ICIC Express Letters
Part B: Applications
Volume **15**, Number **1**, January **2024**

ICIC International ©2024 ISSN 2185-2766
pp. **61–70**

REFINING PROCESS MINING IN PORT CONTAINER TERMINALS THROUGH CLARIFICATION OF ACTIVITY BOUNDARIES WITH DOUBLE-POINT TIMESTAMPS

Yutika Amelia Effendi[1] and Minsoo Kim[2,*]

[1]Robotics and Artificial Intelligence Engineering
Faculty of Advanced Technology and Multidiscipline
Airlangga University
Gedung Kuliah Bersama, Jl. Dr. Ir. H. Soekarno, Mulyorejo, Surabaya, Jawa Timur 60115, Indonesia
yutika.effendi@ftmm.unair.ac.id

[2]Department of Industrial and Data Engineering
Pukyong National University
45 Yongso-ro, Namgu, Busan 48513, Korea
[*]Corresponding author: minsky@pknu.ac.kr

Received May 2023; accepted July 2023

Abstract. *Port container terminals play a vital role in global trade, necessitating effective analysis techniques to optimize operational processes. Process mining methodologies have been widely employed for this purpose, but most existing approaches rely on single-point timestamps, lacking detailed information on activity durations. To overcome this limitation, this paper proposes a new methodology that utilizes the PERT-Beta Distribution to determine a double-point timestamp event log from an organization's traditional database. By incorporating double-point timestamps, the proposed methodology provides a more comprehensive understanding of activity durations. The PERT-Beta Distribution calculates the mean and variance of time for each activity by considering optimistic, most likely, and pessimistic time estimates, along with shape parameters alpha and beta. This approach accounts for uncertainty and enables effective management of project completion time. Additionally, the paper presents a process model discovered using an algorithm based on the double-point timestamp approach. To validate the proposed methodology, real-world data from Terminal Operating Systems (TOS) in Surabaya, Indonesia, are utilized. This research contributes to the field by enhancing process mining techniques through the introduction of a novel methodology that incorporates double-point timestamps and leverages the PERT-Beta Distribution. The case study conducted on a port container terminal provides practical insights for improving operational processes and efficiently managing project completion time.*

Keywords: Double timestamp event log, Conformance checking, PERT-Beta Distribution, Port container terminal, Process discovery, Process mining

1. **Introduction.** Process mining analysis relies on event log data, which should include essential elements for analyzing a process model based on real assumptions [1,2]. These elements typically include Case ID, Activity, and Timestamp. These three elements enable a process-oriented perspective on the data [3,4].

Initially, event logs can be categorized as single-point timestamp event logs, where only one timestamp is recorded for each event [5]. While it is still possible to analyze the time between two process steps using a single timestamp, it is impossible to determine the duration of each activity accurately. This limitation results in the processing time appearing instantaneous. To overcome this, it is crucial to have both start and end timestamps for each activity, which leads to the concept of a double-point timestamp event log

DOI: 10.24507/icicelb.15.01.61

[5]. There are several advantages to using double-point timestamps. Firstly, in a parallel-containing model, a single-point timestamp requires twice the number of traces compared to a double-point timestamp. With a single-point timestamp, a minimum of $n + 1$ traces is needed, while a double-point timestamp only requires $(n/2) + 1$ traces, where n represents the number of parallel activities [6]. Secondly, while single-point timestamp event logs rely on reciprocal relations to determine parallel relations, double-point timestamp event logs utilize time intervals or activity lifespans [6]. Based on these advantages, according to process mining standards, we automate the transformation of an organization's traditional databases (e.g., CSV databases) into event logs.

In this paper, we propose a method that utilizes the beta distribution to define double-point timestamps for each activity in the event log, specifically when the organization's data only contains single-point timestamps for grouped activities. The beta distribution technique allows for the recognition of uncertainty and effective management of project completion time [7]. This method is particularly suitable for capturing the random behavior of proportions in our case study of a port container terminal. Furthermore, once all three requirements for an event log are fulfilled, our approach leverages double-point timestamps in the process discovery phase. While several process discovery algorithms exist, such as alpha miner, genetic process miner, and inductive miner, none of these algorithms have incorporated timestamps in the discovery of process models. Sequential approaches still prevail. Lastly, conformance checking is utilized to evaluate the quality of the discovered process model [3,8]. Based on our proposed methodology which assesses the fitness, precision, and structural appropriateness between the event log and the refined process model, we assert with confidence that the proposed method effectively identifies a process model that faithfully represents the intricacies of the event log.

The structure of this paper is as follows: Section 2 discusses related work, Sections 3 and 4 present the proposed methodology and experimental results, respectively, and Section 5 provides concluding remarks.

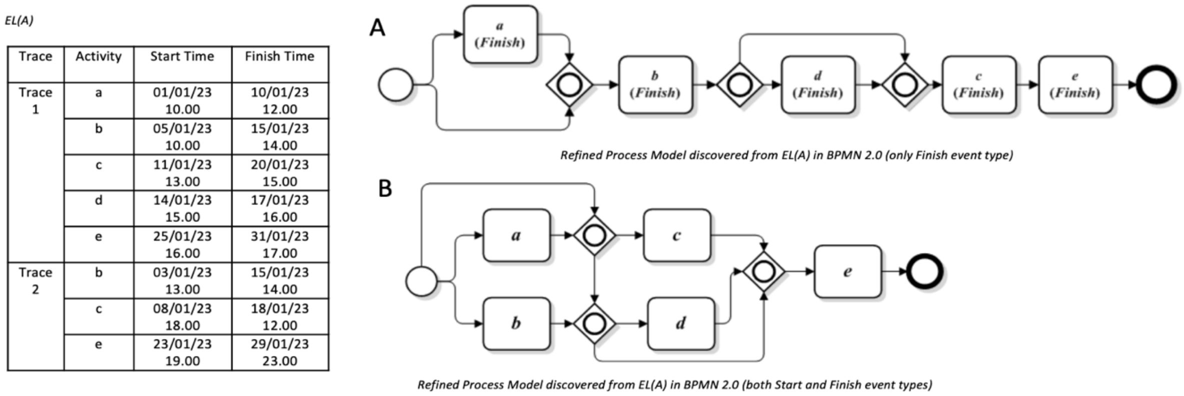
2. **Literature Review.**

2.1. **Event log and process model.**

Process mining has the potential to enhance the mapping of processes, leading to improved process performance. This is achieved through the ease of creating, evidence-based, unbiased, and adaptable process maps in various industries, including healthcare institutions [9-11], manufacturing firms [12,13], automotive companies [14], the study of time-resolved gene expression in biology [15], analysis of website user behavior [16], and can also be integrated with advanced technologies such as deep learning [17], streaming process [18], and robotic process automation [19] to achieve superior outcomes. However, all these studies relied on single-point timestamps and sequential approaches.

In this study, we distinguish between two types of process models based on their use: standard process model and refined process model. The standard process model defines the essential elements that should be integrated into any process definition. On the other hand, the refined process model is derived from the analysis of both the standard process model and event logs.

To illustrate the difference between single-point and double-point timestamps, we refer to Figure 1. The event log, denoted as $EL(A)$, presents two sample traces consisting of five activities along with their respective execution durations. By observing the traces, we can easily identify that in Trace 1, activities "a" and "b" should be in a parallel relation, as well as activities "c" and "d". Consequently, activity "a" is followed by activities "c" and "d", while activity "b" is followed by activity "d", and activities "c" and "d" converge to activity "e". In Figure 1(A), a refined process model using single-point timestamps (only the Finish event type) is depicted. This model fails to capture the parallel relation



EL(A)

Trace	Activity	Start Time	Finish Time
Trace 1	a	01/01/23 10.00	10/01/23 12.00
	b	05/01/23 10.00	15/01/23 14.00
	c	11/01/23 13.00	20/01/23 15.00
	d	14/01/23 15.00	17/01/23 16.00
	e	25/01/23 16.00	31/01/23 17.00
Trace 2	b	03/01/23 13.00	15/01/23 14.00
	c	08/01/23 18.00	18/01/23 12.00
	e	23/01/23 19.00	29/01/23 23.00

A
a (Finish)
b (Finish)
d (Finish)
c (Finish)
e (Finish)
Refined Process Model discovered from EL(A) in BPMN 2.0 (only Finish event type)
B
a
c
b
d
e
Refined Process Model discovered from EL(A) in BPMN 2.0 (both Start and Finish event types)

Figure 1. Sample of event log and process models

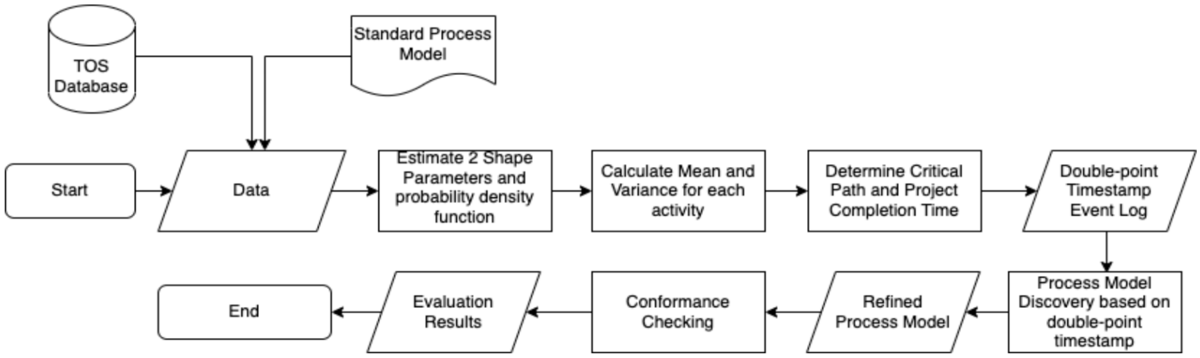
between activities "a" and "b", as well as between activities "c" and "d", which is evident in $EL(A)$. However, when employing double-point timestamps, as illustrated in Figure 1(B), the refined process model effectively represents the parallel disjunctive state of activities "c" and "d". This is observed through traces where activity "a" follows activity "c" without the occurrence of activity "d" within the same trace. Furthermore, this model simplifies the process representation by assigning a single activity for two events.

2.2. **Beta distribution and PERT.** The Beta distribution, defined on the interval $[0, 1]$ and typically denoted by α and β, is extensively utilized across various disciplines to analyze the behavior of random variables with finite intervals [20]. Its probability density functions, cumulative density functions, moment generating functions, and expectations and variances are explained in [7,20].

The Beta distribution can be found in numerous applications, including Bayesian hypothesis testing, the rule of succession, task duration modeling, and project planning control systems such as the Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT) [7,21]. In PERT, all activities are considered independent random variables with beta distributions, parameterized by three time estimates: the optimistic time (a), the pessimistic time (b), and the most likely time (m) [7,21].

3. **Proposed Methodology.**

In this section, we present our approach, encompassing the definition of double-point timestamps, process model discovery, and conformance checking as an evaluation of the refined process model. An overview of the proposed methodology is depicted in Figure 2.



TOS Database
Standard Process Model
Start
Data
Estimate 2 Shape Parameters and probability density function
Calculate Mean and Variance for each activity
Determine Critical Path and Project Completion Time
Double-point Timestamp Event Log
Process Model Discovery based on double-point timestamp
Refined Process Model
Conformance Checking
Evaluation Results
End

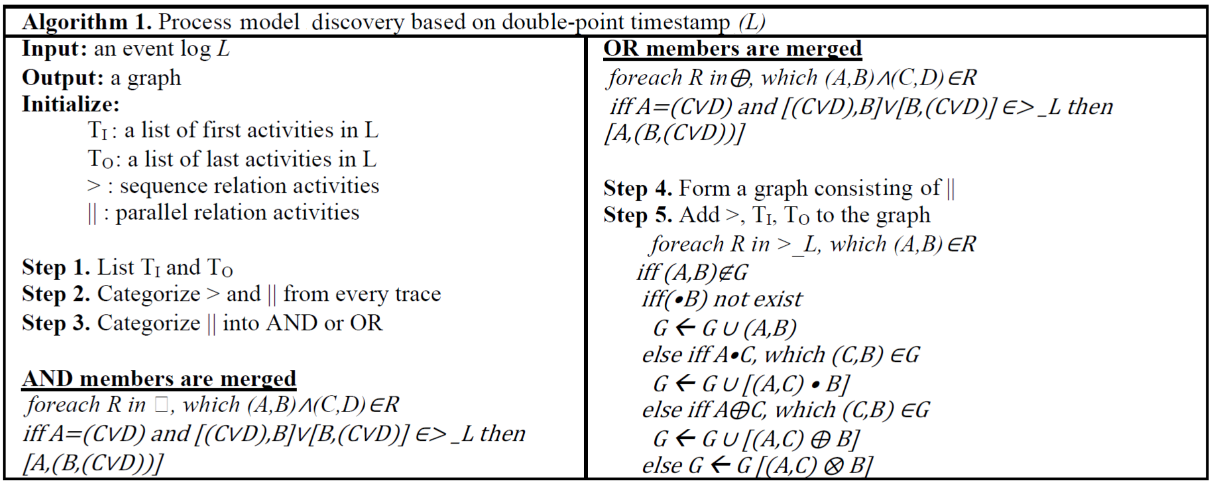
Figure 2. Proposed methodology

3.1. **PERT-Beta Distribution.** The PERT-Beta Distribution is employed to define double-point timestamps for each activity in the event log, specifically when the organization's data contains only timestamps for grouped activities. The methodology involves the following steps.

1) Calculate the optimistic time (a) which represents the fastest execution time, most likely time (m) which denotes the average execution time of an activity, and pessimistic time (b) which indicates the longest execution time for each activity in the event log. The estimated execution time of activity A is obtained by subtracting the single-point timestamp of activity B from activity A. The process optimization is carried out using Lingo tools.
2) Determine the shape parameters of the beta distribution, alpha (α) and beta (β), the expected time duration (mean), variance, probability density function, and cumulative probability function [9] for each activity in every case ID.
3) Calculate the Earliest Start Time (ES) and Earliest Finish Time (EF) based on the forward pass technique. Similarly, determine the Latest Finish Time (LF) and Latest Start Time (LS) using the backward pass technique. Additionally, calculate the Total Float (TF) or slack [10] for each activity, as well as identify the critical path for each case ID. We also need to determine the minimum and maximum project time duration for each possible relationship, including symmetries and asymmetries (left skewed or right skewed).
4) Calculate the overall project completion time for all case IDs in the event log.

3.2. **Process model discovery.** After obtaining the double-point timestamp event log using the PERT-Beta Distribution, the process model discovery phase is initiated. Process model discoveries based on activity lifespan have been introduced by [22] and [23]. A series of definitions are used to develop an algorithm that describes how activities are related in an event log, e.g., before, meets, overlaps, contains, equals, and has the same start time, and has the same end time [11]. However, there are no distinctions between OR and AND in the original algorithm. In this study, we modify the algorithm to distinguish between OR and AND relations (shown in steps 3-5).

The modified algorithm is depicted in Figure 3. The process involves several stages. Initially, it consists of compiling a list of all input and output activities for each trace. Following that, it entails categorizing the sequence and parallel relationships. It concludes with the presentation of the comprehensive activity relationships within the event log.



```
Algorithm 1. Process model discovery based on double-point timestamp (L)

Input: an event log L
Output: a graph
Initialize:
        T_I : a list of first activities in L
        T_O: a list of last activities in L
        > : sequence relation activities
        || : parallel relation activities

Step 1. List T_I and T_O
Step 2. Categorize > and || from every trace
Step 3. Categorize || into AND or OR

AND members are merged
 foreach R in □, which (A,B)∧(C,D)∈R
iff A=(C∨D) and [(C∨D),B]∨[B,(C∨D)] ∈>_L then
[A,(B,(C∨D))]

OR members are merged
 foreach R in ⊕, which (A,B)∧(C,D)∈R
  iff A=(C∨D) and [(C∨D),B]∨[B,(C∨D)] ∈>_L then
[A,(B,(C∨D))]

Step 4. Form a graph consisting of ||
Step 5. Add >, T_I, T_O to the graph
      foreach R in >_L, which (A,B)∈R
    iff (A,B)∉G
     iff(•B) not exist
      G ← G ∪ (A,B)
     else iff A•C, which (C,B) ∈G
      G ← G ∪ [(A,C) • B]
     else iff A⊕C, which (C,B) ∈G
      G ← G ∪ [(A,C) ⊕ B]
     else G ← G [(A,C) ⊗ B]
```

Figure 3. Process model discovery algorithm

3.3. **Conformance checking.** To validate the proposed methodology, three types of scoring are used to measure fitness [8] with Equation (1), precision (behavioral appropriateness) [3] with Equation (2), and structural appropriateness [3] with Equation (3) between the event log and the refined process model. We follow the modification of the appropriateness concept as introduced by [22,24], in accordance with the double-point timestamp concept.

$$Q_f = \frac{casesCaptured}{casesLog} \tag{1}$$

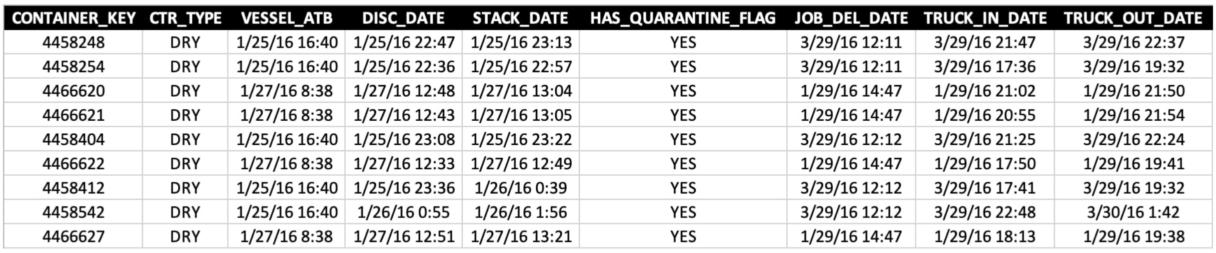
$$b(G, L) = \frac{| \rightarrow_G \cap \rightarrow_L | + |\|_G \cap \|_L|}{| \rightarrow_G \cup \rightarrow_L | + |\|_G \cup \|_L| + 1} \tag{2}$$

$$s(G, L) = \frac{|N_L| + 2}{|N_G|} \tag{3}$$

where Q_f: fitness value; *casesCaptured*: cases parsed in the process model from the event log; *casesLog*: amount of cases recorded in the event log; G: a process model; L: event log; $\rightarrow_G$ and $\rightarrow_L$: sequence relation; $\|_G$ and $\|_L$: parallel relation; N_L: a collection of activity mappings within L; N_G: A grouping of nodes in G (which may include activities and gateways). For the purpose of incorporating artificial *Start* and *End* nodes, a factor of 2 is added.

4. Results and Analysis.

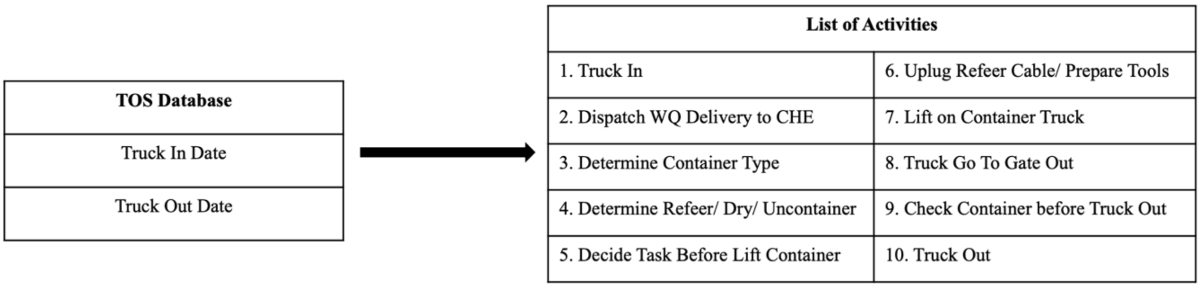
4.1. **Obtaining double-point timestamp using PERT-Beta Distribution.** To demonstrate the effectiveness of our proposed methodology, we conducted experiments using real-world data obtained from the Terminal Operating Systems (TOS) Database of Surabaya International Port Container Terminal. The TOS Database is an information system that generates a CSV file containing records of all processes executed in the port container terminal. The data used in our study were collected from January to March 2015. The TOS Database includes various attributes such as Container Key (CONTAINER_KEY), Container Type (CTR_TYPE), Vessel Berthing Process (VESSEL_ATB), Discard Date (DISC_DATE), Stack Date (STACK_DATE), Quarantine Process (HAS_QUARANTINE_FLAG), Job Delivery Date (JOB_DEL_DATE), Truck In Date (TRUCK_IN_DATE), and Truck Out Date (TRUCK_OUT_DATE), as depicted in Figure 4. We identified there are 12 traces and a total of 159,056 cases in the port container terminal.



CONTAINER_KEY	CTR_TYPE	VESSEL_ATB	DISC_DATE	STACK_DATE	HAS_QUARANTINE_FLAG	JOB_DEL_DATE	TRUCK_IN_DATE	TRUCK_OUT_DATE
4458248	DRY	1/25/16 16:40	1/25/16 22:47	1/25/16 23:13	YES	3/29/16 12:11	3/29/16 21:47	3/29/16 22:37
4458254	DRY	1/25/16 16:40	1/25/16 22:36	1/25/16 22:57	YES	3/29/16 12:11	3/29/16 17:36	3/29/16 19:32
4466620	DRY	1/27/16 8:38	1/27/16 12:48	1/27/16 13:04	YES	1/29/16 14:47	1/29/16 21:02	1/29/16 21:50
4466621	DRY	1/27/16 8:38	1/27/16 12:43	1/27/16 13:05	YES	1/29/16 14:47	1/29/16 20:55	1/29/16 21:54
4458404	DRY	1/25/16 16:40	1/25/16 23:08	1/25/16 23:22	YES	3/29/16 12:12	3/29/16 21:25	3/29/16 22:24
4466622	DRY	1/27/16 8:38	1/27/16 12:33	1/27/16 12:49	YES	1/29/16 14:47	1/29/16 17:50	1/29/16 19:41
4458412	DRY	1/25/16 16:40	1/25/16 23:36	1/26/16 0:39	YES	3/29/16 12:12	3/29/16 17:41	3/29/16 19:32
4458542	DRY	1/25/16 16:40	1/26/16 0:55	1/26/16 1:56	YES	3/29/16 12:12	3/29/16 22:48	3/30/16 1:42
4466627	DRY	1/27/16 8:38	1/27/16 12:51	1/27/16 13:21	YES	1/29/16 14:47	1/29/16 18:13	1/29/16 19:38

FIGURE 4. Terminal Operating Systems (TOS) Database

Initially, activity groups were stored in the TOS Database; therefore, we needed to obtain the detailed activity lists from experts at the Surabaya International Port Container Terminal. An example of activity groups and the detailed activity lists can be seen in Figure 5. To transform the TOS Database, which is in the form of a CSV file, into an event log containing Case ID, Activity, and Timestamp, we applied our proposed method using the PERT-Beta Distribution. We followed the steps outlined in Section 3.1. The calculation results of the PERT-Beta Distribution are presented in Table 1 and Table 2.



TOS Database

Truck In Date

Truck Out Date

List of Activities	
1. Truck In	6. Uplug Refeer Cable/ Prepare Tools
2. Dispatch WQ Delivery to CHE	7. Lift on Container Truck
3. Determine Container Type	8. Truck Go To Gate Out
4. Determine Refeer/ Dry/ Uncontainer	9. Check Container before Truck Out
5. Decide Task Before Lift Container	10. Truck Out

Figure 5. Example of activity groups and detailed activity lists in our case study

Table 1. Calculation results of PERT-Beta Distribution

Case ID	Activity code	a	m	b	Alpha shape	Beta shape	Mean	Variance	$f(x)$	$Beta(\alpha,\beta)$	Skewness
4453526	A	17	30	54	3.065	4.581	31.833	38.028	0.059	0.011	RIGHT
4453526	B	14	28	48	3.474	4.401	29.000	32.111	0.064	0.008	RIGHT
4453526	C	5	18	36	3.523	4.372	18.833	26.694	0.071	0.008	RIGHT
4453526	D	14	29	57	3.048	4.587	31.167	51.361	0.051	0.011	RIGHT
4453526	E	4	15	31	3.446	4.417	15.833	20.250	0.081	0.008	RIGHT
4453526	F	3	16	27	4.208	3.765	15.667	16.000	0.091	0.007	LEFT
4453526	G	3	7	14	3.151	4.551	7.500	3.361	0.200	0.010	RIGHT
4453526	H	9	14	24	2.938	4.617	14.833	6.250	0.147	0.012	RIGHT
4453526	I	2	6	11	3.681	4.270	6.167	2.250	0.243	0.008	RIGHT
4453526	J	2	9	14	4.383	3.506	8.667	4.000	0.183	0.008	LEFT
4453526	K	3	13	21	4.270	3.681	12.667	9.000	0.122	0.008	LEFT
4453526	L	4	6	11	2.595	4.671	6.500	1.361	0.318	0.018	RIGHT
4453526	M	1	2	4	2.938	4.617	2.167	0.250	0.735	0.012	RIGHT
4453526	N	5	9	14	3.681	4.270	9.167	2.250	0.243	0.008	RIGHT
4453526	O	7	15	27	3.397	4.443	15.667	11.111	0.110	0.009	RIGHT
4453526	P	7	21	62	2.367	4.669	25.500	84.028	0.041	0.023	RIGHT
4453526	Q	6	13	19	4.193	3.784	12.833	4.694	0.168	0.007	LEFT

Activity name: Document Entry via PDE (A), Vessel Berthing Process (B), Discharge Container (C), Bring Container to Yard (D), Stack Container in Yard (E), Verification Document Behandle (F), Create Document SPPB (G), Create Job Order Document Delivery (H), Truck in (I), Dispatch WQ Delivery to CHE (J), Determine Container Type (K), Determining Dry (L), Decide Task before Lift Container (M), Lift on Container Truck (N), Truck Go to Gate Out (O), Check Container before Truck Out (P), Truck Out (Q)

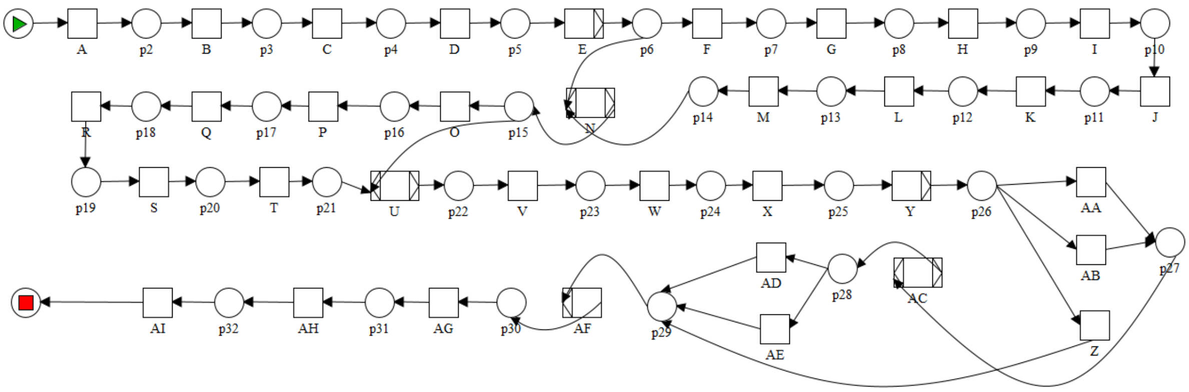
Table 1 displays the values of alpha (α) and beta (β) for each activity, obtained after calculating the optimistic time (the shortest time), most likely time (the most probable time, between the optimistic and pessimistic times), and pessimistic time (the longest time). We also derived the mean, variance, probability density function, and beta function for each activity. Since our case study exhibits asymmetry ($m \neq (a+b)/2$), we defined the skewness, as shown in Table 1. Table 2 presents the results of the forward pass, backward pass, critical path, and total float (slack) for each activity. For case ID 4453526, the average time of the case completion time was calculated as 264 hours, with a variance of 506.033 hours, and standard deviation of 22.495 hours. As the final step of the PERT-Beta Distribution, we obtained the double-point timestamps for each case ID in the event log, indicating the start and finish times for each activity, as shown in Table 2. Our proposed method enabled the definition of double-point timestamps for each activity in the event log, even when only grouped activity timestamps were available in the CSV database.

Table 2. Forward pass, backward pass, and double-point timestamp

Case ID	Activity code	ES	EF	LS	LF	Critical path	Slack	Start time	Finish time
4453526	A	0.000	31.833	0.000	31.833	YES	0	01/01/16 07:22	02/01/16 15:54
4453526	B	31.833	60.833	31.833	60.833	YES	0	02/01/16 15:54	03/01/16 21:23
4453526	C	60.833	79.667	60.833	79.667	YES	0	03/01/16 21:23	04/01/16 16:42
4453526	D	79.667	110.833	79.667	110.833	YES	0	04/01/16 16:42	06/01/16 01:14
4453526	E	110.833	126.667	110.833	126.667	YES	0	06/01/16 01:14	06/01/16 17:30
4453526	F	126.667	142.333	126.667	142.333	YES	0	06/01/16 17:30	07/01/16 09:33
4453526	G	142.333	149.833	142.333	149.833	YES	0	07/01/16 09:33	07/01/16 17:33
4453526	H	149.833	164.667	149.833	164.667	YES	0	07/01/16 17:33	08/01/16 08:33
4453526	I	164.667	170.833	164.667	170.833	YES	0	08/01/16 08:33	08/01/16 15:33
4453526	J	170.833	179.500	170.833	179.500	YES	0	08/01/16 15:33	09/01/16 00:33
4453526	K	179.500	192.167	179.500	192.167	YES	0	09/01/16 00:33	09/01/16 13:33
4453526	L	192.167	198.667	192.167	198.667	YES	0	09/01/16 13:33	09/01/16 20:33
4453526	M	198.667	200.833	198.667	200.833	YES	0	09/01/16 20:33	09/01/16 23:33
4453526	N	200.833	210.000	200.833	210.000	YES	0	09/01/16 23:33	10/01/16 09:33
4453526	O	210.000	225.667	210.000	225.667	YES	0	10/01/16 09:33	11/01/16 01:33
4453526	P	225.667	251.167	225.667	251.167	YES	0	11/01/16 01:33	12/01/16 03:33
4453526	Q	251.167	264.000	251.167	264.000	YES	0	12/01/16 03:33	12/01/16 16:33

4.2. **Process model discovery and quality evaluation.** We proceeded with process model discovery to uncover a refined process model based on the obtained double-point timestamps, which accurately reflects the business process model of the port container terminal. We executed the algorithm presented in Figure 3.

Following the execution, the refined process model is presented using the Petri Nets notation in YAWL for all 159,056 cases of the event log, as depicted in Figure 6. Our case study's refined process model incorporates both sequential and XOR parallel relations, showcasing the predetermined activity order and parallel execution branches, respectively.



Activity Label: Document Entry via PDE (A), Vessel Berthing Process (B), Discharge Container (C), Bring Container to Yard (D), Stack Container in Yard (E), Verification Document Quarantine (F), Create Job Order Document Quarantine (G), Bring Container from Yard to Quarantine (H), Stack Container in Quarantine Area (I), Check Goods Quarantine (J), Create Document KH/KT (K), Send Certificate KH/KT Info (L), Stack Container in Yard from Quarantine (M), Verification Document Behandle (N), Create Job Order Document Behandle (O), Bring Container from Yard to Behandle (P), Stack Container in Behandle Area (Q), Check Goods Behandle (R), Create Document LHP (S), Stack Container in Yard from Behandle (T), Create Document SPPB (U), Create Job Order Document Delivery (V), Truck In (W), Dispatch WQ Delivery to CHE (X), Determine Container Type (Y), Determining Dry (Z), Determining Refeer (AA), Determining Uncontainer (AB), Decide Task before Lift Container (AC), Prepare Tools (AD), Unplug Refeer Cable (AE), Lift on Container Truck (AF), Truck Go to Gate Out (AG), Check Container before Truck Out (AH), Truck Out (AI)

Figure 6. Refined process model

Subsequently, the proposed methodology is rigorously validated through the quantification of fitness, precision, and structural appropriateness, which assess the congruence between the event log and the refined process model using Equations (1), (2), and (3). The outcomes of this conformance checking process are summarized in Table 3. Based on these results, we assert with confidence that the proposed method effectively identifies a process model that faithfully represents the intricacies of the event log.

Table 3. Results of conformance checking

Conformance checking	**Figure 6 (Refined Process Model)**
Fitness	0.894
Precision (Behavioral Appropriateness)	0.917
Structural Appropriateness	0.925

To sum up, our approach focuses on introducing an alternative methodology that discovers process models by considering an activity and both its start and finish duration by applying PERT-Beta Distribution to determine double-point timestamps event log from single timestamp event log of a company's database. Therefore, based on our experiments, we can identify certain drawbacks associated with single-point timestamps as opposed to double-point timestamps, particularly concerning the number of traces employed and the differentiation of parallel gateways in the context of a model that includes parallel processes.

5. **Conclusions.** The transformation of an organization's conventional database into a standardized event log holds significant importance within the realm of process mining, as it serves as the initial step in a comprehensive series of processes aimed at analyzing a refined process model that accurately reflects a standard process. In this paper, we have presented a unique approach that employs the PERT-Beta Distribution to generate a double-point timestamp event log. While this method focuses solely on the time required for executing a single case in a process, encompassing optimistic, most likely, and pessimistic times, it does not account for other factors such as crash time, normal cost, or crash costs associated with individual cases within the process. Furthermore, utilizing the double-point timestamp, we have successfully derived a refined process model. The conformance checking results of our case study, based on real-world data from a port container terminal, demonstrate that our proposed methodology exhibits favorable fitness, precision, and structural appropriateness. As we look ahead, our future work will involve incorporating dispatching rules specific to port container terminals, addressing the demands of Industry 4.0, which promotes enhanced flexibility and dynamism within production systems.

Acknowledgment. This research was supported by the BK21 FOUR of the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 5199990914 451).

REFERENCES

[1] K. Okoye, S. Islam, U. Naeem and M. S. Sharif, Semantic-based process mining technique for annotation and modelling of domain process, *International Journal of Innovative Computing, Information and Control*, vol.16, no.3, pp.899-921, 2020.

[2] N. I. Utama, I. M. Kamal, A. N. Iman, D. Kim, H. Bae and C. Hong, Colored Petri Nets simulation on the cloud with python: A framework overview, *ICIC Express Letter, Part B: Applications*, vol.13, no.7, pp.725-733, 2022.

[3] W. M. P. van der Aalst, *Process Mining: Data Science in Action*, 2nd Edition, Springer, Heidelberg, DOI: 10.1007/978-3-662-49851-4, 2016.

[4] P. Zerbino, A. Stefanini and D. Aloini, Process science in action: A literature review on process mining in business management, *Technological Forecasting and Social Change*, vol.172, DOI: 10.1016/j.techfore.2021.121021, 2021.

[5] Y. A. Effendi and F. Retrialisca, Transforming timestamp of a CSV database to an event log, *2021 8th International Conference on Computer and Communication Engineering (ICCCE)*, pp.367-372, DOI: 10.1109/ICCCE50029.2021.9467226, 2021.

[6] 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, DOI: 10.1109/IC3INA.2015.7377741, 2015.

[7] E. F. Udoumoh and D. W. Ebong, A review of activity time distributions in risk analysis, *American Journal of Operations Research*, vol.7, pp.356-371, DOI: 10.4236/ajor.2017.76027, 2017.

[8] Y. A. Effendi, F. Retrialisca and N. Nuzulita, Conformance checking to evaluate business process models using modified time-based heuristics miner algorithm, *2019 IEEE Conference on Sustainable Utilization and Development in Engineering and Technologies (CSUDET)*, pp.81-86, DOI: 10.1109/CSUDET47057.2019.9214767, 2019.

[9] H. Yeo, Natural language interface for process mining queries in healthcare, *2022 IEEE International Conference on Big Data (Big Data)*, pp.4443-4452, DOI: 10.1109/BigData55660.2022.10020685, 2022.

[10] L. Yang, G. Kang and L. Zhang, Medical treatment process modeling based on process mining and treatment patterns, *China Communications*, vol.18, no.12, pp.332-349, DOI: 10.23919/JCC.2021.12.021, 2021.

[11] A. Hakim, Improving healthcare systems through process mining, *IEEE 23rd International Multitopic Conference (INMIC)*, pp.1-4, DOI: 10.1109/INMIC50486.2020.9318159, 2020.

[12] G. Schuh, A. Gützlaff, S. Cremer, S. Schmitz and A. Ayati, A data model to apply process mining in end-to-end order processing processes of manufacturing companies, *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp.151-155, DOI: 10.1109/IEEM45057.2020.9309946, 2020.

[13] M. J. Sangil, Heuristics-based process mining on extracted Philippine public procurement event logs, *The 7th International Conference on Behavioural and Social Computing (BESC)*, pp.1-4, DOI: 10.1109/BESC51023.2020.9348306, 2020.

[14] H. Magnus, J. Fialho, C. Wanzeller and J. Silva, Process mining – Case study in an industrial assembly process in the shopfloor, *The 16th Iberian Conference on Information Systems and Technologies (CISTI)*, pp.1-8, DOI: 10.23919/CISTI52073.2021.9476389, 2021.

[15] T. Chervenkov, S. Pavlov, D. Marinov, H. Hristov and G. Momcheva, Application of process mining approach to the developmental process of the roundworm C. elegans, *International Conference on Biomedical Innovations and Applications (BIA)*, pp.51-53, DOI: 10.1109/BIA52594.2022.9831332, 2022.

[16] S. Simone, K. Oliver, M. Elisabeth and S. Andreas, Using an event hierarchy for improved process mining of website user behaviour, *The 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pp.1-6, DOI: 10.1109/ICECCME57830.2023.10253047, 2023.

[17] B. Kamala and B. Latha, Process mining and deep neural network approach for the prediction of business process outcome, *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, pp.1-4, DOI: 10.1109/IC3IOT53935.2022.9767941, 2022.

[18] A. Batyuk and V. Voityshyn, Streaming process discovery method for semi-structured business processes, *2020 IEEE 3rd International Conference on Data Stream Mining & Processing (DSMP)*, pp.444-448, DOI: 10.1109/DSMP47368.2020.9204201, 2020.

[19] R. B. Bahaweres, H. Amna and D. Nurnaningsih, Improving purchase to pay process efficiency with RPA using fuzzy miner algorithm in process mining, *International Conference on Decision Aid Sciences and Applications (DASA)*, pp.1483-1488, DOI: 10.1109/DASA54658.2022.9765091, 2022.

[20] S. U. Chidawa and A. D. Mambo, A reputation model for aggregating rating levels using beta distribution, *2021 1st International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS)*, pp.1-6, DOI: 10.1109/ICMEAS52683.2021.9692422, 2021.

[21] K. S. Rao, N. Viswam and G. V. S. R. Anjaneyulu, PERT distribution and its properties, *International Research Journal of Modernization in Engineering Technology and Science*, vol.3, no.10, pp.907-914, 2021.

[22] 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*, pp.137-156, 2014.

[23] R. Sarno, W. A. Wibowo, Kartini, F. Haryadita, Y. Amelia and K. Rossa, Determining model using non-linear heuristics miner and control-flow pattern, *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol.14, no.1, DOI: 10.12928/telkomnika.v14i1.3257, 2016.

[24] J. C. A. M. Buijs, B. F. van Dongen and W. M. P. van der Aalst, On the role of fitness, precision, generalization and simplicity in process discovery, *OTM Federated Conferences, 20th International Conference on Cooperative Information Systems (CoopIS 2012)*, vol.7565, pp.302-322, 2012.