-log	Performance prediction model / Resource allocation model

Table 3.2: Comparison of all techniques.

Chapter 4

Experimental Evaluation and Comparison

In this chapter, the paper provides an evaluation and comparison of three different Human Resource Allocation techniques with real-world data. As mentioned in 1.2 the most of the described papers from the chapter above do not offer an open accessible prototype of their provided allocation technique. Furthermore, even the pseudocode in these papers is often incomplete or represent only some fragments of it, which makes it very difficult to implement these techniques to perform a fair comparison of them. So, in this chapter the evaluation and comparison is done of two approaches which are based on the first and second winner of the Business Process Optimization Competition (BPOC) in 2022 and a random allocation approach.

First, the settings of the evaluation experiments regarding the used data and the three evaluated approaches are described. Then, the execution of the experiments are recorded in section 4.2. Finally, all described approaches are evaluated and compared regarding their results in the experiments.

4.1 Settings

The approaches which are evaluated in this chapter, are developed at the Business Process Optimization Competition (BPOC) which is a competition from the '1st International Workshop on Data-Driven Business Process Optimization' which took place at the '20th Conference on Business Process Management' in Münster, Germany on 12th September 2022. In this competition, the participants had to develop a planning engine that assigns people to tasks such that the total cycle time of the cases is minimized. The developed technique should be applied to the data-log of the Business Process Intelligence Challenge (BPI) 2017, which represents a loan application process case of a Dutch financial service institute.

The event-log of the BPI-Challenge 2017 contains 1,202,267 events pertaining to 31,509 loan applications and for these applications a total of 42,995 offers were created. There exist three different types of events: application state changes, offer state changes, and workflow events. The company has 149 possible resources that can be allocated to the different tasks and for each event, the resource who caused this event, the event timestamp and lifecycle information are recorded. For all applications, the following data are available:

- Requested load amount (in Euro)
- Application type
- Reason the loan was applied for (LoanGoal)
- Application ID

And for all offers, the following data is existing:

- Offer ID,
- The offered amount,
- Initial withdrawal amount,
- Number of payback terms agreed to,
- Monthly costs,
- Creditscore of the customer,
- Employee who created the offer,
- Whether the offer was selected, and
- Whether the offer was accepted by the customer

Tasks must be executed by resources that are shared between all cases and all tasks in the process, while a resource can only work on a single task at a time. Task types have a resource pool, which indicates all resources that can perform tasks of that specific type. A process model of the event-log data, that shows how customer cases passed through the process in a particular simulation run, can be seen in Fig. 4.1.

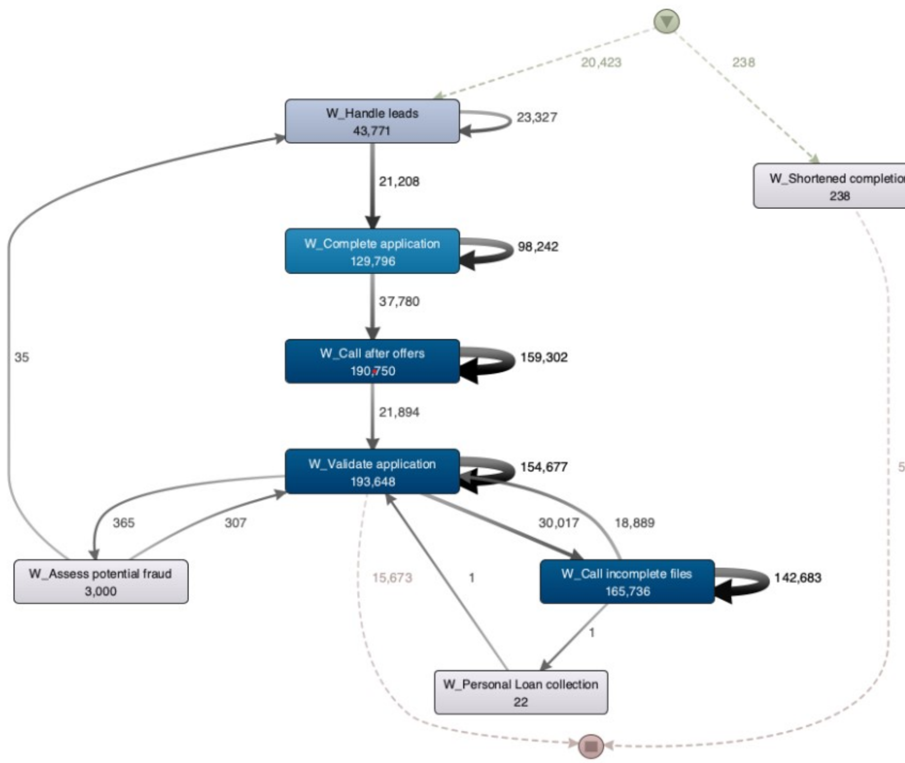

Figure 4.1: The process model of the BPO-Challenge event-log.

The cycle time of a case that should be minimized in the BPOC is defined as the time between the moment at which the case arrives and the moment at which it completes. Therefore, the cycle time includes the time, resources take to execute tasks for the case and the time the case has to wait for a resource to become available.

For the simulation of the process case presented above, exists two files: The first one provides the class *MyPlanner* in which the allocation planning is implemented in the *plan(available_resources, unassigned_tasks, resource_pool)* function. Moreover, in this file the simulation can be tested. The second file *simulator.py* provides the simulator with the help of them the simulation can be done. The simulation represents a full year of simulated customer cases and uses as input data a pickle-file. For the BPOC there are two different datasets available: *BPI Challenge 2017 - instance.pickle* and *BPI Challenge 2017 - instance 2.pickle*. By default, the simulator always uses the same dataset, but it can be changed to the other dataset by changing the filename in line 277 of the *simulator.py* file.

Before the detailed experiment proceedings are described, the three different approaches for human resource allocation that have to be evaluated will be demonstrated. The first technique is the approach of the BPOC 2022 winner: Eliran Sherzer from the University of Toronto, Canada. The second technique to be evaluated is the approach of the second winning team of the BPOC 2022: Peter Fettke and Alexander Rombach from the DFKI and Saarland University, Germany. The last technique for the evaluation is a simple random allocation method.

4.1.1 Eliran Sherzer's Approach

The main idea of this technique is to rate each possible (task, resource)-tuple and assign them accordingly to the rated score that consists of five factors:

1. Mean service time of (task, resource): given from previous service time of the specific tuple
2. Service time variance of (task, resource)
3. Resource ranking of all available resources: ranked by average service time from event-log history
4. Task ranking of all available tasks: ranked by average service time from event-log history
5. Probability of cycle completion: estimated by previous event-log history

After these five factors are determined, they are added up together with different weights to gain the total score. These weights were calculated by E. Sherzer beforehand by using different machine learning techniques to find the on average best performing values for the provided event-log data.

Regarding the provided human resource allocation problem categories of section 3.2 this approach can be classified as a dynamic technique to minimize the process total cycle time, because it does the calculation of the total score for each resource after each event during the simulation.

4.1.2 Technique of Peter Fettke and Alexander Rombach

The authors of this scheduling technique try to allocate the resources to a specific task by analyzing the previous performance of that (resource, task)-tuple regarding their execution times of historic events from the event-log. So, on the one hand, they sort a resource-list of all available resources regarding the mean execution time of each resource for a specific task-type. And on the other hand, this technique sorts a task-list regarding how many resources of the available resources are able to perform a task of a specific task-type. According to this, the resources are now prioritized by their historic execution time and the tasks are prioritized by how many resources can perform each task. Finally, in the allocation planner, the highest prioritized resource is assigned to the highest prioritized task for each event.

This technique can also be categorized as a dynamic approach to minimize the process total cycle time, because it does the prioritization of resources and tasks after each executed event during the simulation.

4.1.3 Random Allocation

The last allocation algorithm is a simple random allocation approach. For each unassigned task, it tries to select a random resource from the available resource-list. If this resource can perform the specific task type of the dedicated task, it is assigned to that task. The pseudocode of the random allocation approach is depicted in algorithm 2 with *aR* = the available resources, *uT* = the unassigned tasks and *rPool* = the resource pool of all task types assigned to the resource-list.

Algorithm 2

RANDOMPLAN(aR, uT, $rPool$)

```
assignments ← []
for task in uT do
    for resource in aR do
        resource ← SELECTRANDOM(aR) ▷ select random resource from available resources
        if resource in rPool then
            REMOVERESOURCE(resource, aR) ▷ remove resource from available resources
            APPENDRESOURCE(task, resource, assignments) ▷ append task, resource tuple to assignments
            break
        end if
    end for
end for
return assignments
```

4.2 Experiments

The three described approaches above are evaluated as a result of the following experiment: First, the initial event-log from the BPI-Challenge 2017 data is analyzed regarding the average cycle time per case via the simulator "Disco", which is a process mining tool that helps to visualize and analyze process data. The tool identifies an average cycle time per case of 21.9 days without any resource allocation optimization. After that, all three algorithms were implemented and tested with the BPI-Challenge 2017 data, which is represented in the two different pickle-files. All the experiments are implemented in Python on a personal computer with 1.50GHz CPU and 16GB RAM.

The simulation provides different results for each run, even when the data input is the same. This is a result of some non-deterministic variables in the simulator-file. Important variables like the inter-arrival rate between two tasks are initialized randomly to represent a more realistic simulation. Addressing this issue, for each instance-file, all approaches are executed five times to get a meaningful result. For each run, the average cycle time per case is depicted. Furthermore, for all five runs per instance the average cycle time per case of all runs, the variance, standard deviation and the average runtime of all runs are analyzed. The gathered results can be seen in Table 4.1.

	P. Fettke / A. Rombach		E. Sherzer		Random allocation	
Instance	1	2	1	2	1	2
# Runs	5	5	5	5	5	5
Run 1	13.166d	13.191d	8.539d	9.373d	19.004d	17.975d
Run 2	13.473d	13.040d	8.945d	8.993d	23.411d	22.109d
Run 3	14.138d	14.176d	9.249d	9.350d	27.562d	24.076d
Run 4	14.780d	12.191d	9.008d	8.837d	22.758d	30.396d
Run 5	15.292d	14.778d	9.310d	9.089d	30.969d	26.779d
Avg. Cycle Time	**14.170d**	**13.475d**	**9.010d**	**9.123d**	**24.741d**	**24.267d**
Variance	0.625	0.821	0.093	0.053	17.073	17.630
Std. Deviation	0.791d	0.906d	0.305d	0.231d	4.132d	4.199d
Avg. Runtime	44s	46s	10469s	10447s	7s	8s

Table 4.1: Results of the Experiments.

4.3 Results

After the experiments are executed successful, the results have to be evaluated and compared within the three provided allocation techniques. The main aim of these experiments and the BPO Challenge 2022 was to find out, which approach has the lowest average cycle time per case, respectively which approach has made the best improvement compared to the initial event-log. Here, you can say that the scheduling planner developed by E. Sherzer is by far the best performing approach. In comparison with the initial average cycle time of 21.9 days, the cycle time of this approach is by average less than half as much as the initial event-log, with only a mean of 9.067 days for both instance files. The cycle time of the technique provided by P. Fettke and A. Rombach is the next best approach, with an average cycle time per case for both data-sets of 13.822 days. Even this time is more than 4 days longer as the best technique, it improves the cycle time of the initial event-log by 8 days. The least performing approach is the Random allocation technique, it has an average cycle time per case of 24.504 for both instances, which is even longer than the initial resource allocation plan.

The next interesting result of the experiments are the values of the variance and the standard deviation of each technique. These values give an expression of how concise the approaches are regarding different data inputs. Here you can see, that the technique from E. Sherzer is again the approach with the lowest fluctuation, because all gathered cycle times stay close to the mean and results in a variance under 0.01 and a maximum standard deviation of 0.305 days (7.2 hours) among the five simulated runs. Also, the technique of P. Fettke and A. Rombach has a only small fluctuation in average cycle time per case, even though it is quite more than the previous approach. The experiments provide hereby a maximum variance of 0.821 and a standard deviation under 24 hours. Conspicuous is the high fluctuation of the random allocation approach, which indicates a maximum variance of 17.630 and a huge standard deviation of more than 4 days. This is due to the fact, that the allocation is done randomly, which can lead to very different cycle times (the experiments shows a range between 19 and nearly 31 days).

Another important result, is the different average runtime for all three evaluated techniques. This value indicates how long on average did the simulation including the allocation planning take time for all five runs. Analyzing this value the Random allocation approach is the fastest, given that it do not need much time for the allocation planning, because the technique does no comparison between the available resources or tasks like the other two approaches. This results in a mean runtime for both instances of 7.5 seconds. The second-fastest approach is the technique from P. Fettke and A. Rombach. It has an average runtime for both instances of 45 seconds, which is a result of the computation of the prioritization-lists after each executed event. Finally, E. Sherzer's technique is by far the worst regarding the average runtime. Due to the fact, that this technique needs to calculate five factors for creating a resource ranking, it needs much more time for the allocation planning than the two previous approaches. It has a mean runtime for both instance files of 10458 seconds, which are 174 minutes (nearly 3 hours) for one simulation run.

Additionally, you can say that all three approaches provide similar results in terms of the average cycle time per case, the variance and standard deviation and also the average runtime for the two different instance-files, that provide two different data-logs. So, that shows that the results gained in this experiment does not depend on one specific input data.

To sum up, you can say that the technique provided by E. Sherzer is the best performing approach regarding the average cycle time per case, but it has issues with the runtime of the allocation planning. So, it might be not suitable for very large input data-logs that contains much more data than the 1,202,267 events of the BPI Challenge 2017 event-log, but for small- or medium-sized input data it is the best provided solution for human resource allocation. If you need to allocate resources for an event-log with a huge amount of process-data, then the technique of P. Fettke and A. Rombach will be the best possible solution, because although it has a longer average cycle time than E. Sherzer's technique, it only needs 0.43% of the time for the allocation planning. Finally, in view of the performance of the Random allocation approach, which was even worse than the initial event-log, you can see that an efficient resource allocation is very important and leads to an enormous better process performance.

Chapter 5

Conclusion

In the final chapter, the key statements of this paper are summarized and a foresight for future work regarding human resource allocation in business process management is given.

5.1 Summary

This paper presents an overview over several human resource allocation techniques for BPM. At first, a look at the related work regarding this topic was taken and the theoretical framework in terms of resource allocation in general and allocation criteria was described. By an intensive literature review over the current academic literature of this topic, a bunch of eleven different allocation techniques was selected. Then, the gathered approaches were categorized into five problem categories regarding their time of allocation and their main optimization goal. After that, this paper analyzed the methodology of the provided approaches by their model input, allocation type and model output and has given a good comparison over all techniques. Finally, three different dynamic resource allocation techniques were evaluated in an experiment against each other with event-log-data from a real world scenario of a financial service institute. They were compared in terms of the average cycle time per case and the execution time it takes to perform the resource allocation. From the results of the experiment, a recommendation for all three techniques was given.

To sum up, you can say that human resource allocation is a huge and quite up-to-date topic in the scope of business process management and the domain of business process optimization. There exist various literature with new approaches or improvements of common techniques for nearly all problem categories. The provided allocation techniques can be differentiated in dynamic or upfront-planning approaches, the most of them can be categorized as 1-to-1 allocation types, but there are also a few techniques which focus on a 1-to-many team allocation. Nearly all approaches need a business process model or an event-log as input data, and provides different resource allocation models or recommendations as output.

The difficulty at the moment is to find the right allocation approach that satisfies the requirements and constraints for the business process that should be optimized. Therefore, this paper provides a good overview including a comparison of eleven techniques from academic literature and presents some experiments with three fully developed approaches. These experiments show that an efficient resource allocation technique is able to lower the average cycle time by more than the half of the initial cycle time without resource allocation. Otherwise, it also demonstrates that a complex and challenging approach has a high runtime when the used event-log contains a huge amount of process data.

5.2 Future Work

In recent literature exist plenty of resource allocation approaches, but several open research topics still exist. Various resource allocation techniques provide various different kinds of business process models or different kinds of input data for their approaches. Therefore, it would be good to have a standardized business process model, which can be used by a wide range of different allocation techniques. This would simplify the use of these techniques for different process cases within an organization.

A large problem, especially for the comparison of the human resource allocation techniques from the literature, is that the most papers do not provide a functional, fully developed and open accessible prototype of their work. So, for future work it will be important, that the authors of the provided papers make their code available for other authors, to allow a better evaluation of their techniques with real-world data. A good example of such a paper is [20] by K. Żbikowski et al., here they provide a resource allocation technique based on double deep reinforcement learning which allows doing the allocation for multiple processes and resources. The code of their prototype is accessible via the online code repository 'GitHub'. Furthermore, would this make it easier for companies to select the most appropriate approach for their business processes and to implement these techniques into their Business Process Management System.

Bibliography

[1] Michael Arias, Jorge Munoz-Gama, and Marcos Sepúlveda. Towards a taxonomy of human resource allocation criteria. In *International Conference on Business Process Management*, pages 475–483. Springer, 2017.

[2] Michael Arias, Eric Rojas, Jorge Munoz-Gama, and Marcos Sepúlveda. A framework for recommending resource allocation based on process mining. In *International Conference on Business Process Management*, pages 458–470. Springer, 2016.

[3] Michael Arias, Rodrigo Saavedra, Maira R Marques, Jorge Munoz-Gama, and Marcos Sepúlveda. Human resource allocation in business process management and process mining: A systematic mapping study. *Management Decision*, 2018.

[4] Giray Havur, Cristina Cabanillas, Jan Mendling, and Axel Polleres. Resource allocation with dependencies in business process management systems. In *International Conference on Business Process Management*, pages 3–19. Springer, 2016.

[5] Zhengxing Huang, Xudong Lu, and Huilong Duan. Mining association rules to support resource allocation in business process management. *Expert Systems with Applications*, 38(8):9483–9490, 2011.

[6] Zhengxing Huang, Xudong Lu, and Huilong Duan. A task operation model for resource allocation optimization in business process management. *IEEE Transactions on Systems, man, and cybernetics-part a: systems and humans*, 42(5):1256–1270, 2012.

[7] Zhengxing Huang, Wil MP 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.

[8] Akhil Kumar, Remco Dijkman, and Minseok Song. Optimal resource assignment in workflows for maximizing cooperation. In *Business process management*, pages 235–250. Springer, 2013.

[9] Linh Thao Ly, Stefanie Rinderle, Peter Dadam, and Manfred Reichert. Mining staff assignment rules from event-based data. In *International Conference on Business Process Management*, pages 177–190. Springer, Berlin, Heidelberg, 2006.

[10] Florian Niedermann, Sylvia Radeschütz, and Bernhard Mitschang. Business process optimization using formalized optimization patterns. In *International Conference on Business Information Systems*, pages 123–135. Springer, 2011.

[11] Luise Pufahl, Sven Ihde, Fabian Stiehle, Mathias Weske, and Ingo Weber. Automatic resource allocation in business processes: A systematic literature survey. 2021.

[12] Chathurangi Shyalika, Thushari Silva, and Asoka Karunananda. Reinforcement learning in dynamic task scheduling: A review. *SN Computer Science*, 1(6):1–17, 2020.

[13] Wil van der Aalst. Data science in action. In *Process mining*, pages 3–23. Springer, 2016.

[14] Arif Wibisono, Amna Shifia Nisafani, Hyerim Bae, and You-Jin Park. On-the-fly performance-aware human resource allocation in the business process management systems environment using naïve bayes. In *Asia-Pacific Conference on Business Process Management*, pages 70–80. Springer, 2015.

[15] Arif Wibisono, Amna Shifia Nisafani, Hyerim Bae Hyerim Bae, and You-Jin Park You-Jin Park. A dynamic and human-centric resource allocation for managing business process execution. *International Journal of Industrial Engineering: Theory, Applications and Practice, 23*, 2016.

[16] Yi Xie, Shitao Chen, Qianyun Ni, and Hanqing Wu. Integration of resource allocation and task assignment for optimizing the cost and maximum throughput of business processes. *Journal of Intelligent Manufacturing*, 30(3):1351–1369, 2019.

[17] Yi Xie, Chen-Fu Chien, and Ren-Zhong Tang. A dynamic task assignment approach based on individual worklists for minimizing the cycle time of business processes. *Computers & Industrial Engineering*, 99:401–414, 2016.

[18] Jiajie Xu, Chengfei Liu, and Xiaohui Zhao. Resource allocation vs. business process improvement: How they impact on each other. In *International Conference on Business Process Management*, pages 228–243. Springer, 2008.

[19] Mehdi Yaghoubi and Morteza Zahedi. Resource allocation using task similarity distance in business process management systems. In *2016 2nd International Conference of Signal Processing and Intelligent Systems (ICSPIS)*, pages 1–5. IEEE, 2016.

[20] Kamil Żbikowski, Michał Ostapowicz, and Piotr Gawrysiak. Deep reinforcement learning for resource allocation in business processes. 2021.

[21] Weidong Zhao, Shi Pu, and Danni Jiang. A human resource allocation method for business processes using team faultlines. *Applied Intelligence*, 50(9):2887–2900, 2020.

[22] Weidong Zhao, Liu Yang, Haitao Liu, and Ran Wu. The optimization of resource allocation based on process mining. In *International Conference on Intelligent Computing*, pages 341–353. Springer, 2015.

[23] Weidong Zhao, Qingfeng Zeng, Guangjian Zheng, and Liu Yang. The resource allocation model for multi-process instances based on particle swarm optimization. *Information Systems Frontiers*, 19(5):1057–1066, 2017.

Appendix A

Program Code / Resources

The source code of the three approaches from the experiments, a PDF version of this thesis, a documentation of the BPO Challenge 2022 and the data from the BPI-Challenge 2017 are available at the online GitHub-repository:
https://github.com/Marco214/Bachelor-Thesis

More information about the BPO-Challenge 2022 from chapter 4 can be found at:
https://sites.google.com/view/bpo2022/competition

The process mining tool “Disco” from section 4.2 is accessible via:
https://fluxicon.com/disco/

Ehrenwörtliche Erklärung

Ich versichere, dass ich die beiliegende Bachelorarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

M. Schulz

Haßloch, den 23.12.2022 Unterschrift