Quantity meets quality for mined process models through simulated events

Simon D. Arends
University of Twente
PO Box 217, 7500 AE Enschede
the Netherlands

ABSTRACT

Real-life systems and simulated models often generate event logs of their processes. These event logs can be mined to create process models that represent and visualize what happened throughout the process. Simulation models are particularly useful since they mimic the real world, with the benefit that it is possible to examine many different variations. To conduct simulation experiments, such models often have to run for a certain amount of time to generate event logs. This research investigates what the effect is of the number of event logs generated by such a simulation model, on the quality of the resulting mined process models. The research partitions a data set into subsets that simulate an increasing quantity of event logs. These subsets are then investigated using performance and conformance analysis. It is found that as the number of event logs increases, the analysis output becomes more reliable and the performance and conformance values stabilize and converge to certain values. A benefit of the approach is that it can lead to a method to assess whether a sufficient amount of event logs is generated, to mine representative process models. This can lead to saving time and more reliable results.

Keywords

Process Mining, Event Logs, Simulation, Logistics

1. INTRODUCTION

Nowadays, many businesses and organizations have moved into the age of digitalization by adopting Enterprise Resource Planning (ERP) systems and other workflow management systems like Customer Relationship Management (CRM) systems or Supply Chain Management (SCM) systems [1]. These systems are all in one way or another designed to support organizations with their different processes. This can be from registering a patient in a hospital ERP system to a new potential sales lead to a CRM system. These systems log their events, leaving traces of the carried out activities. Furthermore, now with the rise of Industry 4.0 strongly accompanied by the use of Internet-of-Things (IoT), more data is becoming available than ever before [19], [24]. Whenever someone or something is processed by a system, a trail of data is left behind. This trail can be visualized in a process model, showing routes of activities possible in the process. The trail can contain valuable information about the behavior of a process and can, therefore, be studied. This trail is also known as a trace, consisting of event logs. An event often contains a case ID, an event ID, a timestamp, an activity, a resource, and a transaction type [4]. There are however many different variations possible, see Table 1.

Table 1 Event log example inspired by experiment 512

ID	Timestamp	Product	Type	Event	Status
21	00:02:28	.MUs.Robot	Robot	Sawing	Start
22	00:02:28	.MUs.Robot	Robot	Sawing	waiting

A way of studying these traces is by using process mining. Process mining provides the bridge between data mining or machine learning techniques and the business process management (BPM) discipline [12]. The idea of process mining is to discover, monitor, and improve real processes by extracting knowledge from event logs readily in today's systems [4]. Thus, optimizing processes by analyzing where the process can be improved. This allows for the development of simulation models based on realistic representations of processes as logged by the underlying system logs [21]. The purpose of simulation models is to make it possible to experiment with settings of systems and monitor the consequences of changing variables. Think, for example, of the impact of using fewer resources to run the same amount of activities. These simulations require time. A simulation model consists of different situations, with different variables. For simulated scenarios, a process model can be mined based on event logs. The number of event logs can change the mined process model. Think for example of two cases: (i) a situation with 10 generated event logs and (ii) the same situation where 10.000 event logs are generated. Process models can be mined for both cases, but the behavior of (ii) will most likely be different from the behavior (i). This is because the mined process model of the second case uses more event logs and is thus able to create a more complete and reliable process model. However, at what point do you have enough event logs so that you can say with confidence that the mined process model is complete and reliable. This research will investigate this effect of quantity and quality.

2. BACKGROUND

2.1 Process discovery

Many business processes are created using a top-down manner, where the management uses BPMN (Business Process Model and Notation). It is expected that these models are followed in the real world when executing processes. Enacting the behavior of a model, is called Play-out. Play-out generates traces as an output from the model. When creating a model based on reality and its event log, one is talking about Play-in. Play-in uses an event log as an input. The final form is Replay. Replay uses an event log and a process model as input. The event log is 'replayed' on top of the process model [4]. Thus, replay can be used for performance and conformance analysis, as done in [10].

33rdTwente Student Conference on IT, July 3rd, 2020, Enschede, The Netherlands. Copyright 2020, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.

2.2 Conformance checking

When assessing the quality of (mined process) models, often one looks at four criteria: fitness, simplicity, precision, and generalization [4], [3], [5], [4]. Discrepancies between the log and the mined process model can be detected and quantified by replaying the log on the model [4], this is called conformance checking [23].

2.2.1 Fitness

As defined by [23], fitness is the extent to which the log traces can be associated with valid execution paths specified by the process model. The fitness of a model is the fraction of traces in the log that can be fully replayed on the model. A fitness of 1 meaning that all traces can be played on the model and a fitness of 0 meaning the opposite.

2.2.2 Simplicity

For simplicity, the notion of Occam's Razor can be used [4], [5], [3], [23], meaning that the simplest model that can explain the behavior seen in the log, is the best model.

2.2.3 Precision

A model is precise if it does not allow for more behavior than the behavior from the event logs [4][8]. An example of this would be a 'flower' model. This model has a start, an end, and one place in the center. It is then possible to reach any number of activities from this central place. Using such a model it is possible to mimic any given event log, but this model also allows for a lot of other potential behavior. Such a model is called under-fitting'.

2.2.4 Generalization

A model should not restrict behavior to the examples seen in the log. This is when a model becomes too specific and would only be able to work for some specific event logs. One could imagine multiple parallel lanes of hardcoded sequences. Such a model is called 'over-fitting' [7], [3], [4].

2.3 Process Model Discovery

For the discovery of process models based on the event logs, different algorithms can be used. Below two algorithms are explained in short that are relevant for the research.

2.3.1 Alpha miner

The Alpha miner plugin uses the Alpha, or α-algorithm [6]. The Alpha miner is one of the simplest discovery algorithms and was proven to be correct for a clearly defined class of processes [3], [6]. The algorithm is based on 9 conditions and simply scans the event log for particular patters; e.g., a is followed by b, but b is never followed by a, this indicates is a causal relationship between a and b[3]. For more information on the alpha miner, see [6].

2.3.2 Inductive miner

The main benefits of the Inductive miner are that it returns sound models (no deadlocks or other anomalies), it handles infrequent behavior well and is relatively quick [18]. The way the Inductive miner can handle infrequent behavior is because by default it filters event logs on noise. It filters out approximately 20%, creating an 80% model using the Pareto principle. For more information on the Inductive miner, see [18].

3. RESEARCH QUESTIONS

The main research question is depicted below by **RQ**, the remaining questions are sub-questions meant for supporting the main research question. The questions **RQ 1.1** and **RQ 1.2** are focused on literature. The question **RQ 1.3** is meant to conceptualize the finding of the literature into an artifact (i.e. method or workflow). **RQ 1.4** and **RQ 1.5** attempt to give more insight into the main **RQ**.

RQ. How does the quantity of event logs generated with a simulation model affects the completeness and reliability of mined process models?

RQ 1.1: What is a representative mined process model according to process mining guidelines?

RQ 1.2: When is a simulation reliable according to simulation modeling guidelines?

RQ 1.3: How can we combine the previous two questions into an artifact (i.e., conceptual model, method, workflow)?

RQ 1.4: What is the effect of an increasing amount of event logs on the performance and conformance of mined process models?

RQ 1.5: To what extent do different process discovery algorithms affect the completeness and reliability of mined process models?

4. RELATED WORK

4.1 State of Art

The preceding work of this research includes [10]. In [10] a Discrete-Event Simulation (DES) model is created to study the behavior of Automated Guided Vehicles (AGVs) in a classic logistics problem. To study the behavior, a set of 27 different scenarios is created, each of which is unique. For these scenarios, a set of event logs is chosen beforehand. The mined process models resulted from the simulations are all analyzed on fitness, precision, and generalization. The simulations aimed to validate the proposed agent-based process mining architecture [10]. The research only considers different scenarios and their effect on fitness, precision, and generalization, while this research seems to neglect the effect of the number of event logs on the KPI's.

There exists already a field around process mining and simulation. In this field, a yearly conference is hosted called the Winter Simulation Conference (WSC). In [21], a simulation model is created based on observations from event logs. The paper does not focus on determining a number of event logs necessary to mine a representative process model. Similar holds for the work of [11], in which an iterative learning approach was used based on event logs and agent decisions. This leaves a knowledge gap, when does one know that enough event logs have been used and/ or generated.

Another paper talks about the simulation-optimization problem [20]. One might think that this is the same problem as we want to solve in this research paper. However, the simulation-optimization problem looks at optimizing the simulation models itself. It sees the simulation model as a 'black-box' problem and tries to 'solve'/ 'optimize' the simulation model based on the event logs. This paper does not state anything about the relation between the number of event logs of a simulation model and the quality of the mined process models.

Even the core books on process mining, do not mention a number of replications or event logs needed to generate a representative mined process model [3], [4].

4.2 Gap

As seen in Section 4.1, there is still a knowledge gap between simulation studies and process mining. It is currently unclear how many event logs are needed from a simulation model to converge to a representative mined process model. This is because the bridge between simulation and process mining is still relatively new [2]. This research will aim at finding the relation between the quantity of event logs and the quality of the mined process models from these event logs. To be able to generate event logs more efficiently and maximize utility by saving time and other resources.

4.3 Simulation stopping criterion

In the world of simulation, it is important to select an appropriate run length. This has to do with the fact that one needs to be able to estimate the model performance with sufficient accuracy [22] [16]. Commonly used methods for determining the run length make use of confidence intervals. A confidence interval is a statistical means for giving an estimated range within which the true mean average is expected to lie [22]. For the confidence intervals, often a range of 95% is chosen. Meaning, that one can say with a confidence of 95% that a value will lie within the given range. This translates into a confidence coefficient of 1.96 or a significance level (α) of 5%. The confidence interval uses the standard deviation, the mean value, the confidence coefficient (t-value with n-1 degree of freedom and significance of a/2), and the number of replications. The formula is given below:

$$CI = \overline{X} \pm 1.96\frac{s}{\sqrt{n}}$$

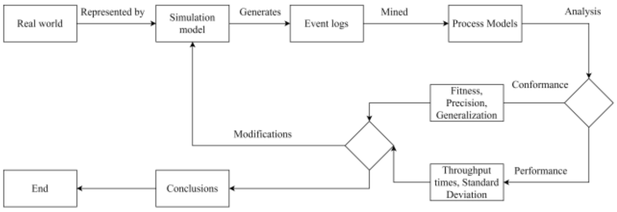
The right part of the formula is also called the 'margin of error', this is what mostly defines the interval. Since you add the margin of error to the mean or average to get the upper bound of the interval, and you subtract the margin of error from the average to get the lower bound. Thus, one would want a margin of error as small and close to zero as possible.

5. METHODOLOGY & APPROACH

5.1 Framework

When conducting process mining using event logs generated by simulation models, the framework depicted in Figure 1 can be used. A simulation model is created to represent the real world. This simulation model is used to generate event logs and these event logs are then mined for process models. This makes it possible to perform an analysis based on conformance and performance. When analyzing performance, one looks at the performance of traces and activities in the model. Common performance indicators are throughput times, which are used to see how long it takes for a trace to complete. When conducting conformance analysis, one looks at the event log and the model and see how well these two align. Once the analysis is completed, often modifications are made to the simulation model. This loop continues until satisfied and conclusions can be drawn. Note, that it is also possible to generate multiple sets of event logs before doing the analysis.

Figure 1 Simulation framework for Process mining



For the experiment, a similar structure can be applied. The goal of the experiment within this paper, is to see what the effect is of the increasing number of event logs for the same simulation run. A simulation is run beforehand, and the data is then partitioned into different subsets. These subsets are then mined for process models and analyzed.

5.2 The experiment use case

The data used in the experiment are event logs, simulated by the DES simulation model of [10]. The simulation model created is based on a typical logistics transportation problem, namely a job-shop problem [17]. A certain number of jobs need to be executed on a predefined number of machines. The model contains a starting point, drain point, and a track in the center. Aligned to the track are several machines. AGVs are used to drive over the track to the machines. These machines then execute their job and then the AGV continues to the next machine. The event logs produced by this simulation model are used for the experiments in this research. An example inspired by the event log of experiment 512 can be seen in Table 1. Three data sets have been chosen; 411, 512, and 613. The first number indicates the number of active AGVs on track, the second number indicates the driving direction, and the last number indicates how the AGVs were dispatched. This is done to be able to diversify the analysis, but still, attain the same complexity for the process models. Because if the driving direction would include backward options, then the models would become more complex, and direct comparison will not be valid.

5.3 Data input

The data used in the experiment originate from [10]. The data is generated based on a simulation model. The data contain different sets that match different situations for the simulation model. For the experiment, multiple sets are analyzed for what the effect is of the quantity of event logs on the quality of the mined process models and the information gathered for the process performance. When analyzing a situation, first the full event log is analyzed to estimate the Events per Trace (EpT). The event log is filtered on complete traces. Then, we divide the total amount of events in the filtered set by the number of complete traces to get an estimate of the EpT. The EpT is then used to partition the full event log in subsets. For the analysis, the complete traces are one of the important aspects. If a trace is incomplete, it should not be evaluated by the process mining algorithms. For the size of the partitions, the following distribution is chosen for the traces: 10, 20, 40, 100, 200, 500, 1000, and the full event log. This is done to check the possible differences in generated process models. Furthermore, the subset of 20 traces, also contains the first 10 traces. This is done by linear partitioning. This way the impact of the additional traces can be measured against the previous subset. Since it is not possible to select the exact number of traces, the EpT is used to estimate the number of events needed to get the desired number of traces. For example, experiment 411 has an average EpT of 35. Then for the first partition of 10 traces, the first 350 events will be selected from the complete event log. These 350 events are then filtered on complete traces. These complete traces will then be used for further analysis.

5.4 Data processing

The data is processed in the following steps:

1. A data set is selected
2. The data is partitioned into subsets
3. The subsets are loaded into Prom and filtered
4. Process mining discovery algorithms are applied to discover the underlying process models
5. The event logs are replayed over these process models
6. Performance and conformance analysis is conducted.

Once a set is selected, the data is converted into a CSV-file format. Then, the number of events is selected for each partition based on the EpT, as explained in the previous section. For the further processing of the data, ProM is used. ProM is used and accepted by the many researcher in the process mining domain. The different subsets are loaded into prom and all filtered on complete traces using the 'Simple Heuristics' plugin. Then, the subsets get mined by two different algorithms for mining process models. The Alpha miner and the inductive miner. Afterward, conformance and performance analysis are applied by replaying the subset's event log on the subset's process model, resulting in output data.

5.4.1 Prom Actions

For the different steps in ProM, different plugins are used. As mentioned before, the Simple Heuristics plugin is used to filter out incomplete traces.

5.4.1.1 Discovery plugins

The next step is to generate a process model. The two discovery algorithms used to generate the Petri nets (process models) are the Alpha miner and the Inductive miner, as explained in the subsections of section 2.4. Two algorithms were chosen to be able to show the effect of the quantity of the event logs on the quality of the process models, whilst staying independent of one algorithm. Both algorithms are left on the default settings. This is done because it is possible to fine-tune the process models, but this could potentially give an unfair representation. The only setting, which is changed, is the classifier, this is set to '(Event Name AND Lifecycle transition)'. This is solely for the structure of the data generated by [10].

5.4.1.2 Performance plugin

For the performance analysis, the plugin 'Replay a Log on Petri net for Performance/ Conformance analysis' by Arya Adriansyah, is used. This plugin uses both a previously generated Petri net and an event log as input. The plugin will then replay the event log over the Petri net and give statistics related to the performance and conformance. Using this plugin, it is possible to find the average throughput time, the minimum and maximum throughput time, the standard deviation, and the total observation period.

5.4.1.3 Conformance plugins

In section 2.2, there are four metrics defined, for this conformance analysis we will only be using fitness, precision, and generalization. This is because simplicity is often too abstract for this analysis. To be able to measure the conformance, two plugins are applied. First, the 'Replay a log on Petri net for Conformance Analysis' by Arya Adriansyah, is used. Since this plugin is required to prepare for the measuring of precision and generalization, this plugin is also used as a source for the fitness. The plugin uses an adaption of the A* algorithm. Full documentation on the method can be found in [7], [9]. All settings are left on default. Second, the previous plugin produces an alignment, of the Petri net and event log, as output. This can be used by the plugin 'Measure Precision/ Generalization' by Arya Adriansyah, to do what the name suggests.

5.5 Data output

The conformance analysis contains information on the fitness, precision, and generalization of the model, as explained in Section 2.3. These values will be different for the two algorithms and will give an interesting insight into how both algorithms behave and evolve over the number of event logs provided. The values will always be in the range 0-1, with 1 being best, and 0 worst. In an ideal situation, a mined process model has a value near 1 for all three metrics. The performance analysis results in the average throughput time for a trace, the minimum/ maximum time for a trace, the standard deviation in throughput times, and the total observation time. It is expected that the average throughput time will deviate between the first couple of subsets, but that it will stabilize at some point. This would also be represented by the standard deviation.

5.6 Analysis

Once the data is gathered for both algorithms per subset, the data should be merged into a single data set. This data set will then show the name of the subset and the number of traces used for the analysis followed by the different measurements for the conformance and performance. It is then possible to visualize the data by generating a line graph including all three conformance metrics (see Section 6 for results). It is expected that for the first couple of subsets the lines will fluctuate but after that will stabilize for the later subsets. For the throughput times, a line graph will also be generated to show the behavior of the throughput times over the increasing number of traces. Margins could be used to visualize the standard deviation for each average throughput time. Furthermore, the confidence intervals will be calculated for the different subsets as explained in Section 4.3.

6. RESULTS

6.1 Data

Table 2 shows an overview of the partitioning of the complete event log for experiment 411 of the data set from [10]. The steps described in Section 5.2 are followed here. First, the full event log (subset 8 in Table 2) is filtered on complete traces. The full event log started with 47,474 events and was filtered down to 44,880 events with 1296 complete traces. This results in an average EpT of 34.623, which is rounded for convenience to 35 Events per Trace. The next step was to select the according number of events for the number of traces, for the distribution from Section 5.2, these are the 'Events pre-filter'. Once the subsets contained the desired number of events, they could be filtered. This resulted in the 'Events post-filter' and the '# Traces'.

Subset	1	2	3	4	5	6	7	8
Events pre-filter	350	700	1400	3500	7000	17500	35000	47474
Expected # traces	10	20	40	100	200	500	1000	1450
Events post filter	144	528	1232	3088	6640	16552	33288	44880
# Traces	4	15	36	90	190	190	963	1296

Table 2 Partitioning for experiment 411

6.2 Analysis

For the analysis, the methods described in section 5.4.1 were applied. Analyzing the data gave some interesting insights based on performance and conformance.

6.2.1 Performance

The results showed some interesting behavior. As can be seen in Figure 2, experiment 512 and 613 are quite similar. Experiment 411 appears to differentiate from the other two experiments. This can be explained by looking at the maximum throughput times. Experiment 411 appears to suffer from a substantial increase in maximum throughput times compared to the other two experiments. A possible cause for this be a blockade in the simulation, which caused major delays. This then caused the maximum throughput times to increase, also affecting the average throughput time. This makes it challenging to draw conclusions based on the performance data of experiment 411. Furthermore, one can see that experiments 512 and 613 are stabilizing when it comes to their throughput times. After around 500 traces, the difference between average throughput times appears to decrease and the lines stabilize and converges towards a certain value. Since the research uses data based on simulation models, the warm-up time of these models should be considered. There are two key issues in assuring the accuracy of estimates of performance based on simulation models. The first is the removal of the initial bias (warm-up period) [22], the second is ensuring that enough data is generated to make accurate performance estimates [14]. If the data sets would be bigger, one could more easily tell if the model had to warm up, for the first 500 traces. What can be concluded is that the average throughput time stabilizes as the number of traces increases.

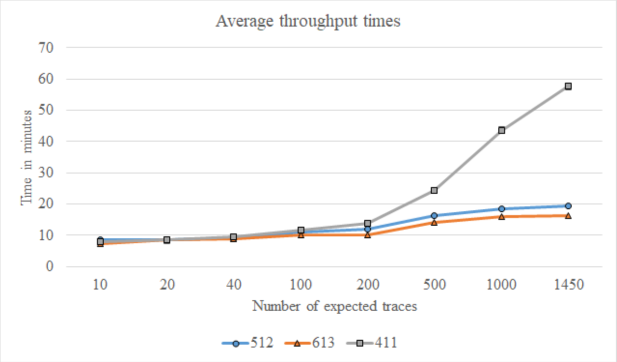


Figure 2 Average throughput time for three different experiments

The data was also analyzed using the confidence intervals, mentioned in Section 4.3. Figure 3 was inspired by the confidence intervals. The graph shows the margin of error, for the different number of subsets, with a confidence coefficient/ t-value of 1.96 (α=5%). What can be seen is that both experiments 512 and 613 stabilize and converge towards a certain margin of error (in this case a margin of error of ±0.2). This means that it can be said with a 95% confidence level, that any next value (throughput time) will lie within the interval of the mean. In the case of the experiment, the confidence interval after all traces is [19.57533, 19.08467]. What can be concluded is that the margin of error for the confidence of 95% on the average throughput time, converges as the number of traces increases.

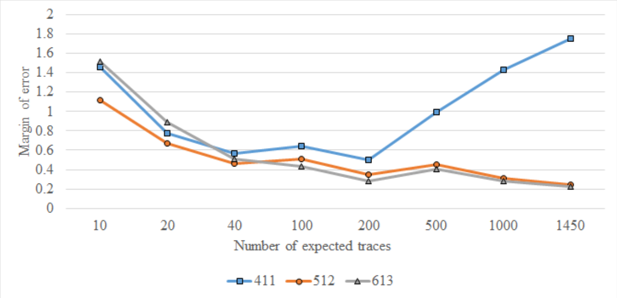


Figure 3 The margin of error for the confidence of 95% on the throughput times of the number of traces

6.2.2 *Conformance*

The conformance is analyzed using the three metrics mentioned before: fitness, generalization, and precision. The fitness is shown in Figure 4. When looking at fitness, one can see that fitness is not affected by the number of traces. The fitness depends mostly on the complexity of the underlying model (and its event logs) and the discovery algorithm. The algorithms by default attempt to get a good fitness score so that the model represents the logs well. This means that if the complexity of the logs remains the same, the fitness will not be considerably affected by the increase in the number of traces. Furthermore, both algorithms appear to perform well on the complexity level of the model, with the Inductive miner producing a slightly higher fitness.

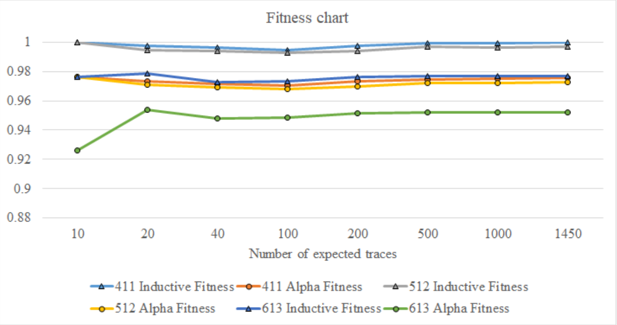


Figure 4 The fitness for the different experiments and the different discovery algorithms. The algorithms can be distinguished using different markers.

The generalization is shown in Figure 5 for the different experiments and the different algorithms. One of the first things that stand out is the fact that for all experiments and algorithms, the generalization seems to stabilize and converge towards 1. Both algorithms appear to behave quite similar to the increase in the number of traces and generalization. One important observation here is that for this simulation model and level of complexity, the generalization stabilizes after approximately 100 traces.

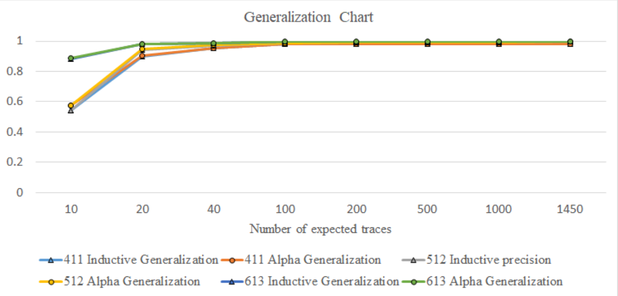


Figure 5 The generalization for the different experiments and the different discovery algorithms. The algorithms can be distinguished using different markers.

The precision is depicted below in Figure 6. The precision is the only metric, out of the three conformance metrics, which appears to show a unique distinction between the Alpha miner and the Inductive miner. This distinction can also be seen when comparing the actual models. The Inductive miner is able recognize the different patterns and considers that the order of some activities can be different. Whereas the Alpha miner tries to create more routes between the different activities for specific combinations. Thus, causing a less precise model. This is also clearly visualized in Figure 6, where the different algorithms are distinguished by their markers. One can see that the Inductive miner performs well on precision and scores around a 0.8, the Alpha miner does not perform well and scores around a 0.2. Furthermore, one can see that both algorithms need a certain number of traces to converge towards their values. Note, line '411 Inductive Precision' lies directly under line '512 Inductive Precision'. The Inductive miner appears to be quicker to converge towards its value than the alpha miner. The Alpha miner needed an approximate 100 traces to stabilize and converge.

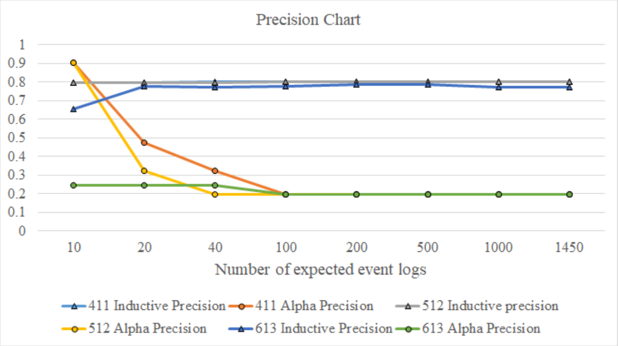


Figure 6 The precision for the different experiments and the different discovery algorithms. The algorithms can be distinguished using different markers.

Finally, Figure 7 shows some of the performance and conformance indicators and metrics for experiment 512. The conformance metrics use the primary vertical axis (left), and the performance indicators (average throughput time and standard deviation) the secondary vertical axis (right). One can see that all conformance metrics are stabilized after approximately 40 and 100 traces. The performance indicator shows that the average throughput time is stabilizing, but it cannot be said (yet) whether it reached its final state. The standard deviation is also visualized using the error bars and can be seen to increase over time but are stabilized in the end.

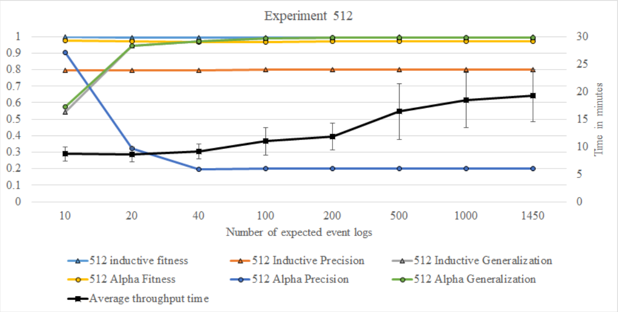


Figure 7 Performance and Conformance of experiment 512

7. DISCUSSION

The results from the previous section do show potential. This research on analyzing the effect of the quantity of event logs on the quality of the mined process models. For the research, the event logs have been translated into (complete) traces, since traces add more value than single event log lines. A trace is a complete case of a resource going through the process from start to end. A single event log would not contribute anything in the sense of an average throughput time or its effect on the quality of the models that will be generated. The results do show that the values converge towards certain values It does visualize that process models are also affected by some form of a warm-up period, as mentioned in the simulation literature [16], [22], [14], [4]. Furthermore, the research does contain some limitations. One of the important limitations was time, which affected all other limitations.

7.1 Limitations

7.1.1 The data set

First, the original data set used for the experiments, based on the simulation model, is not large. This results in a limiting number of traces. This can be seen for the average throughput times since they appear to converge, but no conclusions can be drawn as of now. Second, the number of analyzed sets per experiment is limited. This mostly has to do with time limitations. The experiments can be split up in more subsets to give a better representation of the effect of the increase in the number of event logs. Ideally, the analysis could be conducted for every new trace added to the subset. Finally, the case on which the simulation model and the event logs are based is relatively simple. This caused the discovery algorithms to be able to converge towards their final values quite quickly for the conformance metrics. If the models would have been more complex, it might have been possible to see more of an evolution for the metrics.

7.1.2 Methods

In the end, all the data processing and analysis were done manually. In the earlier stages, there was an emphasis on trying to automate the data processing processes. This would potentially have yielded in more subsets per experiment and more algorithms and thus a more comprehensive analysis. However, due to the limited time and risk-averse behavior, the choice was made to manually conduct the processes. The focus for processing the experiments was first on using the RapidMiner extension RapidProm. However, in the end, this did not meet the quality and stability requirements for the limited time left. In hindsight, another option would have been to use Robotic Process Automation (RPA) to perform the repetitive tasks conducted now manually in Prom [13]. This has the potential to increase the quality of efficiency of any future research.

7.1.3 Conformance analysis

The research only uses a single method for each conformance metric. Thus, is fitness only measured in one way, described in Section 5.4.1.3, but can be measured in different ways. It might be interesting to see what the effect of this would be and whether the fitness would then be affected by the number of traces. Furthermore, an attempt was done to replay the full original event log over the Petri nets generated by the different subsets, to see how well they align. However, this did not add any value since the fitness was always close to 1, because of the potentially low level of complexity.

7.1.4 Statistical analysis

Since the number of subsets was limited, it was challenging to conduct a comprehensive statistical analysis. Ideally, one would want to analyze the effect of an extra trace on for example the throughput time. It would then be possible to calculate the new mean for the increased data set. This would then also allow for an analysis of the var, t-value, and margin of error. This makes it possible to tell when the throughput time is stabilized enough. One could then also see whether the conformance metrics are fully converged.

7.1.5 Simulation models

Since simulation models can be complex, they are prone to possible errors and bottlenecks. When a task gets executed in real life and an anomaly occurs, a human worker can respond by stopping or restarting the process. In a simulation model, this is only possible when specific conditions are set. This is what creates a challenge for analyzing the performance of a simulation model. One would need to have to ability to filter out a specific trace/ outlier so that it does not affect the rest of the results. This is potentially what happened to one for one of the experiments and this affected the confidence of the analysis. Furthermore, simulations suffer from something called a ‘warm-up period’, or an initialization bias. This is the period of a simulation where everything still needs to get up to speed. Warm-up periods can vary in size and effect. If the data set would have been larger, it would have been easier to account for such a warm-up period.

7.1.6 Pre knowledge

Finally, this was a new topic for the researcher. This meant that the researcher first needed to learn as much as possible about the domain of process mining in the first weeks of conducting the research. This affected the amount of time left for conducting experiments.

8. CONCLUSION

The purpose of the research was to study the behavior of process models as the number of event logs as input increased. This is done by partitioning an event log data set into subsets and then creating process models based on these subsets. These subsets were then analyzed based on performance and conformance. The results showed that the fitness scores were consistent over any number of event logs. The precision and generalization did need approximately between the 40 and 100 traces to converge. The average throughput times, the standard deviation, and the margin of error also stabilized over time. To come back to the research questions.

RQ 1.1: Mined process models are assessed based on their conformance scores, which strive towards a score of 1.0.
RQ 1.2: A well-known technique for assessing whether a reliable number of replications is reached, is by using confidence intervals.
RQ 1.3: RQ 1.1 and RQ 1.2 were combined for determining the methods and the workflow used to process and analyze the data and the subsets.
RQ 1.4: An increase in the number of event logs gave a stabilization of the performance and conformance
RQ 1.5: Different discovery algorithms deal with complexity and quantity in different ways, whereas some algorithms need more time to stabilize than others.

The quantity of event logs generated with a simulation model affects the completeness and reliability of the mined process model positively. The more the quantity increased, the more the scores stabilized, and the reliability and completeness increased. Further research is required, but there is definite potential. If one knows how many event logs need to be simulated beforehand, then this can save time and ensure reliability.

9. FUTURE WORK

The future work contains suggestions based on the current limitations of this research and new potential questioned that have arisen.

9.1 Increase quality

9.1.1 Number of subsets

The first possibility for future work is by conducting the same research as described in Section 5, but with more subsets. Ideally, create for every new trace a new process model using the discovery algorithms. If one per trace is not possible (potentially limited storage and/ or time), then simply try to create as many subsets as are possible for the given experiment. A method that can be used for this is potentially RPA, mentioned in Section 7.1.2. RPA has the potential to execute the processes more efficiently, reliably (no possible human error) and on a larger scale [13].

9.1.2 Discovery and analysis Algorithms

This research only used two discovery algorithms for the mining of the Petri nets/ process models. In this research the algorithms were left on the default settings, but it is also possible to finetune the algorithms to meet certain requirements. Furthermore, the quality of the research can benefit from a more in-depth analysis, by seeing how other algorithms deal with the increase in the number of traces for a specific experiment. Finally, different algorithms can also be applied for the conformance checking. Currently, it is only done using one method, Section 5.4.1, but this can be extended. Since fitness for example can be measured in many ways using different settings. It can be interesting to see what the effect is of these settings and different fitness algorithms on the fitness related to the number of traces.

9.1.3 Statistical analysis

The research can benefit from a more in-depth statistical analysis to increase quality. However, this is only possible if the number of subsets increases, as suggested in Section 9.1.1. One could then better analyze the difference of the mean for every new trace (or several traces depending on the subset), the variance, the t-value, the margin of error, and could test if the number of replications satisfies the conditions of the confidence interval. This number of replications could then also be analyzed using the conformance checking methods mentioned in this research to see whether they are fully converged. This could then potentially lead to finding the stopping criterion. Furthermore, an alternative means for determining the number of replications required can be found by rearranging the confidence interval, is suggested by [22]:

$$n = \left(\frac{100S * 1.96}{d\bar{X}}\right)^2$$

Where d is the percentage deviation of the confidence interval about the mean, and the 1.96 is again the confidence coefficient/ t-value (α=5%).

9.2 Bigger and more complex data set

Since the data set used in this research appears to be limited in its scope to observe a full stabilization for the throughput time, a bigger data set might be required. A bigger, and potentially real-life,data set could offer better insights into when the simulation for this simulation model was fully stabilized and could indicate a criterion for stopping a simulation run. If the complexity level would remain the same (based on the same simulation model) then at least a reduplication of the current size would be recommended (± 3,000 traces). Furthermore, if the data set would be more complex, then it might be more interesting to see how different discovery algorithms can handle the level of complexity and how many traces they need, to converge and stabilize for their final values.

9.3 Replicate situations

The experiments used in this research are based on simulation runs for different situations of the simulation model. Every experiment is a different situation, see Section 5.2 or [10]. However, these are all single runs for different situations. This is not problem, but for certain simulation models, each run is unique. Thus, it would be interesting to see how the different experiments would behave when for example experiment 411 is run multiple times. This would produce comparable results and one would be able to see whether the average throughput time would behave the same. When a simulation is run more than once, different results can be obtained, the variability between these results is called the Monte Carlo Error (MCE) [15], and should then be accounted for. One could also vary the run time for these different runs. This would result in a different amount of traces and the situation could then be analyzed similarly as done in this research. The benefit would be that the different sets of traces are independent of each other but still comparable.

10. ACKNOWLEDGMENTS

Special thanks to Bemthuis, Rob H. for supervising this project and providing the raw event logs from the simulation models.

11. REFERENCES

[1] W. M. P. Van Der Aalst, B. F. Van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. J. M. M. Weijters, “Workflow mining: A survey of issues and approaches,” doi: 10.1016/S0169-023X(03)00066-1.

[2] W. M. P. van der Aalst, “Process mining and simulation: A match made in heaven!,” *Simul. Ser.*, vol. 50, no. 10, pp. 39–50, 2018, doi: 10.22360/summersim.2018.scsc.005.

[3] W. M. P. Van Der Aalst, “Process Mining in the large: A tutorial,” doi: 10.1007/978-3-319-05461-2.

[4] W. Van der Aalst, *Process mining: data science in action*. 2016.

[5] W. Van Der Aalst *et al.*, “LNBIP 99 - Process mining Manifesto,” 2012.

[6] W. Van Der Aalst, T. Weijters, and L. Maruster, “Workflow mining: discovering process models from event logs,” *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, pp. 1128–1142, 2004, doi: 10.1109/TKDE.2004.47.

[7] A. Adriansyah, B. F. Van Dongen, and W. M. P. Van Der Aalst, “Conformance checking using cost-based fitness analysis,” *Proc. - IEEE Int. Enterp. Distrib. Object Comput. Work. EDOC*, no. May 2014, pp. 55–64, 2011, doi: 10.1109/EDOC.2011.12.

[8] A. Adriansyah, J. Munoz-Gama, J. Carmona, B. F. van Dongen, and W. M. P. van der Aalst, *Measuring precision of modeled behavior*, vol. 13, no. 1. 2014.

[9] A. Adriansyah, “Cost-based conformance checking using the a* algorithm,” *BPM Cent. Rep. BPM- …*, 2011, [Online]. Available: http://www.win.tue.nl/~aadrians/publications/2011-BPMCenter-CostBasedLogReplay.pdf.

[10] R. H. Bemthuis, M. Koot, M. R. K. Mes, F. A. Bukhsh, M. E. Iacob, and N. Meratnia, “An agent-based process mining architecture for emergent behavior analysis,” *Proc. - IEEE Int. Enterp. Distrib. Object Comput. Work. EDOCW*, vol. 2019-Octob, pp. 54–64, 2019, doi: 10.1109/EDOCW.2019.00022.

[11] R. H. Bemthuis, M. Mes, M.-E. Iacob, and P. Havinga, “Using agent-based simulation for emergent behavior detection in cyber-physical systems,” in *Winter Simulation Conference (WSC). IEEE,* 2020.

[12] M. L. Van Eck, X. Lu, S. J. J. Leemans, and W. M. P. Van Der Aalst, “PM2: A process mining project methodology,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9097, pp. 297–313, doi: 10.1007/978-3-319-19069-3_19.

[13] J. Geyer-klingeberg, J. Nakladal, F. Baldauf, and F. Veit, “Process mining and robotic process automation : A perfect match process mining as enabler for RPA implementation,” *16th Int. Conf. Bus. Process Manag.*, vol. i, no. July, 2018.

[14] K. Hoad, S. Robinson, and R. Davies, “Automating warm-up length estimation,” *J. Oper. Res. Soc.*, pp. 1389–1403, 2010.

[15] E. Koehler, E. Brown, S. A. J-P Haneuse, and S. A. J-p Haneuse, “On the assessment of monte carlo error in simulation-based statistical analyses,” *Am. Stat.*, vol. 63, no. 2, pp. 155–162, 2009, doi: 10.1198/tast.2009.0030.

[16] A. M. Law, *Simulation modeling and analysis, FIFTH EDITION*. 2015.

[17] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, “Lawler, Lenstra, Kan, and Shmoys - Sequencing and scheduling algorithms and Complexity,” *Handbook in OR & MS, Vol 4*. 1993.

[18] S. J. J. Leemans, D. Fahland, and W. M. P. Van Der Aalst, “Discovering block-structured process models from event logs containing infrequent behaviour,” 2014.

[19] Y. Lu, “Industry 4.0: A survey on technologies, applications and open research issues,” *J. Ind. Inf. Integr.*, vol. 6, pp. 1–10, 2017, doi: 10.1016/j.jii.2017.04.005.

[20] A. Matta, G. Pedrielli, and A. Alfieri, “Event relationship graph lite: Event based modeling for simulation-optimization of control policies in discrete event systems,” in *Proceedings - Winter Simulation Conference*, vol. 2015-Janua, pp. 3983–3994, doi: 10.1109/WSC.2014.7020223.

[21] M. Mesabbah and S. McKeever, “Presenting a hybrid processing mining framework for automated simulation model generation,” in *Proceedings - Winter Simulation Conference*, 2019, vol. 2018, pp. 1370–1381, doi: 10.1109/WSC.2018.8632467.

[22] S. Robinson, *Simulation: The practice of model development and use*. Wiley, 2004.

[23] A. Rozinat and W. M. P. Van Der Aalst, “Conformance checking of processes based on monitoring real behavior,” *Inf. Syst.*, vol. 33, pp. 64–95, 2008, doi: 10.1016/j.is.2007.07.001.

[24] L. Da Xu, E. L. Xu, and L. Li, “Industry 4.0: state of the art and future trends,” *Int. J. Prod. Res.*, vol. 56, no. 8, pp. 2941–2962, 2018, doi: 10.1080/00207543.2018.1444806.