International Journal of Innovative Computing, Information and Control
Volume **14**, Number **5**, October **2018**

 ISSN 1349-4198
pp. 1565–1582

MODELING PARALLEL BUSINESS PROCESS USING MODIFIED TIME-BASED ALPHA MINER

Yutika Amelia Effendi and Riyanarto Sarno

Department of Informatics
Faculty of Information and Communication Technology
Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo, Surabaya 60111, Indonesia
yutika.effendi@gmail.com; riyanarto@if.its.ac.id

Received December 2017; revised April 2018

Abstract. *One part of process mining is process discovery. A process discovery collects the information from an event log and obtains the business process model representing the behavior of activities based on the information. A business process model contains activities and their relations, specifically sequential and parallel relations. In this paper, Modified Time-based Alpha Miner algorithm, which is a modification of the Alpha Miner algorithm is presented. The proposed algorithm considers both the sequence of activities and time interval information from the event log to discover parallel business process model. Time interval information from event log can be used to determine parallel gateway AND and OR of business process. The experimental results show that the process models formed by parallel gateway AND and OR are able to be discovered using the proposed algorithm, whereas using the same process models, the existing Alpha algorithm determines them as a parallel gateway AND. The results also show that the proposed algorithm needs less traces than the existing Alpha Miner to discover process models. To evaluate the proposed algorithm and Alpha Miner, the comparison of fitness value is presented. It also clearly states that our proposed algorithm can present better results rather than the existing Alpha Miner algorithm.*
Keywords: Process mining, Process discovery, Double timestamp, Business process model, Alpha Miner, Parallel relation, Fitness, Time interval, Activity lifespan, Gantt chart

1. Introduction.

Information systems are used by organizations or people for supporting decision-making or management of business processes [1]. These systems produce an event log that represents the processed activities of business processes. However, a huge amount of event log makes business processes difficult to be analyzed directly. A study for analyzing business processes based on event log becomes a concern [2].

To analyze the business process, the first thing we need to have is the business process model. The business process model is one of the main input and main output for the tasks of process mining. The aim of the process model is to constitute a guidance for verifying and analyzing the performance of the existing business process [3]. There are various kind representations of the process model. Several of them are Petri Net, BPMN, YAWL and Causal Nets. All of those representations have the same main element, i.e., activity and relations among them.

Activities become the first main element because the main display of the process model is the sequences of activities that occur in the business process models. Furthermore, the second main element is the relations between those activities. The significant difference

DOI: 10.24507/ijicic.14.05.1565

of those representations is the formation of the relations. There are two kinds of relations in a business process model, which are parallel and sequential relations. The sequential relation is used to link one activity and another activity, whereas the parallel relation is used to link one or more activities and other activities.

The technique to analyze the process models based on event log is known as process mining. Process mining connects analysis in data-oriented, i.e., data mining and machine learning and analysis in modeling processes. The aim of process mining is observing processes based on event log. Recently, in the business area [4], environment [5], smartphone [6], and fraud [7] have implemented process mining technique. Actually, there are three parts of process mining, which are process discovery, conformance, and enhancement. In this research, we focus to solve the problem in process discovery area because the most concerned task in process mining is process discovery. A process discovery collects the information from an event log and obtains the process model representing the behavior of activities based on the information [8]. The main goal of process discovery is obtaining process model which describes the real business process models [9].

In the implementation, there are several issues that appear during the process of discovering correct process models. Many algorithms of process discovery are proposed. Each proposed method focuses on handling the issues. They are divided into two categories, which are deterministic algorithms, such as Alpha [10], Alpha+, Alpha++ [11], Alpha#, Alpha$ and heuristics algorithms, such as Heuristics Miner [12] and Fuzzy Miner. Alpha, Alpha+, Alpha++, Alpha# and Alpha$ algorithms are able to discover business process models using sequential and reciprocal relation. They also distinguish parallel gateway AND and XOR of business process model, but none of them is able to distinguish parallel gateway OR of business process model. Meanwhile, the Heuristics Miner and Fuzzy Miner algorithm use frequency of each activity to model the business process and only distinguish parallel gateway AND and XOR.

Because of the parallel gateway OR in the business process model, we propose a modification of the existing Alpha Miner algorithm, called Modified Time-based Alpha Miner so that this algorithm is able to distinguish parallel gateway AND and OR in business process models by considering both the activities and their time interval information. Modified Time-based Alpha Miner algorithm is the extended version of the existing Alpha Miner [10] and process model discovery based on activity lifespan [13]. In this research, start time and end time of event log are used in the proposed algorithm so that we can take benefit from them, especially when the activities in the event log have overlap time [2,13]. From their overlap time, we can analyze them as parallel business process.

The differences between existing Alpha Miner and our proposed algorithm are in the determining of parallel business process and the minimum of traces in the event log used to discover the business process. Alpha Miner classifies the sequential and parallel relations of activities in the business process from all cases in the event log using reciprocal relation and determines only two types of the parallel gateway relation, i.e., AND and XOR relations. Whereas Modified Time-based Alpha Miner uses the time intervals from event log to determine the sequential and parallel relations of activities in the business process and determine three types of the parallel gateway relation, which are AND, OR and XOR relations. If the timestamp in the event log overlaps, then the relation of the process model is parallel. Meanwhile, if the timestamp does not overlap, then the relation of the process model is sequential. To determine the parallel gateway, we calculate the number of activities directly and indirectly followed by other activities which have the overlapped time in the event log. This method can easily distinguish the parallel gateway AND and OR relation in the business process. In this research, the proposed method will define several definitions and conditions to discover parallel gateway AND and OR relations. Other than that, considering the time interval from event log is very helpful because it needs less traces than only considering the sequential of activity using reciprocal relation.

In addition, the Alpha, Alpha+, and Alpha++ algorithms discover the flow between activities using single timestamp principle. However, the single timestamp principle is less efficient compared to double timestamp principle using activity lifespan. The double timestamp principle needs less data in the event log compared to single timestamp principle. Hence, it can be stated that the double timestamp principle can reduce the need of complete event log. So, in this research we use double timestamp event log. After we discover the process model, it is important to evaluate the fitness value of the discovered process model. The fitness value measures how many traces in the event log that are represented in the discovered process model. If the fitness value is high, then a process model fits the reality well.

This paper is separated into following sections. In Section 2, we present the literature reviews related to this research. The literature reviews will explain the differences between all parallel gateway relations, single timestamp and double timestamp principle. The proposed algorithm and the calculation of fitness value, including steps by steps in converting single timestamp into double timestamp will be explained in Section 3. The experimental results present the discovered process model, fitness value and the number of traces from the proposed method and the existing Alpha Miner algorithm in Section 4. Lastly, conclusions are presented to end this paper in Section 5.

2. **Literature Reviews.** In this section, we explain the differences between parallel AND, OR and XOR relation and also single timestamp and double timestamp principle which will be the basis of the proposed method. The existing Alpha Miner algorithm can only distinguish parallel AND and XOR; however, in the process mining, there are actually three types of parallel relations which are AND, OR and XOR.

2.1. **Parallel business process.** Each activity has relations with other activities in the business process model. The relation of activities consists of the parallel and sequential relations. There are three types of parallel gateway relations, i.e., XOR, AND and OR. If we only select exactly one activity to be executed from the process model, then the parallel relation is XOR. OR relation happens when we select one or multiple activities to be executed in the process model. Meanwhile, AND relation is defined if the process model permits some activities to be executed respectively [2].

Because OR relation has high flexibility in the executions of activities, a lot of process discovery algorithms have difficulties in interpreting this relation. OR relation will be equated with XOR relation or AND relation, even though both XOR relation and AND relation do not approach the function of OR relation.

2.2. **Single timestamp event log and double timestamp event log.** Initially, the first form of the event log can be categorized as single timestamp event log. The main characteristic of this event log is only one recorded timestamp for each activity. Mostly timestamp records the time when the activity was finished. The single timestamp event log is used in pioneer process mining technique, like Alpha Miner algorithm [10] and Heuristics Miner algorithm [11].

Nowadays, not all activities in the business process executed sequentially. Several activities are executed in parallel. An activity which is executed in parallel is able to minimize the execution time of a process compared with activity which is executed in sequential. However, using single timestamp event log makes the parallel activity difficult to detect. It is because the single timestamp event log only records the end time of the activity, whereas the detection of the parallel activity needs the start time and end time. It is the reason of double timestamp event log occurring [14]. Several algorithms use the double timestamp event log in their process discovery, such as Heuristics Miner with Time Interval algorithm [14] and Modified Time-based Heuristics Miner [2].

In this paper, we use double timestamp event log to discover business process model. Besides being able to find the process model, the double timestamp event log is very useful to distinguish parallel gateway AND and OR from the discovered process model. Because it utilizes activity lifespan which can show parallelism in the event log. The activity lifespan is its duration from start to finish. Activity lifespan is usually presented in Gantt chart form as shown in Figure 1. From Figure 1, it shows that Maxime and Atika are in parallel relation, so are Jon and Adit. Meanwhile, single timestamp principle regards the relation between activities as sequential relation unless it is reciprocal relation (AB, BA). The reciprocal relation is taken as parallel relation. Using double timestamp is more efficient than single timestamp because double timestamp can discover more relations of business process model. Therefore, it needs less number of traces in the event log. Single timestamp and double timestamp principles use the different minimum numbers of traces in the event log to discover process model. The Alpha algorithm which uses single timestamp principle needs $n+1$ traces as minimum traces, whereas the minimum traces of double timestamp are formulated as $(n/2)+1$ with n as the number of parallel activities [15]. The summary of differences between single timestamp and double timestamp is shown in Table 1.

Table 2 shows an example of double timestamp event log of business process model in the organization. From the event log, we get the benefits using double timestamp event log rather than that of single timestamp event log. Double timestamp event log is used to discover the parallel relation of business process based on time interval. Single timestamp

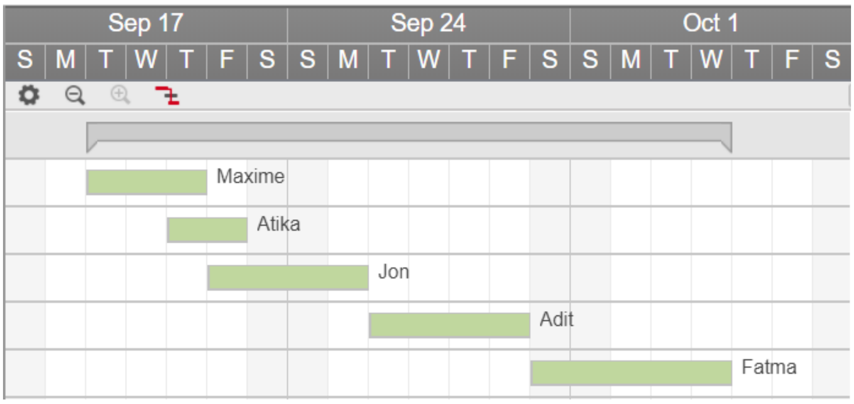


FIGURE 1. Example of Gantt chart

TABLE 1. The differences between single timestamp and double timestamp

Single timestamp	**Double timestamp**
Single timestamp event log only records the end time of the activity	Double timestamp event log records the start time and end time of the activity
A single timestamp requires twice in terms of the number of traces than a double timestamp in the case of a parallel-containing model	
Single timestamp requires reciprocal relation in discovering parallel relation	Double timestamp using activity lifespan to discover parallel relations

TABLE 2. An example of the double timestamp event log in the organization

Trace	Task	Start time	End time
1	A	2015-01-28 10:24:50	2015-01-28 10:24:57
	B	2015-01-28 10:25:00	2015-01-28 10:25:23
	D	2015-01-28 10:25:05	2015-01-28 10:25:18
	E	2015-01-28 10:25:27	2015-01-28 10:25:37
	C	2015-01-28 10:25:43	2015-01-28 10:26:21
2	A	2015-01-28 10:25:46	2015-01-28 10:26:21
	D	2015-01-28 10:26:33	2015-01-28 10:27:59
	B	2015-01-28 10:27:08	2015-01-28 10:28:27
	E	2015-01-28 10:28:17	2015-01-28 10:28:45
	C	2015-01-28 10:28:51	2015-01-28 10:29:20

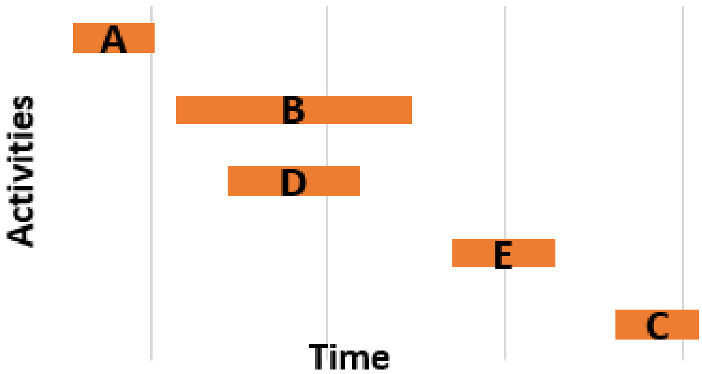


FIGURE 2. Gantt chart from Trace 1 of event log in Table 2

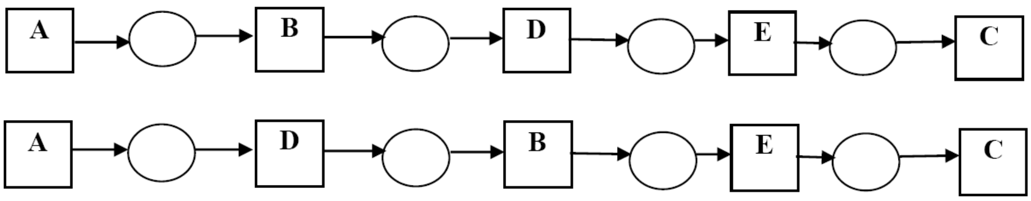


FIGURE 3. Trace 1 and Trace 2 of event log in Table 2

event log uses reciprocal relation to determine the parallel relation. From event log in Table 2, there are two traces in total.

We use Trace 1 and Trace 2 as examples to show the comparison between double timestamp and single timestamp event log. Gantt chart is used to present Trace 1 as shown in Figure 2. Activity B and activity D have overlap time, so we directly analyze them as parallel relation in business process model. Meanwhile, if we use single timestamp event log, then we need Trace 1 and Trace 2 to determine activity B and activity D as parallel relation, as shown in Figure 3. From Figure 3, we can see that activity B and activity D are in reciprocal relation so that the Alpha algorithm determines them as parallel relation. We can conclude that double timestamp event log needs less traces than that of single timestamp event log in discovering process model. Based on the event log in Table 3, double timestamp event log can discover parallel relation with only one trace, which is Trace 1, while single timestamp event log can determine parallel relation if there are minimum two complete traces, which are Trace 1 and Trace 2 as shown in Figure 3.

Table 3. Traces used in single timestamp and double timestamp based on Table 2

Single timestamp	**Double timestamp**
A**BD**EC	A**BD**EC
A**DB**EC	

We obtain all traces used in single timestamp and double timestamp event log in Table 3. It clearly shows that double timestamp need less traces rather than single timestamp using the same event log in Table 2. Double timestamp event log can discover process model using 1 trace, i.e., ABDEC. Meanwhile, single timestamp event log can only discover process model using 2 traces, i.e., ABDEC and ADBEC. In addition, single timestamp event log needs traces which contain BD and DB to decide the parallel relation of business process.

3. Proposed Method.

In this section, our modification algorithm, Modified Time-based Alpha Miner algorithm involves activities in the event log expressed as time interval to reduce the number of traces. The algorithm is able to distinguish parallel gateway OR from AND relation pattern. Time interval utilizes a double timestamp event log to discover sequential and parallel relations of activities in the event log [2,16]. This section also provides the formula to calculate the fitness value of the discovered process model as one of the evaluation criteria.

3.1. Converting single timestamp using sojourn time into double timestamp event log.

The event log can be divided into two types based on the recorded timestamp. The types are single timestamp event log and double timestamp event log. In the organization which runs business process, the event log is available either single timestamp or double timestamp [17]. If the organization provides the start time and end time for all the executed activities, then this algorithm can be directly used to discover the process model. However, there is organization which provides only the end time for all the executed activities, and then the first thing we need to do is converting the single timestamp into double timestamp event log. There are two ways to converting the single timestamp into double timestamp. First way, we can use estimated execution duration from the expert of the organization. We can get the start time of each activity by substracting the end time with the estimated execution time. And the other way, sojourn time can be used to convert single timestamp into double timestamp. Sojourn time is the total time, including execution duration and waiting time of activities needed to complete the process. The steps to convert the single timestamp into double timestamp using sojourn time are:

1) Get the sojourn time for all activities in the event log

$$SojournTime_{activityB} = EndTime_{activityB} - EndTime_{activityA} \tag{1}$$

2) Choose the upper bound and lower bound for each activity from sojourn time
3) Calculate the median value between upper bound and lower bound for each activity
4) Do the normalization for each activity

$$Normalization = \frac{Median\,Value}{AverageofSojournTime} \tag{2}$$

5) Calculate the standard deviation for each activity

$$Stdev = \frac{0.05}{Normalization} \tag{3}$$

6) Get the execution duration for each activity using normal random numbers

$$Norm.inv = Rand();\ normalization;\ stdev \tag{4}$$

7) To obtain the start time for each activity, we substract the end time with execution duration and to obtain the waiting time activity, we substact the end time with start time of activity

$$StartTime_{activityB} = EndTime_{activityB} - ExecutionDuration_{activityB} \tag{5}$$

For example, we have the single timestamp event log in Table 4. Using steps 1 until 5, we can get the start time of each activity. If there are start time and end time of each activity, then we can apply the proposed method to discovering the process model. In Table 4, we explain how to get the sojourn time, average of sojourn time, upper bound, lower bound, median, normalization, standard deviation, and execution duration using Equations (1), (2), (3) and (4) for activity A and activity B in Table 4.

Table 4. Steps for converting single timestamp into double timestamp event log

Case ID	Activity	Timestamp	Sojourn time	Avg sojourn time	Upper bound	Lower bound	Median	Normalization	Stdev	Execution duration
ID01	A	2014-06-20 13:42:02	05:10	06:36	09:59	05:05	07:32	03:22	01:22	04:39
	A	2014-06-20 23:41:25	09:59							04:18
	B	2014-06-21 08:16:31	08:34	10:16	15:35	06:41	11:08	01:59	01:18	01:54
ID02	A	2014-06-21 16:57:19	06:11							03:09
	B	2014-06-22 10:51:47	15:35							01:16
ID03	A	2014-06-23 04:48:10	05:05							03:10
	B	2014-06-23 23:26:19	06:41							03:54

After we get the execution duration of activity A and activity B, then we use Equation (5) to get the start time of activity A and activity B as shown in Table 5. Based on Table 5, the end time of activity A in Case ID01 is 13:42:02 and the execution duration of activity A based on Table 4 is 4 hours 39 minutes, so we substract them to get the start time of activity A which is 09:03:02. The same steps apply to activity B in the event log.

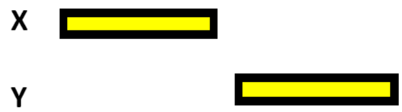
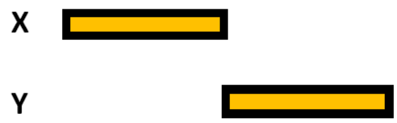
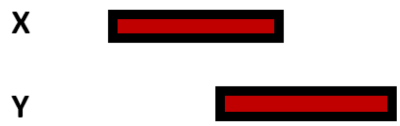
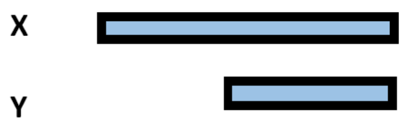
Table 5. The results in double timestamp event log

Case ID	Activity	Start time	End time
ID01	A	2014-06-20 09:03:02	2014-06-20 13:42:02
	A	2014-06-20 19:23:25	2014-06-20 23:41:25
	B	2014-06-21 06:22:31	2014-06-21 08:16:31
ID02	A	2014-06-21 13:48:19	2014-06-21 16:57:19
	B	2014-06-22 09:34:47	2014-06-22 10:51:47
ID03	A	2014-06-23 01:38:10	2014-06-23 04:48:10
	B	2014-06-23 19:31:19	2014-06-23 23:26:19

3.2. **Temporal causal relation.** Six types of temporal causal relations between activities are defined as extended version of standard relations in the event log [13]. We present a concept of double timestamp principle in order to discover the relations of business process model based on time interval. Based on the definition, we also show the process model to represent each definition. For sequential relation, we classify them as *before* and *meets*. Meanwhile, for parallel relation, we classify them as *overlaps*, *contains*, *equals*, *has the same end time*, and *has the same start time*.

Definition 3.1. *Given event log (L) and trace (σ) such that $\sigma \in L$. The causal relation between two activities $X(X_s, X_f)$ and $Y(Y_s, Y_f)$, according to which X, $Y \in L$ can be differentiated as explained in Table 6 which are before, meets, overlaps, contains, equals, has the same end time, and has the same start time.*

TABLE 6. Temporal causal relation

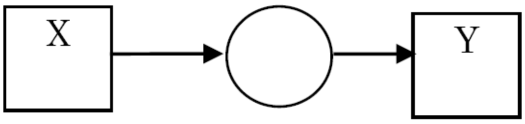
Before, $X > Y$ *iff* $X_f \leq Y_s$	
Meets, $X > Y$ *iff* $X_f \leq Y_s$	
Ovelaps, $X\text{O}Y$ *iff* $X_f > Y_s \wedge X_f < Y_f$	
Contains, $X@Y$ *iff* $X_s < Y_s \wedge Y_f > X_s \wedge X_f > Y_f$	
Equals, $X \approx Y$ *iff* $X_s = Y_s \wedge X_f = Y_f$	
HastheSameEndTime, $X_f Y$ *iff* $X_f = Y_f \wedge X_s < Y_s \wedge Y_s < X_f$	
HastheSameStartTime, $X \rho Y$ *iff* $X_s = Y_s \wedge X_f = Y_f$	

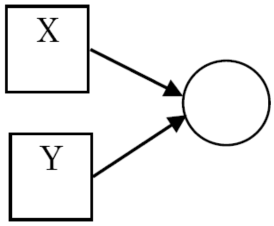
From the temporal causal relation in Table 6, we can model the activity X and activity Y based on their time interval. Table 7 shows the process model formed for each temporal causal relation. It shows that *before* and *meets* are sequential relation and the others are parallel relations.

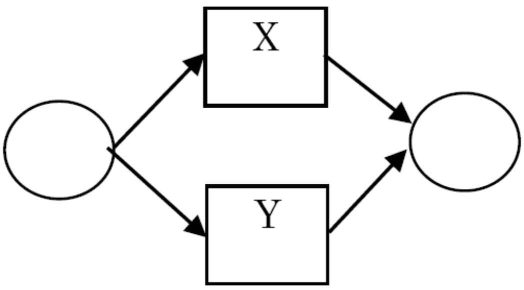
3.3. **Control flow.** To make the discovered process model able to clearly show the control flow between activities and their temporal causal relations (i.e., parallel and sequential

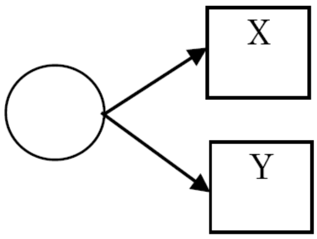
TABLE 7. Process model for each temporal causal relation

<table>
<tr>
<td>Temporal Causal Relation:
before, meets

Business process model:

</td>
<td>Temporal Causal Relation:
has the same end time

Business process model:

</td>
</tr>
<tr>
<td>Temporal Causal Relation:
overlaps, contains, equals

Business process model:

</td>
<td>Temporal Causal Relation:
has the same start time

Business process model:

</td>
</tr>
</table>

relation), we need to give a formal definition of control flow as well. A process variant contains control flows similar to those of a process model (i.e., sequence and parallel) [13].

Definition 3.2. *Given event log (L) and trace (σ) such that $\sigma \in L$. The causal relation between two activities $X(X_s, X_f)$ and $Y(Y_s, Y_f)$, according to which X, $Y \in L$ can be differentiated as follows:*

- *Sequence,* $X \rightarrow Y$ *iff* $X > Y \wedge Y \mathrm{O} X$
- *NotRelated, iff* $X \mathrm{O} Y \wedge Y \mathrm{O} X$
- *Parallel,* $X||Y$ *iff* $X > Y \wedge Y > X \vee \{X \mathrm{O} Y \vee X @ Y \vee X_f Y \vee X \approx Y \vee X \rho Y\}$
- *XORrelation,* $X \otimes Y$ *iff* $X||Y$ *if there exists only* X *or* Y *in any traces in the event log (L)*
- *ANDrelation,* $X \bullet Y$ *iff* $X||Y$ *and there does not exist* $X \otimes Y$ *in any traces in the event log (L)*
- *ORrelation,* $X \oplus Y$ *iff* $X||Y$ *and there exists* $X \otimes Y$ *in any traces in the event log (L)*

3.4. **Integrated discovery approach.** In this section, we describe the steps taken by our proposed algorithm using temporal causal relation and control flow as defined in the previous section. The main steps are, first, listing of all input and output activities for each trace; then, classifying all sequential and parallel relations, and finally, we display the complete set of relations of activities in the event log. There are 13 steps used in our proposed algorithm, Modified Time-based Alpha Miner to discover the business process model.

Step 1. Classify the sequence relation (>) from every case in the event log

Step 2. Classify the parallel relation ($||$) from every case in the event log
Step 3. Remove duplicate sequence and parallel relations from all cases in the event log to get the number of traces
Step 4. Obtain all traces with sequence and parallel relations in the event log
Step 5. Create set of transition (T_L) in workflow net

$$T_L = \{t \in T | \exists_{\sigma \in L} t \in \sigma\}$$

Step 6. Create set of input transition from the source place

$$T_I = \{t \in T | \exists_{\sigma \in L} t = \mathit{first}(\sigma)\}$$

Step 7. Create set of output transition from the sink place

$$T_o = \{t \in T | \exists_{\sigma \in L} t = \mathit{last}(\sigma)\}$$

Step 8. Create the gantt chart for all traces based on temporal causal relation and control flow to clearly show the relation among all activities in the traces
Step 9. Create the places

$$P_L = \left\{p_{(A,B)} | (A, B) \in Y_L\right\} \cup \{I_L, O_L\}$$

Step 10. Create a process model based on the sequence and parallel relations
Step 11. Classify the type of parallel relation, i.e., AND or OR

$$\begin{aligned}&X \bullet Y,\ X \oplus Y \mathit{\ iff\ } X||Y \mathit{\ and\ there\ exists\ } Z \in L\\&\mathit{where\ } X > Z \vee Y > Z \vee X\mathrm{O}Z \vee Y\mathrm{O}Z \mathit{\ in\ any\ } s \in L\end{aligned}$$

AND relation

$$\begin{aligned}&\mathit{foreach}\ L\ \mathit{in}\ \bullet, \mathit{which}\ (X,Y) \wedge (Z,A) \in L\\&\mathit{iff}\ X = [Y,(Z \wedge A)] \wedge [Y,(A \wedge Z)] \wedge [(Z \wedge A),Y] \wedge [(A \wedge Z),Y] \in >_L\\&\mathit{then}\ [A,(B,(C \wedge D))]\end{aligned}$$

OR relation

$$\begin{aligned}&\mathit{foreach}\ L\ \mathit{in}\ \oplus, \mathit{which}\ (X,Y) \wedge (Z,A) \in L\\&\mathit{iff}\ X = (Z \vee A) \wedge [(Z \vee A),Y] \vee [Y,(Z \vee A)] \in >_L\ \ \mathit{then}\ [X,(Y,(Z \vee A))]\end{aligned}$$

Step 12. Add the input, output and sequence relations into the process model

$$\begin{aligned}&\mathit{foreach}\ R\ \mathit{in}\ >_L,\ \mathit{which}\ (X,Y) \in L\\&\qquad \mathit{iff}\ (X,Y) \notin G\\&\qquad \mathit{iff}\ (\bullet Y)\ \mathit{not\ exist}\\&\qquad G \leftarrow G \cup (X,Y)\\&\qquad\qquad \mathit{elseiff}\ X \bullet Z,\ \mathit{which}\ (Z,Y) \in G\\&\qquad\qquad G \leftarrow G \cup [(X,Z) \bullet B]\\&\qquad\qquad\qquad \mathit{elseiff}\ X \oplus Y,\ \mathit{which}\ (Z,Y) \in G\\&\qquad\qquad\qquad G \leftarrow G \cup [(X,Z) \oplus B]\\&\qquad\qquad\qquad\qquad \mathit{else}\ G \leftarrow G \cup [(X,Z) \oplus B]\end{aligned}$$

Step 13. A process model in Petri nets form is complete

$$\alpha(L) = (P_L, T_L, F_L)$$

3.5. **Determining the parallel gateway in the business process model.** As stated before, from the discovered business process model, we can determine the parallel relations which consist of AND, OR and XOR. The calculation of parallel relations requires the frequency of parallel activity in the event log. We can get the frequencies by adding all the parallel relations which have the same parent activity directly and indirectly followed by other activities in the business process model. The threshold intervals used to determine the parallel gateway AND, OR and XOR of business process model are:

- XOR relation

$$\textit{If Avg PM} \leq \textit{Minimum All Sequence Relation in the event log (L), then XOR} \tag{6}$$

- OR relation

$$\begin{aligned}&\textit{If Minimum All Sequence Relation} \leq \textit{Avg PM} \leq \textit{Avg All Sequence Relation}\\&\textit{in the event log (L), then OR}\end{aligned} \tag{7}$$

- AND relation

$$\textit{If Avg All Sequence Relation} \leq \textit{Avg PM in the event log (L), then AND} \tag{8}$$

where:
Avg: average
Avg PM: the average of parallel which has the same parent activity; frequency of one activity directly and indirectly followed by other activity

Based on Equation (6), parallel gateway XOR happens when average of parallel from process models is less than the minimum of all sequence relations. We also can determine the parallel gateway OR if the average of parallel from process model is in the middle between minimum of all sequence relations and average of all sequence relations using Equation (7). Meanwhile, a parallel relation is classified as AND when average of parallel from process model is greater than the average of all sequence relations from Equation (8).

3.6. **Calculating fitness function of the business process model.** In this paper, fitness function is calculated because it contains the evaluation criteria of quality of business process model. There are two terms of fitness function which should be known. The fitness will be high if all activities correctly describe in the event log. Otherwise, the fitness will be low if many activities do not correctly describe in the event log.

In this research, the number of traces which are correctly parsed from event log is related with fitness function. This term cannot be applied in noisy situation because process model cannot correctly parse all traces in the event log. The fitness of a process model can be 100% or 1 in a noise-free situation, this is because all traces in the event log from a process model can be parsed. The value of fitness ranges from 0 to 1. Fitness 1 means all traces in the event log from a process model can be parsed, while fitness 0 means no trace in the event log from process model can be parsed by discovered business process model. Existing Alpha Miner algorithm usually has high fitness because it models all relations in the event log into discovered process model. We can calculate the fitness function using Equation (9).

$$Q_f = \frac{casesCaptured}{casesLog} \tag{9}$$

where:
Q_f: fitness value
casesCaptured: the number of cases in the event log parsed in the process model
casesLog: the number of cases in the event log

4. **Results and Analysis.** In this section, we present the experiments which show that the proposed method can distinguish the parallel relations of business process model formed by AND and OR. We evaluate two kinds of real-life event log in this experiment. Event log which is generated from an organization is single timestamp. So, the first thing we need to do is converting them into double timestamp using sojourn time based on Section 3.1. The result is shown in Table 8. Table 8 is a double timestamp event log which is used in the first experiment.

4.1. **Experimental results.** Double timestamp event log in Table 8 contains 100 cases, 6 activities which are activities A, B, C, D, E, F, start time and end time which are called EL I. Based on Steps 1 and 2 in Section 3.4, we get the relation A→B, B||C, B→D, D||E, C→E, E→F, and D→F from Case ID001, A→B, B||C, B→D, D||E, C→E, E→F, and D→F from Case ID002, A→B, B||C, B→D, D||E, C→E, E→F, and D→F from Case ID003, A→B, B||C, B→D, D||E, C→E, E→F, and D→F from Case ID004, and so on up to Case ID100. After we get all sequence and parallel relations in all cases, we need to obtain the traces by removing the duplicate sequence and parallel relations. From event log in Table 8, we have two traces in total after we do Steps 3 and 4 in Section 3.4. Trace 1 has 80 cases and Trace 2 has 20 cases. All traces with all sequence and parallel relations shown in Table 9. We get that A→B, B→D, C→E, A→C, C→D, E→F, D→F are in sequence relations and B||C, D||E are in parallel relations.

Table 8. Part of double timestamp event log EL I used in this experiment

Case ID	Activity	Start time	End time	Execution duration
ID001	A	2015-01-28 10:05	2015-01-28 10:07	0:02
	B	2015-01-28 10:10	2015-01-28 10:22	0:12
	C	2015-01-28 10:15	2015-01-28 10:39	0:24
	D	2015-01-28 10:27	2015-01-28 11:02	0:34
	E	2015-01-28 10:41	2015-01-28 10:53	0:12
	F	2015-01-28 11:07	2015-01-28 11:28	0:21
ID002	A	2015-01-28 11:33	2015-01-28 11:36	0:03
	B	2015-01-28 11:40	2015-01-28 11:54	0:14
	C	2015-01-28 11:44	2015-01-28 12:14	0:29
	D	2015-01-28 11:59	2015-01-28 12:28	0:29
	E	2015-01-28 12:22	2015-01-28 12:34	0:12
	F	2015-01-28 12:44	2015-01-28 12:56	0:11
ID003	A	2015-01-28 13:01	2015-01-28 13:10	0:09
	B	2015-01-28 13:15	2015-01-28 13:28	0:12
	C	2015-01-28 13:21	2015-01-28 13:36	0:15
	D	2015-01-28 13:30	2015-01-28 13:43	0:13
	E	2015-01-28 13:38	2015-01-28 13:53	0:14
	F	2015-01-28 13:59	2015-01-28 14:12	0:13
...	...	...	...	...
ID100				

Table 9. All traces of event log EL I in Table 8

Trace 1	A→B, B||C, B→D, D||E, C→E, E→F, D→F
Trace 2	A→C, C→D, D||E, E→F, D→F

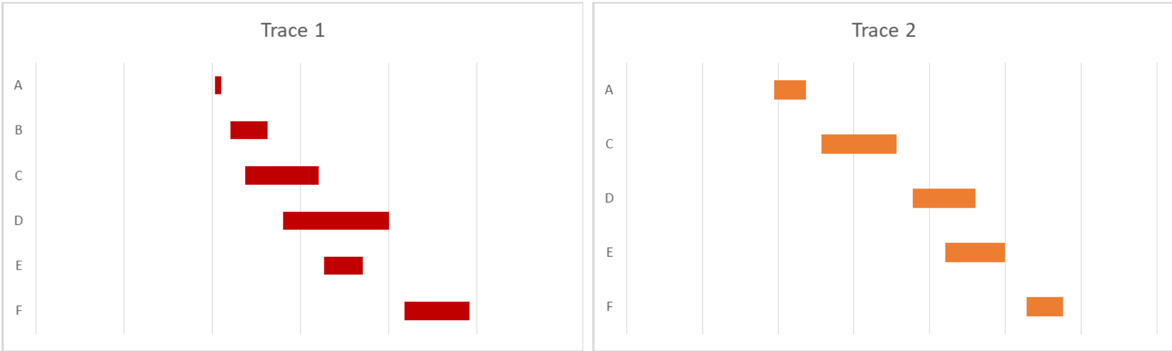


FIGURE 4. All traces of event log EL I in Table 8 presented in Gantt chart

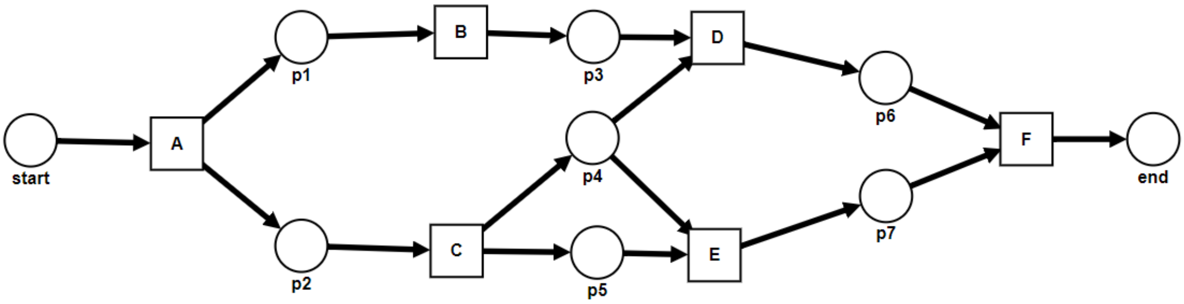


FIGURE 5. A process model of EL I discovered by the proposed algorithm

After we get all the traces from the event log, Gantt chart for all traces is presented. Gantt charts for Trace 1 and Trace 2 as shown in Figure 4 give information that activity A is an input and activity F is an output in the event log. Furthermore, from Trace 1, we know that activity B and activity C should be in parallel relation, as should activities D and E. Trace 2 also explains that activities D and E are in parallel relation. Therefore, we get all the information regarding modeling parallel business process based on Figure 4.

Next step, we create places for each activity as connectors between activities. In this experiment, we create p0 (start), p1, p2, p3, p4, p5, p6, p7, p8 (end). After we have input, output, transition, place, and relations for all activities in the event log, we discover the process model by combining all of them. The discovered process model shows all traces in the event log with respective execution durations as shown in Figure 5. At a glance based on Figure 5, we can easily determine that activities B and C are in parallel relation, so are the activities D and E. Activity A as input is followed by activities B and C, meanwhile activity B is directly followed by activity D and activity C is directly followed by activity D and activity E. Lastly, activities D and E lead to activity F.

After we discover the process model, the next step that we have to do is to classify the parallel relations of process model into XOR, AND or OR using Equations (6)-(8). We need to calculate the frequency of sequence and parallel relations in all cases. The total numbers of each relation of activity in all cases are presented in Table 10. There are 80 cases and 100 cases respectively in the event log which clearly show B||C and D||E.

To determine the parallel relation, we calculate the minimum value of all sequence relations and average of all sequence relations from all relations of activities in Table 10. The results are 20 and 68.57 for minimum of all sequence relations and average of all

Table 10. Total number of each relation of activity in all cases of EL I

Relation of activity	Number of the relations in all cases	Relation of activity	Number of the relations in all cases
A→B	80	D||E	100
B||C	80	C→E	80
A→C	20	E→F	100
B→D	80	D→F	100
C→D	20		

Table 11. The data used to determine the parallel gateway of EL I

Parallel Activities	Minimum of all sequence relations	Avg of parallel	Avg of all sequence relations	Parallel Gateway
B||C	20	60	68.571	OR
D||E		80		AND

Table 12. Final relations of discovered process model of EL I

Input	Split/Join	Output	Input	Split/Join	Output
{Start}	Sequence	A	B, C	OR Join	D, E
A	OR Split	B, C	C	AND Split	D, E
B	Sequence	D	D, E	AND Join	F
C	Sequence	D	F	Sequence	{End}
C	Sequence	E			

sequence relations respectively. Meanwhile, the average of parallel from activity B and activity C is 60 and the average of parallel from activity D and activity E is 80. We get all the data to determine the parallel gateway of the discovered process model in Table 11.

Based on Table 11, we know that the activity B and activity C are in parallel OR relation in business process model because the average of parallel is less than the average of all sequence relations; meanwhile, the activity D and activity E are in parallel AND relation because the average of parallel is higher than the average of all sequence relations. All the relations of the discovered process model are explained in Table 12 and the process model is shown in Figure 6. Apart from the activities B, C, D and E, all the relations of remaining activities are sequence.

We do another experiment using different event log containing 100 cases and 5 traces, which is called event log EL II. The sequence and parallel relations in the event log EL II are A→B, B||C, C→D, D||E, B→D, C→E, D→F, E→F from Trace 1, A→C, C||B, B→D, D||E, C→D, B→E, D→F, E→F from Trace 2, A→B, B||C, C→E, E||D, B→E, C→D, E→F, D→F from Trace 3, A→C, C→D, D||E, C→E, D→F, E→F from Trace 4, and A→C, C||B, B→D, E||D, B→E, C→D, E→F, D→F from Trace 5. Using the same steps to discover process model and to determine the parallel gateway of business process model with the first experiment, Figure 7 shows that Modified Time-based Alpha Miner can discover process model with parallel relations OR and AND from event log EL II.

4.2. **Evaluation of the proposed algorithm.** To evaluate our proposed algorithm, we use two kinds of evaluation, first is the comparison of discovered process model between Modified Time-based Alpha Miner and the existing Alpha Miner algorithm. Second, we

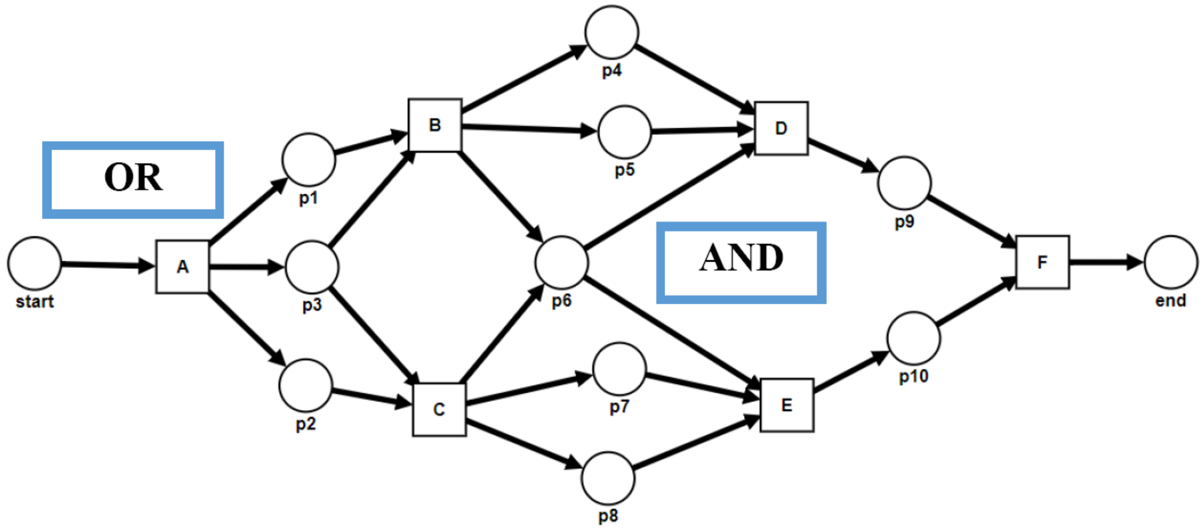


FIGURE 6. The discovered process model of EL I in Petri Net

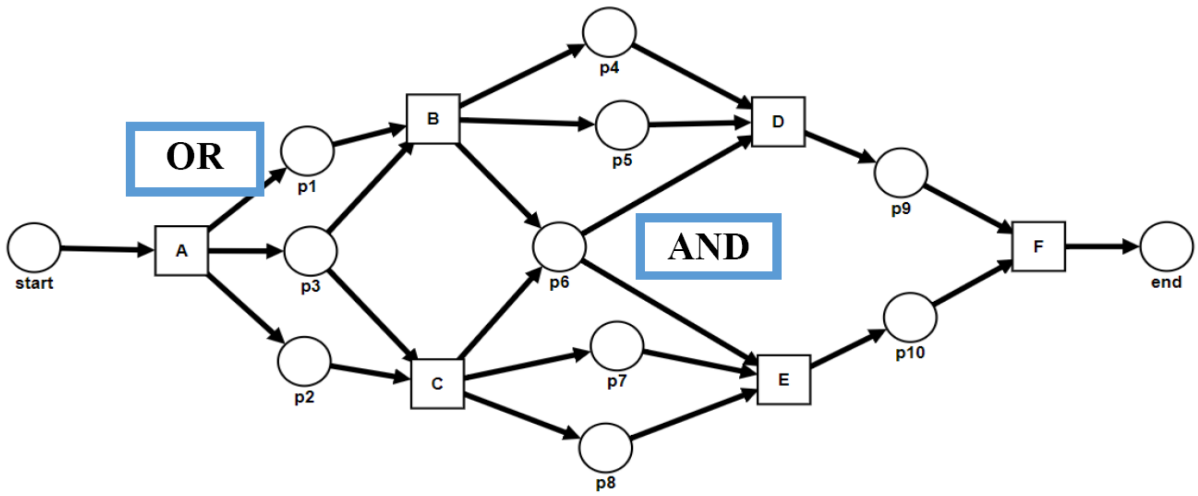


FIGURE 7. Process model of EL II discovered using the proposed algorithm in Petri Net

calculate the fitness function of discovered process model using Equation (9) in Section 3.6 and the total number of traces used in the event log to discover the process model correctly.

Figure 8 presents the process model in the Petri Net form using ProM tool for the same cases of event log EL I as shown in Table 8. Alpha Miner algorithm uses single timestamp principle. Based on Figure 8, we can see that activities B and C are in parallel relation; meanwhile, activity B is not directly followed by activity D. The reason is that Alpha Miner algorithm is able to discover process model based on sequential relation. From the event log shown in Table 8, there is no sequential relation between activities B and D because activity B is directly followed by activity C in all cases, so Alpha algorithm only discovers relation between activity B and activity C.

For the parallel gateway, all parallel relations in the discovered process model which are B||C and D||E in Figure 8 are AND, because Alpha algorithm only has parallel gateway AND and XOR relations. The same thing also applies to traces. Using Alpha algorithm, the total traces which are used to discover process model of EL I are three; meanwhile, using Modified Time-based Alpha Miner, we only need two traces as shown in Table 13.

So, our proposed algorithm which considers the time interval can discover process model better than Alpha Miner algorithm which uses reciprocal relations to discover business process model.

And the last thing to do to evaluate our proposed algorithm is calculation of fitness value using Equation (9). The fitness values of the event log EL I in Table 8 mined by our proposed algorithm and Alpha Miner are 1 and 0.954 as shown in Table 13. Modified Time-based Alpha Miner can discover process model which fits the reality better than that of Alpha Miner algorithm. The result of fitness value shows that there is an increase of the number of all properly completed log traces and all parsed activities in the event log.

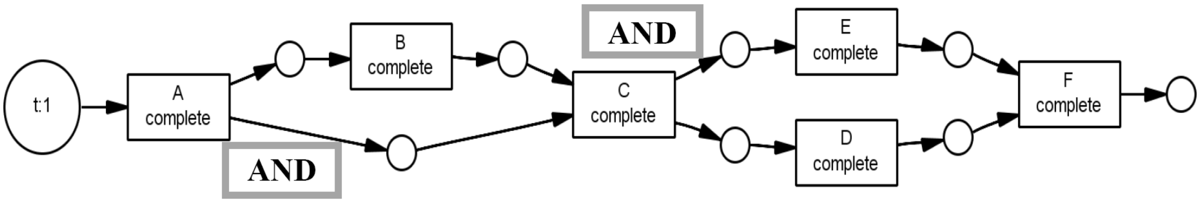


Figure 8. Discovered process model of EL I using Alpha Miner algorithm

Table 13. The final results of event log EL I

Algorithm	Parallel gateway	Fitness value	The number of traces in the EL I	Number of traces used in discovery
Modified Time-based Alpha Miner	B||C: OR D||E: AND	1.000	3	2
Alpha	B||C: AND D||E: AND	0.954		3

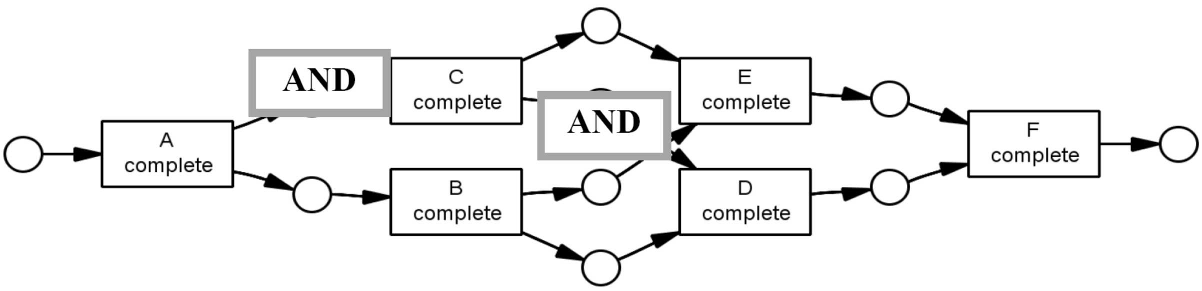


Figure 9. The discovered process model of EL II using Alpha algorithm

Table 14. The final results of event log EL II

Algorithm	Parallel gateway	Fitness value	The number of traces in the EL II	Number of traces used in discovery
Modified Time-based Alpha Miner	B||C, C||B: OR D||E, E||D: AND	1.000	5	3
Alpha	B||C, C||B: AND D||E, E||D: AND	0.977		5

The same thing also applies to event log EL II. We evaluate EL II using ProM tool to discover process model and the result is the same as process model discovered by Modified Time-based Alpha Miner algorithm as presented in Figure 9. This is due to in the event log EL II, there is sequential relation between activities B and D, so the Alpha algorithm can discover process model as same as our proposed algorithm, but the gateways for parallel relations are all AND. We explain other comparisons between our proposed algorithm and Alpha Miner algorithm in Table 14 which are the parallel gateway, the fitness value, and the number of traces for each algorithm.

5. **Conclusions.** In this research paper, we introduced Modified Time-based Alpha Miner algorithm which is a modification of the Alpha Miner algorithm. The proposed algorithm uses activities and their time intervals represented in double timestamp event log to discover sequential and parallel relations of business process model, where the existing Alpha Miner algorithm only considers the sequential and reciprocal relation to discover the business process model. The experimental results have shown that our Modified Time-based Alpha Miner algorithm can discover business process models which contain parallel gateway AND and OR relations, which cannot be discovered by the existing Alpha Miner algorithm. To mine the parallel business process, our proposed algorithm elaborates the temporal causal relation and control flow. Therefore, to determine the parallel gateway AND, OR or XOR, the frequencies of sequential and parallel relations are needed. Based on our experiments, the Modified Time-based Alpha Miner algorithm can discover business process models with less number of traces and also improves the fitness value of the discovered business process models.

Acknowledgment. This research is supported by Information Management Laboratory, Department of Informatics, Faculty of Information and Communication Technology, Institut Teknologi Sepuluh Nopember and the Ministry of Research, Technology and Higher Education of Indonesia. The authors are grateful to the reviewers, who have given the helpful comments and suggestions so that our research presentation can be improved.

REFERENCES

[1] R. Sarno and Y. A. Effendi, Hierarchy process mining from multi-source logs, *Telecommunication, Computing, Electronics and Control (TELKOMNIKA)*, vol.15, no.4, 2017.

[2] R. Sarno, Y. A. Effendi and F. Haryadita, Modified time-based heuristics miner for parallel business processes, *International Review on Computers and Software (IRECOS)*, vol.11, no.3, pp.249-260, 2016.

[3] R. Sarno, W. A. Wibowo, Kartini, F. Haryadita, Y. Amelia and K. Rossa, Determining process model using time-based process mining and control-flow pattern, *Telecommunication, Computing, Electronics and Control (TELKOMNIKA)*, vol.14, no.1, 2016.

[4] R. Sarno, P. L. I. Sari, D. Sunaryono, B. Amaliah and I. Mukhlash, Mining decision to discover the relation of rules among decision points in a non-free choice construct, *Proc. of International Conference on Information, Communication Technology and System (ICTS)*, 2014.

[5] Y. A. Effendi and R. Sarno, Discovering optimized process model using rule discovery hybrid particle swarm optimization, *The 3rd International Conference on Science in Information Technology (ICSITech)*, pp.97-103, 2017.

[6] Y. A. Effendi and R. Sarno, Discovering process model from event logs by considering overlapping rules, *The 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pp.1-6, 2017.

[7] S. Huda, R. Sarno and T. Ahmad, Increasing accuracy of process-based fraud detection using a behavior model, *International Journal of Software Engineering and Its Applications*, vol.10, no.5, pp.175-188, 2016.

[8] W. M. P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, Springer Science and Business Media, 2011.

[9] W. M. P. van der Aalst, A. Adriansyah and B. van Dongen, Causal nets: A modeling language tailored towards process discovery, in *CONCUR 2011 – Concurrency Theory. Lecture Notes in Computer Science*, J. P. Katoen and B. König (eds.), Springer, Berlin, Heidelberg, 2011.

[10] W. M. P. van der Aalst, T. Weijters and L. Maruster, Workflow mining: Discovering process models from event logs, *IEEE Trans. Knowledge and Data Engineering*, vol.16, no.9, pp.1128-1142, 2004.

[11] L. Wen, W. M. P. van der Aalst, J. Wang and J. Sun, Mining process models with non-free-choice constructs, *Data Mining and Knowledge Discovery*, vol.15, no.2, pp.145-180, 2007.

[12] A. J. M. M. Weijters, W. M. P. van der Aalst and A. K. Alves de Medeiros, Process mining with the heuristics-miner algorithm, *Tech. Rep. WP, No. 166*, Technische Universiteit Eindhoven, 2006.

[13] R. A. Sutrisnowati, H. Bae, L. Dongha and M. Kim, Process model discovery based on activity lifespan, *International Conference on Technology Innovation and Industrial Management*, Seoul, pp.137-156, 2014.

[14] A. Burattin and A. Sperduti, Heuristics miner for time intervals, *ESANN 2010 Proceedings, European Symposium on Artificial Neural Networks – Computational Intelligence and Machine Learning*, pp.41-46, 2010.

[15] R. Sarno, Kartini, W. A. Wibowo and A. Solichah, Time based discovery of parallel business processes, *International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*, pp.28-33, 2015.

[16] R. Sarno, F. Haryadita, Kartini, Sarwosri and A. Solichah, Business process optimization from single timestamp event log, *International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*, pp.50-55, 2015.

[17] S. S. Pinter and M. Golani, Discovering workflow models from activities' life spans, *Computers in Industry*, vol.53, pp.283-296, 2004.