2023 International Conference on Technology, Engineering, and Computing Applications (ICTECA) | 979-8-3503-6055-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICTECA60133.2023.10490829

# A Novel Deterministic Algorithm for Optimizing Workflow Discovery with Short-Loops in Large Event-Log

Hermawan Bin Fauzan
Department of Informatics Universitas Trunojoyo Madura, Indonesia
Doctoral Student at Department of InformaticsInstitut Teknologi Sepuluh Nopember, Indonesia. Surabaya, Indonesia
hermawan@trunojoyo.ac.id

Ahmad Saikhu
Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
saikhu@if.its.ac.id

Riyanarto Sarno
Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember Surabaya, Indonesia
riyanarto@if.its.ac.id

Fika Hastarista Rachman
Department of Informatics, Faculty of Engineering, Universitas Trunojoyo Madura, Indonesia
fika.rachman@trunojoyo.ac.id

***Abstract*— One of the main problems in process discovery is obtaining correctness workflow when analysing a Large Event-Log with high concurrent control flow, complex short loops, and uncertainty. $\propto^{T+}$ is a novel method to construct a standardization workflow model in the SWF-Net for discovering a large event log with high concurrency free-choice, parallel, and non-free-choice control flow within an uncertainty short-loop. By the testing result, the short-loop construction places with the optimistic and pessimistic One-Loop-Free approach have the best results compared to other existing methods based on deterministic, heuristic, genetic, inductive, also Integer Linear Programming algorithms. $\propto^{T+}$ has proofed achieved completeness and correctness SWF-Net with high rate on fitness recall up to 95% and appropriateness up to 90%.**

***Keywords—component, formatting, style, styling, insert*** *(key words)*

## I. Introduction

Business Intelligence (BI) is one of the main critical success factors for an organization to align its business strategies with its vision [1]. Process mining as an implementation method in business intelligence is becoming a complement of data mining to extract knowledge [2] and improvement process model as a piece of valuable information that has a benefit for helping the business analyst to manage a complex system [3]. Process mining has the developing stage for extracting, discovering, monitoring, and improving analysis of the event log from data loggers source produced by information system's operation, especially an Enterprise Resource Planning (ERP) system. Therefore, Process mining accomplishes within Process-Aware Information Systems (PAISs) [4].

Business Intelligence (BI) is one of the main critical success factors for an organization to align its business strategies with its vision [1]. Process mining as an implementation method in business intelligence is becoming a complement of data mining to extract knowledge [2] and improvement process model as a piece of valuable information that has a benefit for helping the business analyst to manage a complex system [3]. Process mining has the developing stage for extracting, discovering, monitoring, and improving analysis of the event log from data loggers source produced by information system's operation, especially an Enterprise Resource Planning (ERP) system. Therefore, Process mining accomplishes within Process-Aware Information Systems (PAISs) [4].

As the foremost step in process mining, the process discovery constructs a workflow graph model from the task set relation contained in the trace event log as the process mining's primary goal. Today various discovery methods exist in many process mining tools such as Prom, Disco, Aris, Yet Another Workflow Language (YAWL), Colored Petri Net (CPN), Business Process Modeling Notation (BPMN), and others [5]. Although the various algorithms in these various tools have undergone various improvements, however improvements are still needed when process mining discovery implemented on the real information systems such as the ERP, healthcare, supply chain, financial, and other complex systems which has a large event log traces, consist of many activities event variants, and generally accompanied with a complicated process [2].

By the examination, several algorithms that developed in PROM includes: deterministic α miner [6] that advanced with $\alpha_+$ [7], $\alpha_{++}$ [8], $\alpha_{\#}$[9]; heuristic miner [10], [11] genetic miner [12] with advanced by Evolutionary Tree Miner (ETM) [13], Inductive miner[14], Integer Linear Programming [15] have not achieved completeness and correctness when executing discovery in the large event log collected by BPI Challenge such as Hospital Billing, Financial Loan, and others which are the real live event log [16]. Without fulfilling completeness and correctness, process discovery result does not reflect the real operational activities from the business transactions captured by the event log [17].

Through the discovery testing use deterministic such as $\alpha$, $\alpha_+$, and $\alpha_{++}$ in the PROM, they have many failures. The resulting workflow discovery produces many dead tasks and deadlocks [18] The many failures result caused by the various limitations of the discovery algorithm on executing real large event logs with contained complex and uncertainty short-loop. For the reason, this research proposes the $\propto^{T+}$ algorithm based on the deterministic α as the basis for the short-loop discovery. $\propto^{T+}$ has a specification for constructing short loop control flow through a partition multi-loop concurrency in the optimistic and pessimistic -◊-free patterns as the main contribution from the method.

These are many contributions of $\propto^{T+}$. First, At the pre-discovery stage, $\propto^{T+}$ simplifies the footprint features as many independent tuples relation by constructing the generalizations of input, output, choice, parallel, two-loop, one-loop, and activities weight within the *Adjacency List Tree* (ALT) structure [19], it reduces collision between different short-loop pattern on one-loop (◊) and two-loop (Δ). Second, the discovery stage $\propto^{T+}$ applies the elimination and fusion rules between the input and output tuples for reconstructing the membership of set places of -◊-free. Applying the optimistic and pessimistic -◊-free patterns to reduce under-fitting generalization on control flow that caused ambiguity in SWF-Net [18].

## II. METHOD

### A. *Algorithm Comparison*

Within the novel algorithm development, why is the classical deterministic still required for process discovery although the latest advanced algorithms already exist [11, 12, 13, 14, 15, 16], where in certain aspects such as robustness and speed it has better performance than deterministic [20]. For fulfilling the standardization of control flow, deterministic is still appropriate because it is specified to describe the entire activity traces without simplifying the causality logic pattern so that the workflow discovery resulted in the real-life event log [21]. However, many strict patterns implemented by deterministic has a high risk of failure on dead task and dead lock also tend to producing overfitting control flow. On appropriateness comparison, workflow result that generated by heuristic, genetic, inductive, declarative, ILP, and artificial intelligence does not meet the soundness WF-Net (SWF-Net) based on the Petri Net where they have more relaxed soundness with risk unable to find any meaningful detail patterns within the event log [22].

As for the various algorithms also have model specifications. Heuristic builds a workflow network with a direct follow graph within the C-Net standardization [23], then genetic [15] and inductive [16, 23] miners use the process tree approach that cannot discover complex loops or situations multi-choice in parallel [24]. C-Net and Process Tree based on the direct follow graph, where it has advantage free from discovery failures. WF-Net especially produced by deterministic has a more mature model in fulfilling soundness rule than others [18]. Otherwise, with strict control flow rules in detail, SWF-Net's flexibility is lower than others, where it doesn't guarantee free from deadlocks and dead tasks [25]. Therefore, improving the deterministic algorithm through the advancing of new causality patterns makes it appropriate converted to others standardization business workflow models such as CPN, BPMN, YAWL [26]. So that, it has high scalabity to build as plugin within the existing tools.

### B. *Deterministic Miner Contribution*

Today many variants of deterministic algorithms developed to overcome the specific control flow to fulfil the SWF-Net control flow, as described in Table 1. The implementation of deterministic in many general cases with executing with the PROM has success in obtaining completeness and correctness workflow discovery, but with exceptions in the real even-log in many competition event logs supplied by BPI can't resolve the problems [27]. Our algorithm contribution in the case as shown as Table 1.

TABLE 1, THE SPECIFICATION OF DETERMINISTIC ALGORITHM

| Year | Algorithm | Contribution Pattern |
|---|---|---|
| 2003 | α | Serial, XOR, AND |
| 2004 | $\alpha_+$ | Short Loop |
| 2007 | $\alpha_{++}$ | Non-Free Choice |
| 2010 | $\alpha_{\#}$ | Invisible Choice |
| 2022 | $\propto^{T+}$ | Un-Safeness Loop |

### C. *Identify Deterministic Weakness*

To find the main contribution, we need to find the main deterministic weakness which can be represented with $\alpha_+$ and $\alpha_{++}$. By the testing identified, the construction of control flow pattern $\lozenge -free$ they have the weaknesses to perform generalization in safeness under two conditions:

- If the event log occurs contiguously ◊ in the input and output place like as $D_{\lozenge}\{E_{\lozenge}, F\}$, elimination $\lozenge -free$ output is not safeness deletion on $D_{\lozenge}$ so that is causing the dead task.
- If the event log has more than ◊ in single places with the same input share but different output shares such as $G\{H_{\lozenge}, I_{\lozenge}\} - H\{J, K\} - I\{M, N_{\lozenge}\}$, it made $\alpha_{++}$ construct $\lozenge -free$ aggregation on the two independent places: $G_1\{H_{\lozenge}, J, K\}$ and $G_2\{I_{\lozenge}, M, N_{\lozenge}\}$ where causing the ambiguous causality changes from XOR to AND. $\lozenge -free$ output is not safeness deletion on $N_{\lozenge}$ so that is producing the dead tasks.

### D. *Solving Method*

The deterministic discovery process aims to establish an independent relationship between place and task transitions so that the resulting workflow of model avoids ambiguity and redundancy. To meet the SWF-Net concurrency control flow, $\propto^{T+}$ eliminates parallel links, chooses the maximum path, and finishes with firing the tasks and places. To solve the multi-loop

Identify applicable funding agency here. If none, delete this text box.

concurrency problem, $\propto^{T+}$ determines a formula with 2 generalization terms, namely: optimistic and pessimistic -◊-*free*.

### *D.1. Optimistic -◊-free*

Generalization -◊-*free* is performed optimistically if the input place $t_{\Diamond} \in p_{I\approx\Diamond} \in T_I$ contains the causality of tuple $A \rightarrow B$ in the place $A\{B\}$, with the terms as described in (4).

1. If $\mathrm{Y_w}$ existed by single ◊ in the tuples sequence $\mathrm{A\{B_{\Diamond}\}\epsilon t_{\Diamond}, B_{\Diamond}\{C\}\epsilon t_{\Diamond} \Rightarrow p_{I\approx\Diamond}A\{B_{\approx\Diamond}, C\}}$ AND $\mathrm{B\{\}}$
2. If $\mathrm{Y_w}$ existed by multi ◊ in the same place: $\mathrm{A\{B_{\Diamond}, C_{\Diamond}\}\epsilon t_{\Diamond}, t_{out}\epsilon B \subseteq t_{out}\epsilon C \Rightarrow p_{I\approx\Diamond}A\{B_{\approx\Diamond}, C_{\approx\Diamond} \cup t_{out}\epsilon B \subseteq t_{out}\epsilon C\}}$
3. If $\mathrm{Y_w}$ existed by dual Δ in the same place: $\mathrm{A\{B_{\Delta}, C_{\Delta}\}\epsilon t_{\Delta}, B\Delta C, C\Delta B \Rightarrow p_{I\approx\Diamond}A\{B_{\approx\Diamond}, C_{\approx\Diamond} \cup t_{out}\epsilon B \subseteq t_{out}\epsilon C\}}$
4. If $\mathrm{Y_w}$ existed by ◊ and Δ in the same place: $\mathrm{A\{B, C\}\epsilon t_{\Diamond}, B\Delta C \Rightarrow p_{I\approx\Diamond}A\{B_{\approx\Diamond}\}, p_{I\approx\Diamond}B\{C_{\approx\Diamond} \cup t_{out}\epsilon B\}}$

   (4)

**Proof**:

The first, -◊-free generalization is executed if the ◊ predecessor has only one output or has only followed causality or $\mathrm{t_{out}\epsilon A \subseteq t_{out}\epsilon B_{\Diamond}}$, so implies $\mathrm{t_{out}\epsilon B_{\Diamond}}$ that can be composed directly into the -◊-free place $\mathrm{p_{I\approx\Diamond}A\{B_{\approx\Diamond} \cup t_{out}\epsilon B_{\approx\Diamond}\}}$. As for if $|\mathrm{t_{out}\epsilon A}| > 1$ and $\mathrm{t_{out}\epsilon A \notin t_{out}\epsilon B_{\Diamond}}$, it is not possible to generalize for the overall output $\mathrm{t_{out}\epsilon A \cup t_{out}\epsilon B}$, the generalization can be applied to the same output only if $\mathrm{t_{out}\epsilon A \subseteq t_{out}\epsilon B_{\approx\Diamond}}$ that implies $\mathrm{p_{I\approx\Diamond}A\{B_{\approx\Diamond}, \forall_{out}\}}$ with the composing output $\mathrm{\forall_{out} = t_{out}\epsilon A \wedge t_{out}B_{\approx\Diamond}}$. Within the provisions, -◊-free ensures the removal of all task outputs $\mathrm{t_{out}\epsilon B_{\approx\Diamond}}$ which are related with $\mathrm{P_I}$ $\mathrm{A\{B\} \Rightarrow t_{out}\epsilon B_{\Diamond} = \emptyset}$ and $\mathrm{P_O\ \{A\}B \Rightarrow t_{in}\epsilon B_{\Diamond} = \emptyset}$ is safeness as shown as Figure 1.

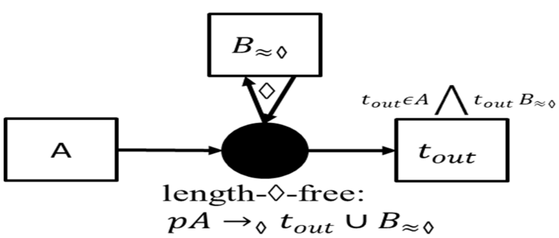


Fig. 1. The basic generalization of Optimistic -◊-free in $\propto^{T+}$

The second, -◊-free generalization is executed when many ◊ exist in the same output with termination as the first. For the three and four terms, with use dummy event log **L1** generalization, -◊-free can be executed in Δ if there is dual Δ or multi concurrency ◊ and Δ such as tuple causalities $\Delta\{..\mathrm{BCB}..,..\mathrm{CBC}..\}$ and $\Diamond\ \{..\mathrm{BB}..,..\mathrm{CC}..\}$ as the result shown in the Figure 2, also with **L2** single composition as $\Delta\{..\mathrm{BCB}..\}$ and $\Diamond\{..\mathrm{BB}.,..\mathrm{CC}..\}$ as shown resulting in Figure 3 has the same soundness when executing use $\propto^{\mathrm{T}+}$. Otherwise, -◊-free generalization has un-soundness when executed with $\boldsymbol{\alpha}_{+}$ and $\boldsymbol{\alpha}_{++}$ as shown as Figure 4 and Figure 5 because both doesn't have condition when $\mathrm{p_{I\approx\Diamond}}$ is constructed so it should search the input and output tuple $\mathrm{A \rightarrow_{\Diamond} B_{\approx\Diamond}}$ which not free from Δ to avoid removal because it's no safeness.

**L1**: ABBCBD, ACBCCD, ABCBD, ACBCD, ABD, ACD, AD

**L2**: ABBCBD, ABCBCD, ABCBD, ABBCCD, ABCD, ABD

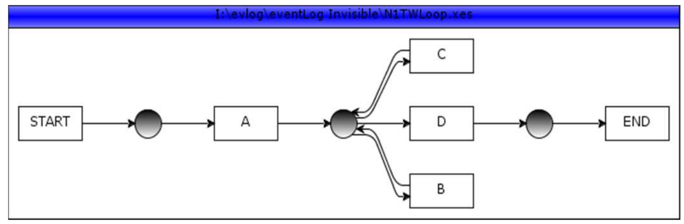


Fig. 2. Soundness -◊-free for Dual Δ by $\propto^{T+}$

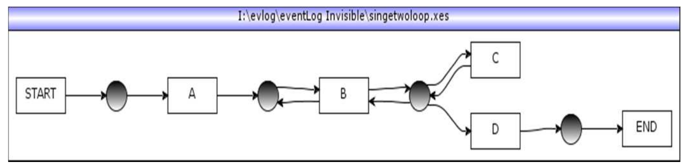


Fig. 3. Soundness -◊-free for Single Δ by $\propto^{T+}$

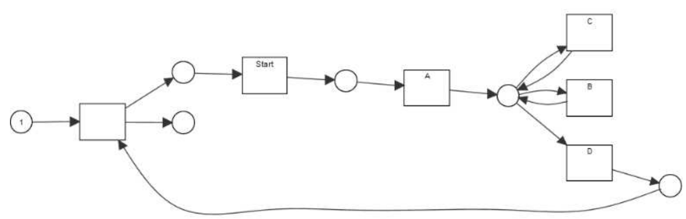


Fig. 4. Completeness Un-soundness -◊-free for Dual Δ L3 by $\boldsymbol{\alpha}_{++}$

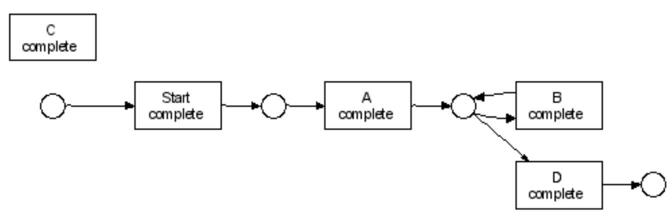


Fig. 5. Failed -◊-free for Singe Δ L4 by $\boldsymbol{\alpha}_{++}$

### *B. Pessimistic length-◊-free*

Pessimistic -◊-*free* is constructed if $T'_{\underline{\Diamond}}$ has different outputs, that lead to under-fitting by generalizing the entire output loop, and over-fitting which results in structural changes that cause ambiguity to maintain a soundness at SWF-Net. $\propto^{T+}$ performs an invisible -◊-*free* pattern in the invisible task loop $T'_{\underline{\Diamond}}$ which has functioned as a predecessor that insulates generalizations in both choice and parallel concurrencies as figured as Figure 6.

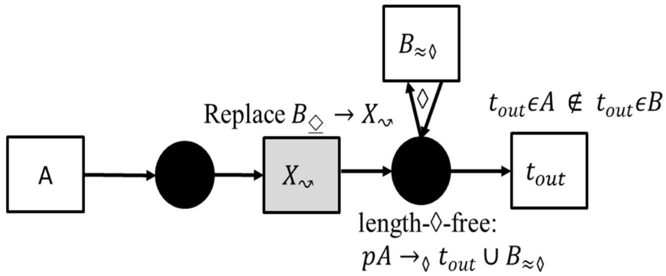


Fig. 6. Generalization on the Pessimistic length-◊-free $\propto^{T+}$

$T'_{\underline{\lozenge}}$ replaces task $B_{\approx\underline{\lozenge}}$ so that the composition of place $A \rightarrow_{\lozenge} B_{\approx\underline{\lozenge}}$ changes to an independent structure $p_{I_1} A \rightarrow T'_{\underline{\lozenge}}$ and $p_{I_2} T'_{\underline{\lozenge}} \rightarrow B_{\approx\underline{\lozenge}} \cup \{t_{out} \in B\}$. The removal of all task output relations at place $pB\{\}$ ensures safeness. Within the generalization of the output that only occurs at $p_2$, there is no need to change the composition of the output on the multi-concurrency XOR-loop. To maintain the AND place concurrency, the AND place ensures a single input-output is performing with the changes in composition. The composition of the place input is same as the generalization process on XOR, but within the output place $T_{out}$, a parallel place is checked, which is contain loops to add invisible task input so that the parallel independence relationship becomes $p_O\{T'_{\underline{\lozenge}}, B_{\approx\lozenge}\} \rightarrow t_{out}$ as constructed by Table 2.

Through the testing use $L_1$, $\propto^{T+}$ generates the addition of short-loop combination in the parallel input place. $\propto^{T+}$ capable to generalize multi-concurrency -◊-*free* on combination of Δ, XOR and AND to fulfilling the SWF-Net without modification control flow logic through the addition of an invisible-loop such as in the AND causality in tuples $D\{E', F'\}$, then for XOR in $G\{H', I'\}$ as shown in Figure 7. The result can be comparing with $\alpha_{++}$ which has failed discovery.

**$L_1$**: ABDDEFGIIM, ABDFEEGINN, ABDEEFGHHJL, ABDFEGHJL, ABDEFGHKL, ABCBDEFGIIM, ABCBDFEGINN, ACCBDEEFGHHJL

TABLE 2, THE SET OF PLACE INPUT OF INVISIBLE-LOOP

| footprint | | Values |
|---|---|---|
| $\forall_{\|\|}$ | : | {E, F} |
| $\forall_{\Delta}$ | : | {B, C} |
| $\forall_{\lozenge}$ | : | {B, C, D, E, H, I} |
| $T_{in}$ | : | A{B, C}, B{C, D}, C{B}, D{E, F}, E{F, G}, F{E, G}, G{H, I{M, N}, J{L}, K{L} |
| $T_{out}$ | : | B{A, C}, C{A, B}, D{B}, E{D, F}, F{D, E}, G{F}, H{}, I{} |
| $P_{\approx\lozenge}$ | : | {B, C, E, H, I} |
| $P_I$ | : | A{B, C, D′}, B{ }, C{ }, D{ }, $D'${D, E, G}, E{ }, F{ $H'${H, J, K}, $I'${I, M, N}, J{}, K{} |
| $P_O$ | : | B{}, C{}, D{}, E{}, F{D, $D'$}, G{F}, H{}, I{}, J{}, K{}, L{J, |

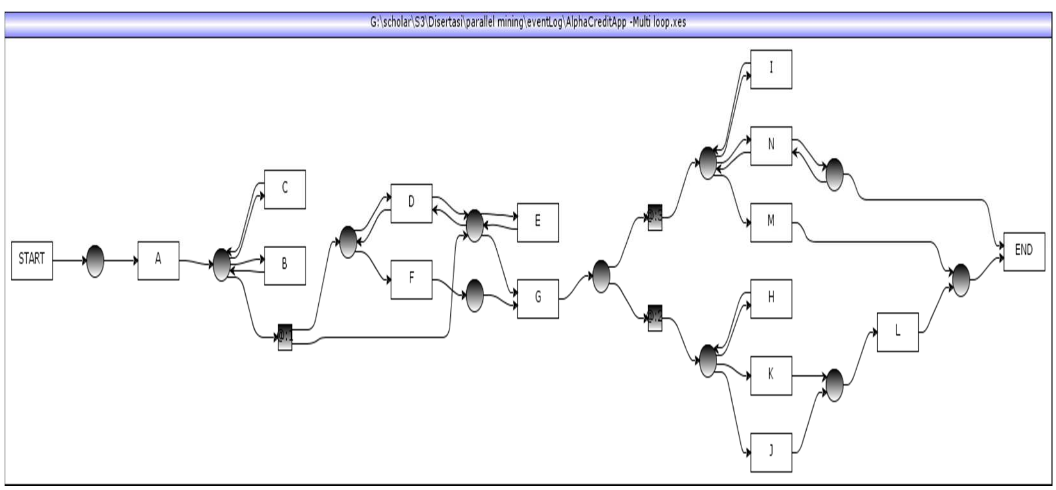


Fig. 7. , $\propto^{T+}$ Soundness Discovery for the contiguous -◊-free

As shown in Figure 7, $\propto^{T+}$ gets a soundness for the high complexity event-log which has multi concurrency on ◊, Δ, XOR, AND with the loop contiguous in the input and output. As the same case, $\alpha_{++}$ has failed discovery same as $\alpha_{+}$. Both has failure because the deletion -◊-*free* in $\alpha_{+}$ and $\alpha_{++}$ is not safeness if implemented on the contiguous loop that consist in the parallel and free-choice.

## III. RESULTS AND DISCUSSION

By the applying the methods and testing of various short-loop multi-concurrency patterns within ◊, Δ, XOR, also AND, $\propto^{T+}$ algorithm capable to perform ◊-free generalizations correctly to meet the Structured Workflow net (SWF-Net) soundness. On the other hand, when there is no short-loop concurrency, $\propto^{T+}$ is still able to maintain control-flow independently without being influenced by ◊-free generalizations. To measure the SWF-Net quality of discovery is based on three primary assessments: Correctness, Completeness dan Precision [25, 28].

Correctness is *soundness rules* that is addressed to fulfill the SWF-Net soundness such as rules:

1. For each $p \in \mathrm{P}$ and $t \in \mathrm{T}$ within $\{p, t\} \in \mathrm{F}$: $|p \bullet| > 1$ implies $|\bullet t| = 1$
2. For each $p \in \mathrm{P}$ and $t \in \mathrm{T}$ with $\{p, t\} \in \mathrm{F}$: $|t \bullet| > 1$ implies $|\bullet p| = 1$.
3. For each $p \in \mathrm{P}$ no implicit places, no need synchronization winner transition on concurrent relation within $\{p, t\}$.

Completeness is conformance checker that applied by calculate *fitness reply* $f_r$ as in (6) and also precission measured with appropriateness $a_B$ [27] as in (7).

$$\mathrm{f_r} = 0.6\left(1 - \frac{\sum_{i=1}^{k} n_i.m_i}{\sum_{i=1}^{k} n_i.c_i}\right) + 0.4\left(1 - \frac{\sum_{i=1}^{k} n_i.(r+a)_i}{\sum_{i=1}^{k} n_i.p_i}\right) \quad (6)$$

Where,

| | | |
|---|---|---|
| $\boldsymbol{n_i}$ | : | *set of instance logs trace* |
| $\boldsymbol{m_i}$ | : | *set of missing links* |
| $\boldsymbol{r+a}$ | : | *number of remaining Node (r) and remaining alternative link (a)* |
| $\boldsymbol{c_i}$ | : | *set of consumed tokens* |
| $\boldsymbol{p_i}$ | : | *set of produced tokens* |

Then behavioral appropriateness $a_B$,

$$a_B = \frac{\sum_{i=1}^{k} \mathbf{n}_i(|T_V| - x_i)}{\sum_{i=1}^{k} \mathbf{n}_i(|T_V| - 1)} \quad (7)$$

Where,

| | | |
|---|---|---|
| $\boldsymbol{k}$ | : | *number of different traces* |
| $\boldsymbol{i}$ | : | *number of log traces* |
| $\boldsymbol{n_i}$ | : | *set of process instances combined into current trace* |
| $\boldsymbol{x_i}$ | : | *set of enabled transitions* |
| $\boldsymbol{T_V}$ | : | *number of visible tasks* |

Table 3 shows the comparison of soundness, reply-fitness and precision appropriateness for $\propto^{T+}$ compared to $\alpha_+$ and $\alpha_{++}$ to discover the short-loop patterns. The comparisons are only made between deterministic because non-deterministic algorithms are not intended to construct the short-loop soundness which meets with the SWF-Net rule. Point 1 is showing a comparison of multi-loop Δ concurrency accompanied by uncertainty order sequences which can be discovered correctly by $\alpha_+$ and $\alpha_{++}$, but in point 2 $\alpha_+$ has un-soundness also $\alpha_{++}$ has failed because the algorithm limitations in generalizations ◊-free. Generalization $\propto^{T++}$ is success in discovering this pattern even though without apply non invisible pattern analysis.

TABLE 3, THE COMPARISON OF SHORT-LOOP DISCOVERY BETWEEN $\mathrm{A}_+$, $\mathrm{A}_{++}$, AND $\propto^{T+}$.

| Concurrency ◊-free | $\alpha_+$ S | F | P | $\alpha_{++}$ S | F | P | $\alpha_{T+}$ S | F | P |
|---|---|---|---|---|---|---|---|---|---|
| Dual-Δ Loop | V | 1 | 1 | v | 1 | 1 | v | 1 | 1 |
| Single-Δ Loop | V | 1 | 0.9 | x | 0.7 | 0.3 | v | 1 | 1 |
| Contiguous loop | X | 0.9 | 0.7 | x | 0.8 | 0.5 | v | 1 | 1 |
| NFC | X | 0.7 | 0.3 | v | 1 | 0.9 | v | 1 | 1 |
| Hospital Billing | X | 0.1 | 0.1 | x | 0.2 | 0.3 | v | 1 | 0.8 |
| BPI-Challenge 18 | X | 0.1 | 0.1 | x | 0.2 | 0.3 | v | 1 | 0.7 |

In Table 3 points 3 and 4, show a multi-concurrency multi-loop with XOR and AND patterns failed to discover use $\alpha_+$ and $\alpha_{++}$. The existence of contiguous short-loops at the input and output places is a major caution of discovery failure due to the un-safeness elimination which produces deadlocks and dead-tasks. Besides, there has been a change in the SWF-Net causality logic which can made ambiguity in the control flow analysis. Although in point 4 $\alpha_{++}$ is correctness to discover NFC both for parallel and free-choice with explicit ◊-free but SWF-Net looked complex and over-fitting.

By the testing using the real large event-log collected by BPI in Hospital Billing and BPI-Challenge 18, it was found that both $\alpha_+$ and $\alpha_{++}$ failed to discover because of various algorithm weaknesses in the various patterns mentioned, particularly the existence of contiguous short-loops at the input and output places and uncertainty relation with many invisible patterns. While $\propto^{T++}$ is soundness and free from deadlocks and dead-task. The picturing of contiguous short-loops in the workflow discovery result, indicating the business process is not efficient in applications. The other algorithms such as heuristic, genetic, inductive, and ILP are not the solutions to construct the SWF-Net loops because they haven't intended to produce the soundness ◊-free loop pattern.

## IV. CONCLUSION

The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font). To create multileveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled.

Provide by the testing results it can be concluded that the novel $\propto^{T+}$ algorithm using the independent ALT structure is able to accurately perform short-loop discovery patterns for the high concurrency both on a structural and behavioral perspectives. By the throughout performance, $\propto^{T+}$ has fitness reply 99% and precission of 95% that is better than another deterministic extension such as $\alpha_+$, and $\alpha_{++}$ for the short-loop application on the uncertainty loops. With the using of pessimistic and optimistic -◊-free extension, $\propto^{T+}$ is free from dead-task and dead-lock which caused by high concurrency loop in parallel, free-choice and explicit NFC.

Furthermore, $\propto^{T+}$ can discover event-log that uncertainty sequence order event though for large event-log because of it has simple pattern matcher and tracer. Compared with other Algorithm such as Heuristic, Inductive, and Genetic Miner for discovery in large event-log, $\propto^{T+}$ is far better because has capability to solve concurrency loop and the others are not. Because $\propto^{T+}$ has capability to apply deterministic without apply data filtering, so it describe a real business process activity in complete. It has benefit and very important for auditing to improvement the business process, so the business process can be aligned with the goals expected by the organization on the future.

## FUTURE WORKS

In the study $\propto^{T+}$ algorithm implemented with Java programming and JGraph visualization library. For further development, we plan to develop several features and tools that cover various standard business process models, such as YAWL and BPMN [29], for online streaming conformance evaluation [30], business process auditing also analyze another aspect such as frequency and time based [31].

## ACKNOWLEDGMENT

Author thanks to *Lembaga Pengelolah Dana Pendidikan* (LPDP) of Indonesian Ministry of Finance that supported the study fund and also gratefully acknowledge by the helpful, comments, and suggestions of the supervisors, reviewers, and editors, which have improved the paper presentation

## REFERENCES

[1] D. Larson and V. Chang, "A review and future direction of agile, business intelligence, analytics and data science," Int J Inf Manage, vol. 36, no. 5, pp. 700–710, Oct. 2016, doi: 10.1016/j.ijinfomgt.2016.04.013.

[2] E. Rojas, J. Munoz-Gama, M. Sepúlveda, and D. Capurro, "Process mining in healthcare: A literature review," J Biomed Inform, vol. 61, pp. 224–236, Jun. 2016, doi: 10.1016/j.jbi.2016.04.007.

[3] R. T. Whitman, M. B. Park, S. M. Ambrose, and E. G. Hoel, "Spatial indexing and analytics on Hadoop," in Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, New York, NY, USA: ACM, Nov. 2014, pp. 73–82. doi: 10.1145/2666310.2666387.

[4] M. de Leoni, W. M. P. van der Aalst, and M. Dees, "A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs," Inf Syst, vol. 56, pp. 235–257, Mar. 2016, doi: 10.1016/j.is.2015.07.003.

[5] E. Börger, "Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL," Softw Syst Model, vol. 11, no. 3, pp. 305–318, Jul. 2012, doi: 10.1007/s10270-011-0214-z.

[6] 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," Data Knowl Eng, vol. 47, no. 2, pp. 237–267, Nov. 2003, doi: 10.1016/S0169-023X(03)00066-1.

[7] J. Li, D. Liu, and B. Yang, "Process Mining : Extending the α -algorithm to Mine Short Loops," in Advances in Web and Network Technologies, and Information Management, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 396–407. doi: 10.1007/978-3-540-72909-9_43.

[8] L. Wen, W. M. P. van der Aalst, J. Wang, and J. Sun, "Mining process models with non-free-choice constructs," Data Min Knowl Discov, vol. 15, no. 2, pp. 145–180, Oct. 2007, doi: 10.1007/s10618-007-0065-y.

[9] L. Wen, J. Wang, W. M. P. van der Aalst, B. Huang, and J. Sun, "Mining process models with prime invisible tasks," Data Knowl Eng, vol. 69, no. 10, pp. 999–1021, Oct. 2010, doi: 10.1016/j.datak.2010.06.001.

[10] A. J. M. M. Weijters and J. T. S. Ribeiro, "Flexible Heuristics Miner (FHM)," in 2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, Apr. 2011, pp. 310–317. doi: 10.1109/CIDM.2011.5949453.

[11] W. Li, Y. Fan, W. Liu, M. Xin, H. Wang, and Q. Jin, "A Self-Adaptive Process Mining Algorithm Based on Information Entropy to Deal With Uncertain Data," IEEE Access, vol. 7, pp. 131681–131691, 2019, doi: 10.1109/ACCESS.2019.2939565.

[12] A. K. A. de Medeiros, A. J. M. M. Weijters, and W. M. P. van der Aalst, "Genetic Process Mining: A Basic Approach and Its Challenges," 2006, pp. 203–215. doi: 10.1007/11678564_18.

[13] 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," 2012, pp. 305–322. doi: 10.1007/978-3-642-33606-5_19.

[14] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, "Discovering Block-Structured Process Models from Incomplete Event Logs," 2014, pp. 91–110. doi: 10.1007/978-3-319-07734-5_6.

[15] H. M. W. Verbeek and W. M. P. van der Aalst, "Decomposed Process Mining: The ILP Case," 2015, pp. 264–276. doi: 10.1007/978-3-319-15895-2_23.

[16] A. Syamsiyah, B. F. van Dongen, and W. M. P. van der Aalst, "DB-XES: Enabling Process Discovery in the Large," 2018, pp. 53–77. doi: 10.1007/978-3-319-74161-1_4.

[17] S. Suriadi, R. Andrews, A. H. M. ter Hofstede, and M. T. Wynn, "Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs," Inf Syst, vol. 64, pp. 132–150, Mar. 2017, doi: 10.1016/j.is.2016.07.011.

[18] W. M. P. van der Aalst et al., "Soundness of workflow nets: classification, decidability, and analysis," Formal Aspects of Computing, vol. 23, no. 3, pp. 333–363, May 2011, doi: 10.1007/s00165-010-0161-4.

[19] Hermawan and Riyanarto Sarno, "A More Efficient Deterministic Algorithm in Process Discovery," IJICIC, vol. 14, no. 3, pp. 971–995, Jun. 2018. doi: DOI: 10.24507/ijicic.14.03.971.

[20] R. Islam, C. Weir, and G. Del Fiol, "Heuristics in Managing Complex Clinical Decision Tasks in Experts' Decision Making," in 2014 IEEE International Conference on Healthcare Informatics, IEEE, Sep. 2014, pp. 186–193. doi: 10.1109/ICHI.2014.32.

[21] N. Tax, N. Sidorova, and W. M. P. van der Aalst, "Discovering more precise process models from event logs by filtering out chaotic activities," J Intell Inf Syst, vol. 52, no. 1, pp. 107–139, Feb. 2019, doi: 10.1007/s10844-018-0507-6.

[22] S. J. van Zelst, B. F. van Dongen, W. M. P. van der Aalst, and H. M. W. Verbeek, "Discovering workflow nets using integer linear programming," Computing, vol. 100, no. 5, pp. 529–556, May 2018, doi: 10.1007/s00607-017-0582-5.

[23] E. Verbeek, "Discovering an S-Coverable WF-net using DiSCover," in 2022 4th International Conference on Process Mining (ICPM), IEEE, Oct. 2022, pp. 64–71. doi: 10.1109/ICPM57379.2022.9980723.

[24] W. van der Aalst, A. Adriansyah, and B. van Dongen, "Causal Nets: A Modeling Language Tailored towards Process Discovery," 2011, pp. 28–42. doi: 10.1007/978-3-642-23217-6_3.

[25] A. Rozinat and W. M. P. van der Aalst, "Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models," 2006, pp. 163–176. doi: 10.1007/11678564_15.

[26] M. Adams, A. V. Hense, and A. H. M. ter Hofstede, "YAWL: An open source Business Process Management System from science for science," SoftwareX, vol. 12, p. 100576, Jul. 2020, doi: 10.1016/j.softx.2020.100576.

[27] M. Rovani, F. M. Maggi, M. de Leoni, and W. M. P. van der Aalst, "Declarative process mining in healthcare," Expert Syst Appl, vol. 42, no. 23, pp. 9236–9251, Dec. 2015, doi: 10.1016/j.eswa.2015.07.040.

[28] A. Rozinat and W. M. P. van der Aalst, "Conformance checking of processes based on monitoring real behavior," Inf Syst, vol. 33, no. 1, pp. 64–95, Mar. 2008, doi: 10.1016/j.is.2007.07.001.

[29] M. Adams, A. V. Hense, and A. H. M. ter Hofstede, "YAWL: An open source Business Process Management System from science for science," SoftwareX, vol. 12, p. 100576, Jul. 2020, doi: 10.1016/j.softx.2020.100576.

[30] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, "Scalable process discovery and conformance checking," Softw Syst Model, vol. 17, no. 2, pp. 599–631, May 2018, doi: 10.1007/s10270-016-0545-x.

[31] R. Sarno, W. A. Wibowo, K. Kartini, Y. Amelia, and K. Rossa, "Determining Process Model Using Time-Based Process Mining and Control-Flow Pattern," TELKOMNIKA (Telecommunication Computing Electronics and Control), vol. 14, no. 1, p. 349, Mar. 2016, doi: 10.12928/telkomnika.v14i1.3257.