Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-2 (2023) 6174–6180

# A Log-Likelihood-Based Evaluation Metric for the Reproducibility and Simplicity of Logistics Graphs

**Yuriko Takakura** * **Junichi Mori** ** **Hirokazu Kobayashi** ***

* *Nippon Steel Corporation, Research & Development, Process Research Laboratories, Chiba, Japan, (e-mail: takakura.kg6.yuriko@jp.nipponsteel.com).*
** *Nippon Steel Corporation, Research & Development, Process Research Laboratories, Chiba, Japan, (e-mail: mori.nn4.junichi@jp.nipponsteel.com)*
*** *Nippon Steel Corporation, Research & Development, Process Research Laboratories, Chiba, Japan, (e-mail: kobayashi.a2c.hirokazu@jp.nipponsteel.com)*

**Abstract:** Logistics are sometimes complex and usually entail the interrelation of different processes. It is beneficial to visualize actual process-flow logs to better understand the underlying processes. However, it is difficult to analyze logistics using graph visualization as the graphs are typically quite large and complex. In the field of process mining, several metrics have been proposed in previous studies to examine the quality of process models created using process mining algorithms. However, these metrics evaluate the efficacy of the graph in terms of reproducing the actual process-flow logs. Thus, there is often process-flow information loss in the resulting graph structure when nodes or edges are aggregated to simplify the graph. To address these limitations, in this paper, we propose a maximum log-likelihood-based metric for measuring the reproducibility of graphs and define the concept in terms of how much of the actual process-flow information the graph retains. To obtain the metrics, we developed a graph model that can generate process-flow logs according to its probability parameters. In addition, we also developed an evaluation metric, which is a weighted sum of both the log-likelihood and model dimensions. An empirical evaluation was conducted using the actual process-flow patterns of steel-making process data. The results revealed that the maximum log-likelihood-based metric effectively evaluated the process by which the graphs with node and edge aggregations reproduced the actual process-flow patterns.



*Keywords:* Machine learning and data analytics in process control, Digital twins for power and process systems, Data visualization, Logistics in manufacturing, Production & logistics over manufacturing networking

## 1. INTRODUCTION

Processes are integral aspects of disparate workflows, including business, industry, and traffic. When logistics are intricate and involved, the relationship between each process is complicated (see van der Aalst (2016)). Taking the in-house logistics of a steel-making process as an example, various types of products are produced using limited production lines in a multi-stage process (see Kumar et al. (2006)). Therefore, each product undergoes different processes, including yards, processing lines, and carriers, and there is a large number of process-flow patterns in a year. In this paper, the term "process-flow pattern" is used for representing the process-flow of each product.

When the logistics become complicated, major challenges such as the shortage or oversupply of stock, lack of carriers, and missed delivery dates are frequently encountered. When analyzing the root cause of these issues, it is not sufficient to investigate the affected process. The entire logistics should be analyzed, including anterior and posterior processes. However, the relationships between these processes are usually complicated and it is difficult to analyze them.

As previously indicated, logistics usually consist of a variety of process-flow patterns. Therefore, it is essential to determine the logistics process-flow patterns by collecting actual data. However, it is difficult to analyze the relationships between each process step when the scale of the logistics is large. To analyze these relationships, previous studies frequently visualize them as a graph (see Pini et al. (2015)), which is sometimes visually cluttered as the process is quite complex and has large scale. Thus, it is necessary to simplify the graph to identify logistics issues.

In previous studies, several process mining methods have been proposed for summarizing the nodes, edges, and event logs to simplify a process model and its graph


10.1016/j.ifacol.2023.10.731

visualization. Process mining is a field of study that uses the process models created from actual process-flow patterns, such as human behavior logs and material flows, and the process models are usually visualized as graphs. In general, the previous methods simplify the process model by removing or aggregating nodes and edges based on various criteria such as frequency thresholds. For example, some metrics are proposed as a part of Fuzzy Mining (see Günther and van der Aalst (2007), van Dongen and Adriansyah (2010)). However, if the process-flow required for logistics analysis contains low-frequency nodes or edges, the obtained process model suffers a loss of critical information required to perform subsequent analyses. As such, these conventional methods do not consider information loss in a process-flow. To consider process flows when aggregating nodes and edges, Trace Clustering was proposed by Song et al. (2009), from which other studies evolved (see Bose and van der Aalst (2010) and Chatain et al. (2017)). In this approach, the process-flow is initially divided into homogeneous subsets and a process model is then created for each subset. However, Trace Clustering methods create several process models based on the assumption that the process-flows do not interact with each other, which may not be realistic for industrial applications.

In logistics analysis, the graph generated from actual process-flow data needs to retain their information. In particular, it is important to know the number of products in each process-flow pattern. Thus, we have to consider the reproducibility of process-flow patterns when simplifying the graph. In this paper, we define reproducibility as how the graph structure retains the required process-flow log information. To simplify the graph structure with the required process-flow log information, a metric for evaluating the process-flow reproducibility of a graph is needed. Accordingly, our goal is to develop a graph structure that can be used as a good estimator for process-flow logs. We propose a maximum log-likelihood-based metric to evaluate the efficacy of a graph structure in terms of process-flow log reproducibility. The basic idea is based on a graph structure and its probability parameters, which can generate process-flow logs, so that reproducibility can be evaluated based on the maximum log-likelihood function. As such, we can simplify the graph by evaluating the maximum log-likelihood function as well as the model dimensions.

The remainder of this paper is organized as follows. In Section 2, we introduce related work. In Section 3, the proposed evaluation method for graph structure is presented. The empirical evaluation results based on the actual process-flow logs for a steel-making process are presented in Section 4. Finally, Section 5 summarizes our main conclusions and presents prospects for future work.

## 2. RELATED WORK

In this section, we summarize previous studies on evaluation metrics for the process models or graph structures that represent process-flows. In process mining, which is a representative field of process-flow analysis, the process model examination is called conformance checking (van der Aalst (2016)). For conformance checking, the

Table 1. Example of actual process-flow data

| Index | Process-flow pattern | Number of products |
|---|---|---|
| 1 | S1-V1-G1 | 100 |
| 2 | S1-V2-G1 | 90 |
| 3 | S1-V3-G1 | 10 |
| 4 | S2-V2-G1 | 10 |
| 5 | S2-V3-G1 | 90 |

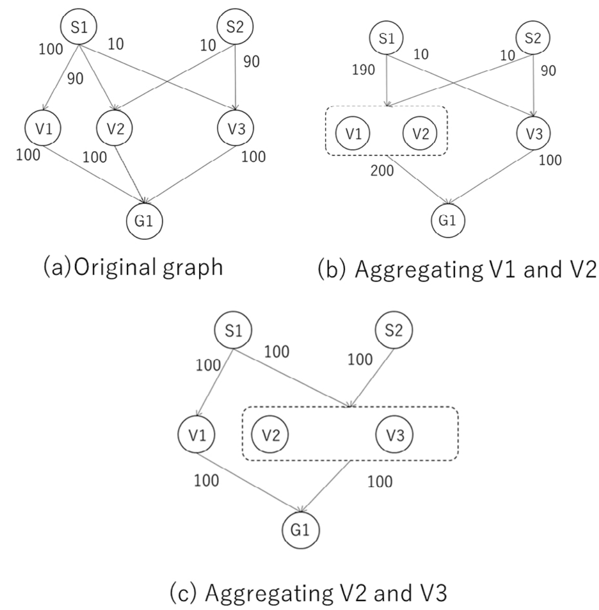


Fig. 1. Examples of node aggregations using process-reproduced graph models

following four metrics are generally used: “fitness,” “precision,” “generalization,” and “simplification” (see van der Aalst (2016), Debois et al. (2017) and Buijs et al. (2012)). As this study aims at logistics analysis of past process data and does not consider the future process-flow behavior, we do not introduce generalization in this section.

Fitness is frequently used to evaluate the process model and is sometimes considered within the context of recall, which deals with how much actual process-flow patterns are reproduced in the process model (Weerdt et al. (2012)). Several metrics have been proposed in previous studies for fitness. The standard equation is as (1):

$$\text{fitness} = \frac{L_{traced}}{L_{full}} \tag{1}$$

where the $L_{full}$ is the number of actual process-flow patterns, and $L_{traced}$ represents the number of traced process-flow patterns in the process model.

Precision is a metric for evaluating whether the process models produce process-flow patterns that are not related to the actual process-flow patterns, and various types of metrics have been proposed (see van der Aalst (2016), Tax et al. (2018)). Greco et al. (2006) proposed a precision measure named soundness. Weerdt et al. (2011) presented a precision metric considering the metric used in data mining. They also evaluated the process models using the F-measure, which is a harmonic mean of the precision and recall metrics. Precision considers the number of process-flow patterns in the model that are seen in the actual process-flow logs divided by the number of all process-flow patterns in the model.

Precision is calculated as (2):

$$\text{precision} = \frac{\sum_{k=1}^{K} n_k T_k}{\sum_{k=1}^{K} n_k T_k + \sum_{k=1}^{K} n_k F_k} \tag{2}$$

where $K$ is the number of the process-flow pattern groups in the model and $n_k$ is the number of actual process-flow logs corresponding to the k-th process-flow pattern group. $T_k$ is the number of unique process-flow patterns of the k-th group in the model that are seen in the actual process-flow logs and these process-flow patterns are called true positive. Meanwhile, $F_k$ represents the number of process-flow patterns that are never made in the actual logs and are called false positive.

Simplicity is a representation metric with a simple structure (see Lieben et al. (2018)). One example of simplicity is how many nodes and edges are included. Mendling et al. (2007) and Mendling (2007) discussed the relationship between the complexity and understandability of a process model.

In process mining, the process models are abstracted and simplified and resulting models are evaluated. As previously indicated, there are two main approaches for graph simplification: aggregating and cutting nodes and edges. Fitness works well when evaluating how many actual process-flow patterns are reproduced from the process models after cutting nodes and edges. This is because when nodes or edges are removed from a graph, the corresponding process-flow patterns cannot be reproduced, which can be measured by the fitness metric. Meanwhile, regardless of the number of aggregated nodes and edges, we can replay all process-flow logs, which means fitness remains the same regardless of how nodes are aggregated. This suggests that fitness does not evaluate the process-flow reproducibility of the process model after aggregation. Conversely, precision can evaluate the true positive patterns with respect to all process-flow patterns in the model. Using the precision, we can consider the process-flow patterns which only included in the process model. However, as nodes are aggregated, the number of false positive patterns tends to increase, so that the precision gets worse regardless of how well nodes are aggregated.

Fig. 1(a) shows a simple logistics graph, where each product is delivered from start nodes S1 or S2 to end node G1. Each process-flow pattern and the corresponding number of products are shown in Table 1. Process patterns S1-V3-G1 and S2-V2-G1 occur infrequently with ten products each.

Now we consider the two aggregated graphs shown in Fig. 1(b) and 1(c). As mentioned above, the frequency of patterns S1-V3-G1 and S2-V2-G1 is very low; therefore, the aggregation of V1 and V2 shown in Fig. 1(b) seems to be more reasonable than that of V2 and V3 shown in Fig. 1(c). However, the precision of the structure in Fig. 1(c) is 1.00, which is higher that of the one in Fig. 1(b) (0.98). This example verifies that precision does not have the capability of considering the frequency of process-flow patterns.

Table 2. Notation used in Section 3

| Notation | Description |
|---|---|
| $\theta_k$ | k-th probability parameter |
| $G$ | A given graph structure |
| $e_i$ | i-th process-flow log. $e_i \in \boldsymbol{E}$ |
| $p_j$ | j-th process-flow pattern |
| $N_j$ | Number of process-flow logs in each $p_j$ |
| $q_k$ | k-th process-flow pattern generated from the graph structure. $q_k \in \boldsymbol{Q}$ |
| $\lambda$ | A Lagrange multiplier |
| $w$ | Score coefficient of model dimensions |

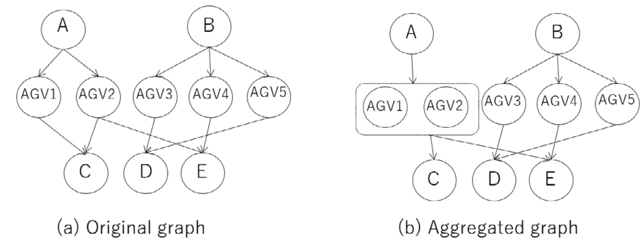


Fig. 2. Examples of original and aggregated graphs

Fitness values for all logistics graphs shown in Fig. 1 are equal to 1.00, which means that all actual process-flow patterns can be represented. This implies that fitness is not appropriate for evaluating the extent to which process-flows are reproduced in the graphs.

In this paper, a novel evaluation metric is proposed for the reproducibility of graphs where some nodes and edges are aggregated or removed. The basic idea is to use the graph structure as well as the probability parameters that can consider the frequency of process-flow patterns. Then, the likelihood of the process-flow logs given the graph structure is calculated to measure the reproducibility of process-flow logs. Furthermore, to evaluate the simplicity of the graph structure, we introduce the model dimensions, which equal the number of probability parameters.

## 3. EVALUATION METRIC

As previously indicated, it is essential to reproduce the process-flow logs from graph structure $G$. The notation used in this section is shown in the Table 2. In the reproducibility evaluation, we consider a graph model that can generate process-flow logs according to its probability parameters $\boldsymbol{\Theta} = [\theta_1, ...\theta_K] \in R^K$, where $K$ is the number of process-flow patterns generated from the graph structure. Using this graph model, it is possible to evaluate the reproducibility of process-flow logs based on the likelihood function. The log-likelihood function given the actual process-flow logs $\boldsymbol{E} = \{e_1, ..., e_I\}$ is described as (3):

$$\ln f(\boldsymbol{E}|\boldsymbol{\Theta}; G) = \sum_i \ln f(e_i|\boldsymbol{\Theta}; G) \tag{3}$$

where $I$ is the number of products and $e_i$ represents the process-flow log of the i-th product and is defined as (4):

$$e_i \equiv \text{ProcessFlowPattern} \tag{4}$$

The function $f(e_i|\boldsymbol{\Theta}; G)$ is the likelihood of each process-flow log $e_i$ given probability parameters $\boldsymbol{\Theta}$ and graph

Table 3. Process-flow patterns

| $i$ | Actual process-flow patterns |
|---|---|
| 1 | A-AGV1-C |
| 2 | A-AGV1-C |
| 3 | A-AGV1-C |
| 4 | A-AGV1-C |
| 5 | A-AGV1-C |
| 6 | A-AGV1-C |
| 7 | A-AGV1-C |
| 8 | A-AGV1-C |
| 9 | A-AGV1-C |
| 10 | A-AGV1-C |
| 11 | A-AGV2-C |
| 12 | A-AGV2-C |
| 13 | A-AGV2-C |
| 14 | A-AGV2-C |
| 15 | A-AGV2-C |
| 16 | A-AGV2-E |
| 17 | A-AGV2-E |
| 18 | A-AGV2-E |
| 19 | A-AGV2-E |
| 20 | A-AGV2-E |
| 21 | B-AGV3-D |
| 22 | B-AGV3-D |
| 23 | B-AGV3-D |
| 24 | B-AGV3-D |
| 25 | B-AGV3-D |
| 26 | B-AGV3-D |
| 27 | B-AGV3-D |
| 28 | B-AGV3-D |
| 29 | B-AGV3-D |
| 30 | B-AGV3-D |
| 31 | B-AGV4-E |
| 32 | B-AGV4-E |
| 33 | B-AGV4-E |
| 34 | B-AGV4-E |
| 35 | B-AGV4-E |
| 36 | B-AGV4-E |
| 37 | B-AGV4-E |
| 38 | B-AGV4-E |
| 39 | B-AGV4-E |
| 40 | B-AGV4-E |
| 41 | B-AGV5-D |
| 42 | B-AGV5-D |
| 43 | B-AGV5-D |
| 44 | B-AGV5-D |
| 45 | B-AGV5-D |
| 46 | B-AGV5-D |
| 47 | B-AGV5-D |
| 48 | B-AGV5-D |
| 49 | B-AGV5-D |
| 50 | B-AGV5-D |

structure $G$. Examples of process-flow logs are shown in Table 3. The 1st product from process A to process C using AGV1 is represented as $e_1$ =A-AGV1-C. The graph structure is visualized in Fig. 2(a). Next, we use the model's maximum log-likelihood (MLL) estimation to evaluate the graph structure (not for evaluating the model parameters), since the model with the maximum likelihood is the one that can best reproduce the process-flow logs given the graph structure. By rearranging (3), it can be determined that the likelihood function depends on the number of products for each actual process-flow pattern $\boldsymbol{P} = \{p_1, ...p_J\}$, where $p_j$ represents each process-flow pattern and $J$ is the number of actual process-flow patterns. Equation (3) can be rewritten as (5):

$$\sum_j N_j \ln f(p_j|\boldsymbol{\Theta}; G) \tag{5}$$

where $\boldsymbol{N} = [N_1, ..., N_J] \in Z^J$ represents the number of products for each process-flow pattern $p_j$. Using the example in Table 3, a set of actual operation patterns can be written as $\boldsymbol{P}$={A-AVG1-C, A-AGV2-C, A-AGV2-E, B-AVG3-D, B-AGV4-E, B-AGV5-D} and $\boldsymbol{N} = [10, 5, 5, 10, 10, 10]$. To calculate the log-likelihood in (5) given the graph structure $G$, we need to compute $\boldsymbol{\Theta}$ in advance. First, we introduce a set of the process-flow patterns generated from the graph structure, which is represented as $\boldsymbol{Q} = \{q_1, ...q_K\}$. For instance, when we aggregate nodes AGV1 and AGV2 into one node, $\boldsymbol{Q}$ is described as in Table 4. We also introduce subsets $\boldsymbol{E}_k \subseteq \boldsymbol{E}$ and $\boldsymbol{P}_k \subseteq \boldsymbol{P}$, where $\boldsymbol{E}_k$ and $\boldsymbol{P}_k$ are sets of process-flow patterns corresponding to $q_k$ as (6) and (7), respectively:

$$\boldsymbol{E}_k = \{e_i \in \boldsymbol{E} | e_i \;\; can\ be\ replayed\ on\ q_k \;\; \forall k\} \tag{6}$$

$$\boldsymbol{P}_k = \{p_j \in \boldsymbol{P} | p_j \;\; can\ be\ replayed\ on\ q_k \;\; \forall k\} \tag{7}$$

For instance, if $q_1$ = A-<AGV1,AGV2>-C, then $\boldsymbol{E}_1 = \{e_1, ..., e_{15}\}$, $\boldsymbol{P}_1$ = {A-AGV1-C, A-AGV2-C} and $n(\boldsymbol{P}_1) = 2$ because all elements of $\boldsymbol{E}_1$ and $\boldsymbol{P}_1$ can be replayed on process-flow pattern $q_1$. Table 4 shows the results obtained based on the graph model in Fig. 2(b). We also define function $h(j)$ as (8):

$$h(j) := \begin{cases} k & where\ p_j \in \boldsymbol{P}_k \\ 0 & if\ \{p_j \in \boldsymbol{P}_k\} = \emptyset \end{cases} \tag{8}$$

which means that $h(j)$ returns index $k$ so that $p_j$ can be replayed on $\boldsymbol{P}_k$. Using (8), (5) can be rewritten as (9):

$$\sum_j N_j \ln \frac{\theta_{h(j)}}{n(\boldsymbol{P}_{h(j)})} \tag{9}$$

$\theta_{h(j)}$ is a probability parameter when $h(j)$ is equal to $k$. Note that $q_k$ can include multiple process-flow patterns and thus $\theta_{h(j)}$ must be divided by $n(\boldsymbol{P}_{h(j)})$. Since $n(\boldsymbol{E}_k)$ can be written as (10).

$$n(\boldsymbol{E}_k) = \sum_{\{j|k=h(j)\}} N_j \tag{10}$$

then we can write (9) as (11):

$$\begin{aligned} \sum_j N_j \ln \frac{\theta_{h(j)}}{n(\boldsymbol{P}_{h(j)})} &= \sum_k \sum_{\{j|k=h(j)\}} N_j \ln \frac{\theta_{h(j)}}{n(\boldsymbol{P}_{h(j)})} \\ &= \sum_k n(\boldsymbol{E}_k) \ln \frac{\theta_k}{n(\boldsymbol{P}_k)} \end{aligned} \tag{11}$$

The log-likelihood function can now be calculated given graph structure $G$ and probability parameters $\boldsymbol{\Theta}$. We are only interested in the graph structure, so that the maximum likelihood solution for $\boldsymbol{\Theta}$ can be obtained given graph structure $G$. Therefore, we need to maximize (11) with respect to $\theta_k$ taking into account the constraint that the sum of $\theta_k$ must be unity. This can be achieved using a Lagrange multiplier $\lambda$ and maximizing, and is expressed as (12):

$$F(\boldsymbol{\Theta}, \lambda) = \sum_j n(\boldsymbol{E}_k) \ln \frac{\theta_k}{n(\boldsymbol{P}_k)} - \lambda(\sum_k \theta_k - 1) \tag{12}$$

The derivative of the log-likelihood in (12) with respect to $\boldsymbol{\Theta}$ and $\lambda$ is given by (13) and (14).

$$\frac{\partial}{\partial \theta_k} F(\boldsymbol{\Theta}, \lambda) = \frac{n(\boldsymbol{E}_k)}{\theta_k} - \lambda \quad \forall k \tag{13}$$

$$\frac{\partial}{\partial \lambda} F(\boldsymbol{\Theta}, \lambda) = \sum_k \theta_k - 1 \tag{14}$$

By setting this derivative to zero, we obtain the solution for the maximum likelihood estimate of the probability parameter, which is given by (15)

$$\theta_k^{ML} = \frac{n(\boldsymbol{E}_k)}{\sum_k n(\boldsymbol{E}_k)} \quad \forall k \tag{15}$$

which is a fraction of the number of products corresponding to process-flow pattern $q_k$. Finally, we can calculate the maximum log-likelihood function as (16):

$$\theta_k^{ML} = \sum_k n(\boldsymbol{E}_k) \ln \frac{\theta_k^{ML}}{n(\boldsymbol{P}_k)} \tag{16}$$

Table 4. Patterns reproduced from graph (b)

| $k$ | Process-flow patterns reproduced from graph model, $q_k$ | $n(\boldsymbol{P}_k)$, the Number of process-flow patterns in $q_k$ | $n(\boldsymbol{E}_k)$, the number of actual-process patterns in $q_k$ |
|---|---|---|---|
| 1 | A–$<AGV1, AGV2>$–C | 2 | 15 |
| 2 | A–$<AGV1, AGV2>$–E | 2 | 5 |
| 3 | B–AGV3–D | 1 | 10 |
| 4 | B–AGV4–E | 1 | 10 |
| 5 | B–AGV5–D | 1 | 10 |

If process pattern $p_j$ cannot be replayed on any process-flow pattern $\boldsymbol{P}_k$ (which means $\{p_j \in \boldsymbol{P}_k\} = \emptyset$), then neither $\theta_{h(j)}^{ML}$ nor $n(\boldsymbol{P}_h(j))$ can be calculated. Therefore, we apply a smoothing technique, such as Laplacian smoothing as (17):

$$\theta_k^{ML} = \frac{n(\boldsymbol{E}_k) + 1}{\sum_k n(\boldsymbol{E}_k) + K} \tag{17}$$

where we define $\boldsymbol{P}_0 : = \emptyset$. Using this approach, it is possible to evaluate how well the graph structure reproduces the actual process-flow logs using the maximum likelihood function in (16). To consider the trade-off between reproducibility and simplicity, which is the sum of the log-likelihood in (16) and the model dimensions $\mathrm{Dim}[G]$, which is related to the number of model parameters $K$ as (18):

$$\mathrm{score}(G; \boldsymbol{E}) = \sum_k n(\boldsymbol{E}_k) \ln \frac{\theta_k^{ML}}{n(\boldsymbol{P}_k)} - w \cdot \ln(I) \cdot \mathrm{Dim}[G] \tag{18}$$

where $w$ represents a score coefficient of the model dimensions which is set to a larger value if more simplified graph structures are required for logistics analysis. Owing to the term of model dimensions, the proposed score includes a trade-off between the likelihood and model complexity. With the score function, the optimal graph $G^*$ can be calculated as (19):

$$G^* = \underset{G}{\mathrm{argmax}}\, \mathrm{score}(G : \boldsymbol{E}) \tag{19}$$

When the targeted graph structure has loops, the number of generated process-flow patterns $K$ can be infinite. Then, the process should be $\boldsymbol{Q} = \{q_1, ... q_\infty\}$, $\boldsymbol{\Theta} = [\theta_1 ... \theta_\infty]$. However, $\theta_k$=0 when process-flow pattern $q_k$ is not included in the actual process-flow data. Thus, it is sufficient that we compute $\theta_k^{ML}$ within a finite number of $K$. Therefore, the set of $k$ is defined as (20) when $\theta_k^{ML}$ is calculated.

$$\theta_k^{ML} = \frac{n(\boldsymbol{E}_k)}{\sum_k n(\boldsymbol{E}_k) + K} \quad \forall \{k | \exists q_k \ can\ replay\ e_i,\ e_i \in \boldsymbol{E}, q_k \in \boldsymbol{Q}\} \tag{20}$$

It should be noted that this combinational optimization problem is known to be difficult to solve. Therefore, any heuristic search algorithm can be used; however, this is outside the scope of this paper.

## 4. EMPIRICAL EVALUATION

We conducted an empirical evaluation using actual process-flow data of steel-making process. The process included yards, processing lines, and carriers, and it has 1188 process-flow patterns during the period from January to December 2019. We created a process model using this data and visualized it using Disco (see Fig. 3, see Fluxicon (2022)). We found that it is difficult to extract information from this complicated process model.

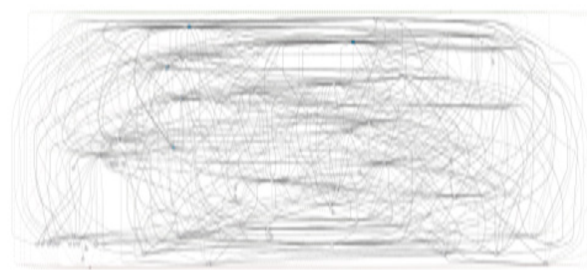

Fig. 3. Visualization result for one year of a steel-making process logs. The process-flows are complicated and the process-flow features are difficult to extract.

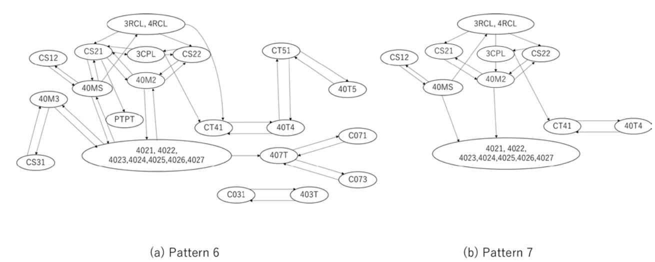


Fig. 4. Visualization results for patterns 6 and 7

The graph nodes show the processes, including yards, processing lines, and carriers. The edges show the transfer of products and each edge has information on the frequency of the transfers in the period. To evaluate the proposed metric, we created graphs with aggregating nodes and cutting edges. In Table 5, the aggregated nodes are enclosed with angle brackets, <>. The column "Threshold for edge reduction" shows the minimum frequency of cutting the edges. For example, if the number in this column is 50, the edges are cut if their frequencies are below 50.

This section shows the proposed metric evaluation results using the actual process-flow data from April to October 2019. We created several graphs with node aggregation and edge cutting, as shown in Table 5.

This table shows the maximum log-likelihood values for several graphs, which are compared to the previous metrics (i.e., fitness, precision, and F-measure). Using the recall and precision, the F-measure is written as (21):

$$\mathrm{Fmeasure} = \frac{2 \cdot \mathrm{recall} \cdot \mathrm{precision}}{\mathrm{recall} + \mathrm{precision}} \tag{21}$$

Table 5. The values of the evaluation metric in each case

| Patterns | Aggregated nodes | Threshold for edge reduction | Maximum log-likelihood | Fitness | Precision | F-measure |
|---|---|---|---|---|---|---|
| 1 | Non | 0 | -41311.2 | 1 | 1 | 1 |
| 2 | <4021, 4022, 4023, 4024, 4025, 4026, 4027> | 0 | -42551.66 | 1 | 0.1766 | 0.3001 |
| 3 | <4021, 4022, 4023, 4024, 4025, 4026, 4027> <3RCL, 4RCL> | 0 | -43786.86 | 1 | 0.1476 | 0.2572 |
| 4 | <4021, 4022, 4023, 4024, 4025, 4026, 4027> <3RCL, 4RCL, 3CPL> | 0 | -55555.47 | 1 | 0.0125 | 0.0250 |
| 5 | <4021, 4022, 4023, 4024, 4025, 4026, 4027> <3RCL, 4RCL><CS21, CS22> | 0 | -53018.68 | 1 | 0.0010 | 0.0019 |
| 6 | <4021, 4022, 4023, 4024, 4025, 4026, 4027> <3RCL, 4RCL> | 50 | -43633.88 | 0.93 | 0.2528 | 0.3975 |
| 7 | <4021, 4022, 4023, 4024, 4025, 4026, 4027> <3RCL, 4RCL> | 300 | -45791.77 | 0.82 | 0.9915 | 0.8976 |

We use the fitness represented in (1) to calculate the recall.

Pattern 1 shows the original graph without any aggregation and cutting. The graphs for patterns 2 to 5 are created based on the original graph and in which some nodes in pattern 1 are aggregated. In pattern 2, seven AGVs, numbered 4021 to 4027, are aggregated. These AGVs run in the same area for carrying products. Pattern 3 has two aggregated nodes, the same AGVs in pattern 1, and two processing lines, 3RCL and 4RCL. "RCL" represents a type of processing line in the steel-making process, and 3RCL and 4RCL have the same function. Pattern 4 aggregates the seven AGVs, the two RCLs, and 3CPL. 3CPL is a different processing line from the RCLs. Pattern 5 has three aggregated nodes, seven AGVs, two RCLs, and two carriers, CS21, and CS22. These carriers also run in the same area and have the same function.

Among patterns 2 to 5, pattern 2 achieved the maximum value of the maximum log-likelihood, which decreased from -41311.2 to -42551.6 and declined less than the other patterns. The aggregated node in pattern 2 consists of seven AGVs that run in the same area. In this case, the aggregation is effective. Meanwhile, pattern 4 has the worst value among the four aggregation patterns. The aggregated nodes of pattern 4 include seven AGVs, two RCL processing lines, and processing line 3CPL. As previously indicated, the two RCL processing lines 3RCL and 4RCL have the same functions; however, the 3CPL does not have any relationship to the RCLs. When these nodes are aggregated, the maximum log-likelihood decreases dramatically because we aggregate the nodes with different processing lines. Pattern 5 aggregates two carriers in addition to the aggregated nodes in pattern 3. The maximum log-likelihood value of pattern 5 is better than pattern 4 even though pattern 5 aggregates more nodes than pattern 4. We can state that the proposed evaluation metric effectively evaluates node aggregations in similar functions.

Conversely, the fitness values for pattern 2 to pattern 5 equal unity and it is impossible to measure the difference in how nodes are aggregated.

The precision values are changed to reflect the node aggregations. However, we can readily observe that the precision value of pattern 4 is 0.0125, which is higher than that of pattern 5 (0.0010). However, pattern 4 aggregates nodes which indicate different functions. Therefore, this comparison suggests that precision cannot measure how nodes are aggregated. The maximum log-likelihood-based metric takes into account the probability of each actual process-flow pattern. Meanwhile, precision focuses only on the number of process-flow patterns included in the graph. The F-measure values have same trend as the precision values since they are the harmonic mean of fitness and precision, and fitness is unity in these patterns.

Finally, we examine the results using the values calculated with (18). Patterns 6 and 7 are the results for edge reduction based on changing the threshold to 50 and 300 in the column "Threshold for edge reduction." When the threshold is 50, 35 edges are cut, and when the threshold is 300, 44 edges are cut. Fig. 4 shows the visualization results of these two patterns. The maximum log-likelihood values of patterns 6 and 7 are -43633.88 and -45791.77, respectively, which indicates that the likelihood decreases when the cut edges increase. The fitness values of patterns 6 and 7 are 0.93 and 0.82, respectively. Based on these results, the maximum log-likelihood of a graph can evaluate the edge-cutting results similarly to fitness. The model dimensions of patterns 6 and 7 are 410 and 304, respectively. We changed $w$ to 1 and 10 and compared the results. For the former, the metric's value is better in the case of pattern 6, -47280.13, than that of pattern 7, -48494.54. Conversely, when $w$ is changed to 10, the metric indicated that pattern 7 is better, with values of -72826.63 and -80096.40 for in patterns 7 and 6, respectively. These results show that if $w$ is large, the metric emphasizes the simplicity of the graph; and if $w$ is small, reproducibility is highlighted. In addition, the proposed evaluation metric can extract graphs that are aggregated with yards, processing lines, and carriers, which have the same capabilities.

## 5. CONCLUSION

We proposed a maximum log-likelihood-based metric to effectively evaluate simplified logistics graphs considering reproducibility, which is how the graph retains information related to the targeted logistics. The graph model is a probabilistic model that can be used to evaluate reproducibility using the maximum log-likelihood value. In addition, we also consider the score which is the sum of the maximum log-likelihood and the model dimensions. The first term of (18) evaluates the reproducibility with a maximum log-likelihood calculated from the graph and actual process-flow logs. The second term of (18) shows the complexity of the graph. It represents the ability of logistics graphs in terms of reproducing the actual process-flow patterns.

We conducted an empirical evaluation of the proposed metric using actual process-flow data for a steel-making process. The results show that the maximum log-likelihood-based metric can effectively evaluate the reproducibility of the process-flow, even if the nodes of the graph are aggregated. As indicated in Section 2, it is difficult to evaluate the results of node aggregations using traditional metrics. In addition, when $w$ in the second term of the "score" equation is changed, the simplicity of the extracted graph can be considered. These results show that the appropriate graph can be extracted for logistics analysis using the proposed evaluation metric. In future work, we will apply the proposed metric to explore the best graph for logistics analysis.

## REFERENCES


Bose, R.P.J.C. and van der Aalst, W.M.P. (2010). Trace clustering based on conserved patterns: Towards achieving better process models. *Business Process Management Workshops 2009. Lecture Notes in Business Information Processing*, 43, 170–181.

Buijs, J.C.A.M., van Dongen, B.F., and van der Aalst, W.M.P. (2012). On the role of fitness, precision, generalization and simplicity in process discovery. *On the Move to Meaningful Internet Systems 2012, Lecture Notes in Computer Science*, 7565, 305–322.

Chatain, T., Carmona, J., and van Dongen, B.F. (2017). Alignment-based trace clustering. *Conceptual Modeling 2017, Lecture Notes in Computer Science*, 10650, 295–308.

Debois, S., Hildebrandt, T.T., Laursen, P.H., and Ulrik, K. (2017). Declarative process mining for dcr graphs. In *Proceedings of the Symposium on Applied Computing*, 759–764.

Fluxicon (2022). Discover your processes. URL `https://fluxicon.com/disco/`.

Greco, G., Guzzo, A., Pontieri, L., and Sacca, D. (2006). Discovering expressive process models by clustering log traces. *IEEE Transactions on Knowledge and Data Engineering*, 18(8), 1010–1027.

Günther, C.W. and van der Aalst, W. (2007). Fuzzy mining–adaptive process simplification based on multi-perspective metrics. *Business Process Management*, 4714, 328–343.

Kumar, V., Kumar, S., Tiwari, M.K., and Chan, F.T.S. (2006). Auction-based approach to resolve the scheduling problem in the steel making process. *International Journal of Production Research*, 44(8).

Lieben, J., Jouck, T., Depaire, B., and Jans, M. (2018). An improved way for measuring simplicity during process discovery. *Enterprise and Organizational Modeling and Simulation 2018, Lecture Notes in Business Information Processing*, 332, 49–62.

Mendling, J. (2007). Detection and prediction of errors in epc business process models.

Mendling, J., Neumann, G., and van der Aalst, W.M.P. (2007). Understanding the occurrence of errors in process models based on metrics. *On the Move to Meaningful Internet Systems 2007, Lecture Notes in Computer Science*, 4803, 113–130.

Pini, A., Brown, R., and Wynn, M. (2015). Process visualization techniques for multi-perspective process comparisons. *Asia Pacific Business Process Management, Lecture Notes in Business Information Processing*, 219, 183–197.

Song, M., Günther, C.W., and van der Aalst, W.M.P. (2009). Trace clustering in process mining. *Business Process Management Workshops 2008. Lecture Notes in Business Information Processing*, 17, 109–120.

Tax, N., Lu, X., Sidorova, N., Fahland, D., and van der Aalst, W.M.P. (2018). The imprecisions of precision measures in process mining. *Information Processing Letters*, 135, 1–8.

van der Aalst, W. (2016). *Process Mining: Data Science in Action*. Springer, Berlin Heidelberg.

van Dongen, B.F. and Adriansyah, A. (2010). Process mining: Fuzzy clustering and performance visualization. *Business Process Management Workshops International Workshops*, 43, 158–169.

Weerdt, J.D., Backer, M.D., Vanthienen, J., and Baesens, B. (2011). A robust f-measure for evaluating discovered process models. In *Proceedings of IEEE Symposium on Computational Intelligence and Data Mining*, 148–155.

Weerdt, J.D., Backer, M.D., Vanthienen, J., and Baesens, B. (2012). A multi-dimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs. *Information Systems*, 37(7), 654–676.