Addressing Convergence, Divergence, and Deficiency Issues

Jan Niklas Adams[0000−0001−8954−4925] and Wil M.P. van der Aalst[0000−0002−0955−6940]

Chair of Process and Data Science, RWTH Aachen University, Aachen, Germany
{niklas.adams,wvdaalst}@pads.rwth-aachen.de

Abstract. The application of process mining algorithms to event logs requires the extraction of cases, describing end-to-end runs through the process. When extracting cases for object-centric event data, this extraction is often subject to convergence, divergence, and deficiency issues. Recently, connected-components extraction was proposed, extracting graph-based cases, called *process executions*, from the graph of event precedence constraints. This paper shows that only case extraction based on connected-components is free of convergence, divergence, and deficiency issues. This proof has several implications for future research in object-centric process mining. First, if a downstream process mining task is negatively affected by these quality issues, connected-components extraction is the only way to mitigate these. Second, additional requirements that would conflict with connected-components extraction would render the mitigation of quality issues infeasible, making trade-offs between quality issues necessary. Third, as traditional event logs are a special case of object-centric event logs and connected-components extraction is equivalent to the traditional case concept for a traditional event log, new extraction techniques, as well as object-centric adaptations of algorithms, should be backward-compatible.

Keywords: Object-Centric Process Mining · Event Data · Flattening

1 Introduction

In most information systems, events are documented in relation to multiple entities, or objects [8]. Process mining algorithms require input in the form of cases, which are sets of events with precedence constraints. When each event is connected to precisely one object, case extraction is trivial as every object defines its own case [1]. However, when multiple objects are linked to a single event, extraction becomes problematic. For example, extracting one case per object results in event duplication when an event is associated with two objects.

Previous research has identified three quality problems when extracting cases from event data with multiple objects per event (object-centric event data): convergence, deficiency, and divergence [2,11,8]. These quality problems can lead to challenges in downstream process mining tasks due to cases containing inaccurate data.

Adams et al. have proposed connected-components process execution extraction [5]. This extraction merges all events and precedence constraints connected through common objects into graph-based cases, called *process executions*. In this paper, we show that connected-components extraction is free of convergence, divergence, and deficiency issues and that all extraction techniques not based on connected components do not meet the requirements of being convergence, divergence, and deficiency-free. We situate these findings with respect to the fields of object-centric and traditional process mining and discuss the implications of our proof for the field of process mining.

2 Event Data

A sequence of length $n \in \mathbb{N}$ is a function $\sigma : \{1, \ldots, n\} \to X$. We denote a sequence with $\sigma = \langle x_1, \ldots, x_n \rangle$. Sequences can be concatenated, denoted by $\langle x_1, \ldots, x_n \rangle \cdot \langle y_1, \ldots, y_m \rangle = \langle x_1, \ldots, x_n, y_1, \ldots, y_m \rangle$. A sequence $\delta_{sub} = \langle y_1, \ldots, y_m \rangle$ is a subsequence of $\delta_{sup} = \langle x_1, \ldots, x_n \rangle$ if the complete δ_{sub} can be mapped to consecutive indices of δ_{sup}, i.e., $\exists_{i \in \{0,\ldots,n-m\}} \forall_{j \in \{1,\ldots,m\}} x_{i+j} = y_j$. We denote this by $\delta_{sub} \in \delta_{sup}$. The powerset of a set X defines the set of all possible sets and is denoted by $\mathcal{P}(X)$. The power set without the empty set is denoted by $\mathcal{P}^+(X)$. A directed graph is a tuple $G = (E, K)$ of nodes E and edges $K \subseteq E \times E$. A path between two nodes $e, e' \in E$ describes a sequences of nodes that are connected by edges $e \sim_G e' = \langle e_1, \ldots, e_n \rangle$ such that $e_1 = e \wedge e_n = e' \wedge \forall_{1 \leq i < n} (e_i, e_{i+1}) \in K$. We abbreviate the existence of a path with $e \sim_G e'$. If there is no path between two nodes $e, e' \in E$ then $e \nsim_G e'$. We explicitly define the path from a node to itself as $e \sim_G e = \langle e \rangle$.

Lemma 1 (Connecting Edge). *Let $G = (E, K)$ be a graph and $e, e' \in E$ be two nodes. There is a set of events $E' \subseteq E$ with e' being part of the set and e not being part of the set: $e \in E \setminus E' \wedge e' \in E'$. If there exists a path between e and e' then there is an edge that connects the two sets $E \setminus E'$ and E': $e \sim_G e' \Rightarrow \exists_{(e_1,e_2) \in K}\ e_1 \in E \setminus E' \wedge e_1 \notin E' \wedge e_2 \in E' \wedge e_2 \notin E \setminus E'$.*

Proof. The path between $e, e' \in E$ is $e \sim_G e' = \langle e_1, \ldots, e_n \rangle$ such that $e_1 = e \wedge e_n = e' \wedge \forall_{1 \leq i < n} (e_i, e_{i+1}) \in K$. Each node of E is either part of E' or $E \setminus E'$. We prove by contradiction: If there would not exist an edge that connects both sets, all elements in the path would need to be of the same set: $\neg \exists_{\langle e_1,e_2 \rangle \in e \sim_G e'}\ e_1 \in E \setminus E'' \wedge e_2 \in E'' \Rightarrow \{e_1, \ldots e_n\} \subseteq E \setminus E' \vee \{e_1, \ldots e_n\} \subseteq E'$. However, $e_1 \in E \setminus E'$ and $e_n \in E'$, i.e., there is at least on element of both sets in $e \sim_G e'$. Therefore, $\exists_{(e_1,e_2) \in K}\ e_1 \in E \setminus E' \wedge e_1 \notin E' \wedge e_2 \in E' \wedge e_2 \notin E \setminus E'$.

Lemma 2 (Transitivity). *Let $G = (E, K)$ be a graph and $e, e' \in E$ be two nodes with $e \nsim_G e'$ with $E_1 = \{e'' \in E \mid e \sim_G e'' \vee e'' \sim_G e\}$ and $E_2 = \{e'' \in E \mid e' \sim_G e'' \vee e'' \sim_G e'\}$. The reachable nodes from e do not overlap with the reachable nodes from e': $E_1 \cap E_2 = \emptyset$.*

Proof. We prove this by contradiction. Assume there is an event $e'' \in E$ such that $e'' \in E_1 \wedge e'' \in E_2$. By construction of E_1 and E_2, it holds that $e \sim_G e''$ or $e'' \sim_G e$ and $e' \sim_G e''$ or $e'' \sim_G e'$. Therefore, there exists a path $e \sim_G e'$ by concatenating these two paths which conflicts the initial statement.

An event log consists of events. Each event is identified through an element from the universe of event identifiers $\mathcal{E}$. An event describes the execution of an activity at a given time for affected objects. The universe of activities is denoted by $\mathcal{A}$, the universe of timestamps is denoted by $\mathcal{T}$, and the universe of objects is denoted by $\mathcal{O}$. Each object is of a type from the universe of types $\mathcal{OT}$. This type is given by the typing mapping $\pi_{type} : \mathcal{O} \to \mathcal{OT}$.

Definition 1 (Event Log). *An event log is a tuple* $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *consisting of*

- *events* $E \subseteq \mathcal{E}$*, objects* $O \subseteq \mathcal{O}$*, object types* $OT = \{\pi_{type}(o) \mid o \in O\}$*,*
- *event-object associations* $\pi_{obj} : E \to \mathcal{P}(O)$*,*
- *event-activity mappings* $\pi_{act} : E \to \mathcal{A}$*,*
- *event-timestamp mappings* $\pi_{time} : E \to \mathcal{T}$*.*

Each object can be associated with multiple events. The sequence in which events related to the same object occur establishes the precedence constraints of the event log. By merging together all precedence constraints for all objects we derive a global view of the events and how they are connected via objects' precedence constraints. As one event can be related to multiple objects, this global view is a graph.

Definition 2 (Event-Object Graph). *Let* $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *be an event log. We define the events and precedence constraints of an event log in one single object: the event-object graph* $EO_L = (E, P)$*. The nodes of the event-object graph are the events of the event log, the edges are the precedence constraints defined by the objects* $P = \{(e, e') \in E \times E \mid e \neq e' \wedge \exists_{o \in O}\ o \in \pi_{obj}(e) \wedge o \in \pi_{obj}(e') \wedge \neg\exists_{e'' \in E}\ o \in \pi_{obj}(e'') \wedge \pi_{time}(e) < \pi_{time}(e'') < \pi_{time}(e')\}$.

To analyze a process, we extract cases from the event log that describe end-to-end runs through the process. In traditional process mining, this is trivial as a case identifier directly identifies an end-to-end run through the process. In the more general case (i.e., object-centric event logs), events do not refer to one case identifier but to multiple case identifiers, i.e., objects. Case extraction refers to the notion of assigning events and their precedence constraints to different cases.

Definition 3 (Case Extraction). *Let* $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *be an event log with event-object-graph* $EO_L = (E, K)$*. To analyze the process, cases are extracted. An extraction technique* $ext(L) \subseteq \mathcal{P}^+(E)$ *extracts cases as sets of events, where events and precedence constraints are defined by cases* $C_{ext}(L) = \{(E', K') \mid E' \in ext(L) \wedge K' = E' \times E' \cap K\}$.

The result of a case extraction are, in general, graphs. In traditional process mining, cases are assumed to be sequences, however, sequences are also special cases of graphs [10]. Different case extraction methods have been proposed and discussed. Van der Aalst introduces the extraction method of flattening on a single object type, i.e., taking the objects of one object type and using their event sequences as cases for traditional process mining [2], which has also been

discussed in earlier publications addressing the data extraction from information systems with object-centricity [9]. Adams et al. propose two methods, one that uses connected components of the event-object graph and one that uses connected subgraphs of a leading object type [5], also representing the cases as graphs instead of sequences. Furthermore, Calvanese et al. describe an extraction technique that would be equivalent to the connected-components extraction of Adams et al. and squashing the resulting graph into a sequence [7]. We discuss the different extraction techniques in Sec. 6.

3 Quality Issues

Extracted cases can be subject to different quality problems. These quality problems can affect downstream process analysis by providing misleading statistics, incorrect process models, and missing information. The three quality problems are convergence, deficiency, and divergence [2].

Definition 4. *(Convergence) Let* $L=(E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *be an event log with event-object graph* $EO_L=(E, K)$ *and cases* $C_{ext}(L)=\{(E_1, K_1), \ldots (E_n, K_n)\}$. $C_{ext}(L)$ *is convergence-free iff* $\forall_{e \in E} \neg \exists_{(E_i,K_i),(E_j,K_j) \in C_{ext}(L)} (E_i, K_i) \neq (E_j, K_j) \wedge e \in E_i \wedge e \in E_j$.

If an event is contained in two cases there is a convergence problem. Duplicated events lead to issues in downstream-process mining, such as increased activity counts that trigger problems in other areas like feature engineering in predictive process monitoring.

Definition 5. *(Deficiency) Let* $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *be an event log with event-object graph* $EO_L = (E, K)$ *and extracted cases* $C_{ext}(L) = \{(E_1, K_1), \ldots (E_n, K_n)\}$. *The cases are deficiency-free iff* $\forall_{e \in E} \exists_{(E_i,K_i) \in C_{ext}(L)} e \in E_i$.

If an event is contained in no case there is a deficiency problem. Missing events lead to missing information in downstream process mining as events that could contain important hints about activities, bottlenecks, or features are not considered.

Definition 6. *(Divergence) Let* $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *be an event log with event-object graph* $EO_L = (E, K)$ *and extracted cases* $C_{ext}(L) = \{(E_1, K_1), \ldots (E_n, K_n)\}$. *The extracted cases are divergence-free iff* $\forall_{(e,e') \in E \times E} (e, e') \in K \Leftrightarrow \exists_{(E_i,K_i) \in C} (e, e') \in K_i$.

If the precedence constraints contained in the cases do not match the precedence constraints of the event log there is a divergence problem. This means, that the directly-follows relationships that are contained in the event log do not reflect the ones that are represented by the cases. By our definition of case extraction, the cases cannot contain precedence constraints that are not in the event log. Removed constraints will lead to issues with downstream process mining tasks such as discovery, as the resulting model will be based on incorrect precedence constraints.

4 Process Executions from Connected Components

In this paper, we investigate the properties of connected-components process execution extraction introduced by Adams et al. [5] with respect to quality issues. Process executions are graph-based cases that are built by using the weakly connected components of the event-object graph, i.e., merging all interdependent objects and their precedence constraints into one case.

Definition 7 (Connected-Components Extraction). *Let* $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *be an event log with event-object graph* $EO_L = (E, K)$*. The connected-component extraction extracts graph-based cases (called: process executions) as connected components of the event-object graph, i.e.,* $ext\_cc(L) = \{E' \in \mathcal{P}^+(E) \mid e, e' \in E' \Leftrightarrow e \sim_{EO} e'\}$.

All events that are connected to each other via a path in the event-object graph are grouped into one process execution. All precedence constraints for which both events are in this group are added to the process execution.

5 Only Connected Components Mitigate Quality Issues

In this section, we prove that only execution extraction based on connected components is free of convergence, deficiency, and divergence issues. To do so, we prove that connected-components extraction does not have any quality problems and, subsequently, prove that all techniques not employing connected components have quality issues.

5.1 Connected Components Mitigate Quality Issues

Theorem 1. *Let* $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *be an object-centric event log and let* $C_{ext\_cc}(L) = \{(E_1, K_1), \dots (E_n, K_n)\}$ *be extracted process executions with connected-components extraction. Then,* $C_{ext\_cc}(L)$ *does not suffer from convergence, divergence, or deficiency problems.*

We prove this theorem by proving that connected-components extraction is subject to neither convergence, deficiency, or divergence. For each of those, we show that extracted cases with the corresponding quality issue cannot stem from connected-components extraction.

Lemma 3 (Convergence Free). *Let* $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *be an event log and let* $C = C_{ext}(L) = \{(E_1, K_1), \dots (E_n, K_n)\}$ *be extracted cases using* $ext(L)$ *such that* $\exists_{e \in E}\ \exists_{(E_i,K_i),(E_j,K_j) \in C}\ (E_i, K_i) \neq (E_j, K_j) \wedge e \in E_i \wedge e \in E_j$ *(a convergence problem is present). Then* $C \neq C_{ext\_cc}(L)$*, i.e.,* C *cannot stem from a connected-components extraction.*

Proof. We prove our lemma by contradiction: Suppose we have an event that is part of two cases, i.e., a convergence problem $\exists_{e \in E}\ \exists_{(E_i,K_i),(E_j,K_j) \in C}\ (E_i, K_i) \neq (E_j, K_j) \wedge e \in E_i \wedge e \in E_j$ and both cases would stem from connected-components extraction $(E_i, K_i), (E_j, K_j) \in ext_{cc}(L)$. For both cases, their events would be constructed by adding all events reachable from event e, according to the connected-components extraction, i.e., $E_i = \{e' \in E \mid e \sim_{EO} e'\}$ and $E_j = \{e' \in$

$E \mid e \sim_{EO} e'\}$, i.e., $E_i = E_j$. Thus, $(E_i, K_i) = (E_j, K_j)$ which conflicts with the definition of convergence. Therefore, an extraction with convergence issues cannot stem from connected-components extraction.

Lemma 4 (Deficiency Free). *Let $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ be an event log and let $C = C_{ext}(L) = \{(E_1, K_1), \ldots (E_n, K_n)\}$ be extracted cases using $ext(L)$ such that $\exists_{e \in E} \neg\exists_{(E_i,K_i)\in C}\ e \in E_i$ (a deficiency problem is present). Then $C \neq C_{ext\_cc}(L)$, i.e., C cannot stem from a connected-components extraction.*

Proof. We prove this lemma by contradiction: We assume that $\{(E_1, K_1), \ldots, (E_n, K_n)\} \subseteq C_{ext\_cc}(L)$ are cases retrieved from connected-components extraction and there is an event that is not part of a case $\exists_{e\in E} \neg\exists_{(E_i,K_i)\in C_{ext\_cc}(L)}\ e{\in}E_i$ Therefore, the event is not part of any case $\forall_{(E_i,K_i)\in C_{ext\_cc}(L)}\ e \notin E_i$. Due to the equivalence relation of connected-components extraction, it must hold that e has n opath to any other event $\forall_{(E_i,K_i)\in ext_{cc}(L)}\ \forall_{e'\in E_i} e \nsim_{EO} e'$. However, e has a path to itself $e \sim_{EO} e$, therefore, e must appear in one of the cases extracted by connected components and cases suffering from the deficiency problem cannot stem from connected-components extraction.

Corollary 1. *Let $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ be an event log and $\{(E_1, K_1), \ldots (E_n, K_n)\} = C_{ext\_cc}(L)$ be cases form connected-components extraction. Since they do not have convergence or divergence issues $E{=}E_1 \cup \cdots \cup E_n$.*

If no event is missing, the set of events included in the cases corresponds exactly to the events in the event log.

Lemma 5 (Divergence Free). *Let $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ be an event log and let $C = C_{ext}(L) = \{(E_1, K_1), \ldots (E_n, K_n)\}$ be extracted cases using $ext(L)$ such that $\exists_{(e,e')\in E\times E}\ (e, e') \in K \nLeftrightarrow \exists_{(E_i,K_i)\in C}\ (e, e') \in K_i$ (a divergence problem is present). Then $C \neq C_{ext\_cc}(L)$, i.e., C cannot stem from a connected components extraction.*

Proof. We prove that an extraction with a divergence problem cannot stem from connected-components extraction by proving both directions of the equivalence relation hold for connected components: First ($\Rightarrow$), a precedence constraint present in the event log must be in the cases and, second ($\Leftarrow$), a precedence constraint present in the cases must be in the event log. For the first, we assume that $\{(E_1, K_1), \ldots (E_n, K_n)\} \in ext_{cc}(L)$ and show a contradiction.

1) $\Rightarrow$ Suppose there is a precedence constraint in the event log, i.e., the event-object graph, but not in the cases $\exists_{(e,e')\in E\times E}(e, e') \in K \nRightarrow \exists_{(E_i,K_i)\in C}\ (e, e') \in K_i$. This precedence constraint connects two nodes e and e'. Since all events are covered in connected-components extraction $E = E_1 \cup \ldots \cup E_n$ the event e needs to be in one case $\exists E_i \in \{E_1, \ldots, E_n\}$ such that $e \in E_i$. Since e has a path to e', $e \sim_{EO} e' = \langle e, e' \rangle \neq \perp$, it holds that e' must also be in the same case $e' \in E_i$. However, since the precedence constraints of the case are defined through the precedence constraints of the events in the event log $K_i = E_i \times E_i \cap K$ and $e, e' \in E_i$ and $(e, e') \in K$ it holds that the precedence constraint must be

contained in the event log $(e, e') \in K_i$. This is a contradiction. Therefore, an extraction where a precedence constraint is present in the event log but not in the cases cannot stem from connected-components extraction.

2) By definition, the precedence constraints of the extracted cases are a subset of the ones in the event log. Therefore, this direction holds.

5.2 Only Connected Components Mitigate Quality Problems

As we have shown that connected-components extraction does not suffer from convergence, deficiency, or divergence issues, we want to examine other extraction techniques in this section. We prove that all techniques not building process executions from connected-components have quality issues.

Theorem 2. *Let $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ be an event log and let $C = C_{ext}(L) = \{(E_1, K_1), \dots (E_n, K_n)\}$ be extracted cases with another extraction than connected-component extraction. Then, C is either composed of multiple connected components or suffers from a convergence, divergence, or deficiency problem.*

We prove this theorem by deconstructing the definition of connected-components extraction to list all ways in which other extraction techniques can differ from it. These two ways are events connected by a path ending up in different cases and events without a path ending up in the same case. For both these ways, we show that it always introduces quality problems or boils down to an extraction where multiple connected-components are grouped to one process execution.

Proof. Let $L=(E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ be an event log and let $C = C_{ext}(L) = \{(E_1, K_1), \dots (E_n, K_n)\}$ be extracted cases with another extraction technique than connected-component, i.e., $\exists_{(E_i,K_i)\in C}\ e, e' \in E_i \nLeftrightarrow e \sim_{EO} e'$. We break this down into the possible conditions that would fulfill a non-connected-components extraction. First, it may include two events in a case that are not connected by a path $\exists_{(E_i,K_i)\in C}\ e, e' \in E_i \wedge e \nsim_{EO} e'$, or, second, it may exclude an event with a path to another event from the same case $\exists_{(E_i,K_i)\in C}\ \{e, e'\} \nsubseteq E_i \wedge e \sim_{EO} e'$. For both possible conditions, we will individually prove that there is a convergence, divergence, or deficiency problem.

1) $\exists_{(E_i,K_i)\in C}\ e, e' \in E_i \wedge e \nsim_{EO} e'$ then C has a convergence, divergence, or deficiency issue.
When two events are contained in the same case without a path $\exists_{(E_i,K_i)\in C}\ e, e' \subseteq E_i \wedge e \nsim_{EO} e'$ we apply the transitivity lemma (cf. Lemma 2) to show that these two events are connected to a disjoint set of other events, $E_1 = \{e'' \in E' \mid e \sim_{EO} e'' \vee e'' \sim_{EO} e\}$ and $E_2 = \{e'' \in E' \mid e' \sim_{EO} e'' \vee e'' \sim_{EO} e'\}$ with $E_1 \cap E_2 = \emptyset$. For these two event sets, we define a tautology comprising a statement and its negation, implying that either the statement or its negation must be satisfied. $\forall_{E''\in\{E_1,E_2\}}$
a) $\neg\exists_{e\in E\setminus E''}\forall_{e'\in E''}\ e \sim_{EO} e' \vee e' \sim_{EO} e$ or
b) $\exists_{e\in E\setminus E''}\forall_{e'\in E''}\ e \sim_{EO} e' \vee e' \sim_{EO} e$.

The first part of the tautology states that there is no event outside this set that has a path to this set's elements. The second part states that an event outside a set element has a path to that set's elements. For both parts of the tautology, we show there is a convergence, deficiency, or divergence problem or it boils down to connected-components extraction if it is fulfilled:

a) E'' constructs a connected component and does not induce any quality problem (cf. Theorem 1).
b) Here we can directly apply the connecting-edge lemma (cf. Lemma 1). $\exists_{e \in E \setminus E''} \exists_{e' \in E''} \; e \sim_{EO} e' \Rightarrow \exists_{(e_1,e_2) \in (E \setminus E'' \times E'')} \; (e_1, e_2) \in K$. With $e_1 \sim_{EO} e_2 \wedge e_2 \in E''$ it holds that $e_1 \notin E'$, as e_1 would be part of E'' if it would be in E' since it is connected to nodes of E''. Therefore, $(e_1, e_2) \notin E' \times E'$ and following that $(e_1, e_2) \notin E' \times E' \cap K = K'$. Therefore, a divergence problem is present unless both events are part of another case, which would introduce a convergence problem.

2) $\exists_{(E_i,K_i) \in C} \; e \in E_i \wedge e' \notin E_i \wedge e \sim_{EO} e'$ then C has a convergence, divergence, or deficiency issue.

When considering two events connected by a path but not included in the same case, we extend the formula by adding a tautology with a statement and its negation $\exists_{(E_i,K_i) \in C} \; e \notin E_i \wedge e' \in E_i \wedge e \sim_{EO} e' \wedge$

a) $(\neg \exists_{E'' \in (E_1,\ldots,E_n)} \; e \in E'' \vee$
b) $\exists_{E'' \in (E_1,\ldots,E_n)} \; e \in E'')$

The first part states that the event connected by a path but not in the case is also not in the other case. The second part states that there is another case that contains the event. Either one of the two needs to be fulfilled, therefore, we will show that both lead to convergence, divergence, or deficiency.

a) This corresponds to the definition of a deficiency problem, i.e., an event is in no case.
b) It holds that $e \in E'' \wedge e \notin E'$. We can generalize this such that $e \in E \setminus E'$ and apply Lemma 1. $e \in E \setminus E' \wedge e' \in E' \wedge e \sim_{EO} e' \Rightarrow \exists_{(e_1,e_2) \in E \setminus E' \times E'} \; (e_1, e_2) \in K$. Since $e_1 \notin E'$ it also holds that $(e_1, e_2) \notin E' \times E'$ and $(e_1, e_2) \notin E' \times E' \cap K$. Therefore, the edge (precedence constraint) cannot be included in the case (E', K'). To avoid a divergence problem, it would need to be included in another case. However, then both events of the edge would need to be included in the case, leading to a convergence issue since e appears in both cases. Therefore, either a convergence or divergence issue is present.

6 Discussion, Limitations, and Implications

In this section, we discuss the relationship of our proof to traditional process mining, draw limitations of our proof for practice, and derive the implications for future process mining research.

6.1 Practical Limitations

Our paper has shown that case extraction can only conform to the conditions of convergence-freeness, divergence-freeness, and deficiency-freeness when using connected-components extraction. While these are important and foundational

Table 1: Different case extraction techniques and their properties. Divergence issues that are only introduced by deficiency issues are depicted with (✓)

Extraction Technique	Convergence-Free	Divergence-Free	Deficiency-Free
Single-Type Flattening [2]			
Composite-Type Flattening [7,4]	✓		✓
Leading-Type Extraction [5]		(✓)	
Maximal-Type-Set Extraction	✓	(✓)	
Connected-Components Extraction [5]	✓	✓	✓

criteria for the correctness of event data, there are two major factors to take into consideration: First of all, for some downstream process mining tasks, the presence of quality problems might not affect the quality of the results. For example, when computing the cycle time of different objects, e.g., orders that are placed and then delivered, it would not matter if some events that are shared between orders are multiplied, as this does not affect the computation of cycle times. Second, one might have additional conditions when extracting cases, rendering the problem of fulfilling these conditions and mitigating quality problems infeasible. For example, requiring exactly one sales order object per case would conflict with connected-components extraction if one connected component contains two sales orders. *This means, that mitigating convergence, divergence, and deficiency is often infeasible in practice.* For such cases, a trade-off between different quality issues has to be made. We provide a collection of current alternative case extraction techniques along with associated quality issues in Table 1. These techniques could also be employed if their underlying quality problems would not affect the results of the employed analysis.

Techniques enforcing sequential, traditional cases are called flattening. All other listed techniques produce graph-based process executions. As enforcing sequentiality tempers with precedence constraints, flattening techniques can never guarantee divergence-freeness. Deficiency implies divergence, as missing events lead to missing precedence constraints. Therefore, we depict a technique that is only subject to divergence issues introduced by deficiency issues with a checkmark in parentheses.

As discussed by van der Aalst [2], single-type flattening is subject to any of the three quality problems. When choosing a single object type and considering the event sequence of each object as a case, events might get duplicated, missing, and precedence constraints get left out. However, single-type flattening can still offer valuable insights into the subprocess of a single object type, especially when considering that the whole traditional process mining pipeline can be applied.

Composite-type flattening describes an extraction technique that collects connected objects and merges their events into an event sequence [7]. This is equivalent to compressing the results of connected-component extraction by Adams et al. [4] into a sequence. As it builds on connected-components, it is free of convergence and deficiency issues. Therefore, it can produce valuable

insights when precedence constraints are not important. However, divergence issues might be very consequential for some tasks like discovery.

Leading-type extraction extracts subgraphs of the connected components that are associated with objects of a leading type and their closest related objects of other types [5]. As the closest objects of one leading object can also be the closest objects of another object (i.e., two sales orders having the same delivery object), this extraction is subject to convergence. Some objects can be left out, e.g., if they are not connected to any leading object, introducing deficiency and, therefore, also divergence problems. These quality problems are further illustrated in [6]. However, the precedence constraints within the process execution are correctly represented. Furthermore, this extraction technique allows to incorporate related objects, in contrast to single-type flattening.

Maximal-type-set extraction is a new extraction technique suggestion based on the results of this paper. Consider a setting where a user has an additional condition that the extracted cases should satisfy. An extraction technique could eliminate a minimal set of object types such that the connected-components extraction satisfies the condition. Under this consideration, maximal-type-set extraction would only be subject to deficiency and deficiency-induced divergence and could be an alternative to single-type flattening since it can incorporate information about related objects.

6.2 Traditional Event Logs and Cases

Traditional event logs, where each event is associated with precisely one object and all objects are of the same type, are a special case of object-centric event logs. When constructing the event-object graph for such logs, each object creates its own weakly-connected component, as no events are shared between objects. Consequently, connected-components extraction aligns with the traditional case notion, resulting in one event sequence per object. This underscores two points: First, connected-components extraction is backward compatible with traditional process mining, i.e., if the input is a traditional event log, the process executions correspond to cases. Second, if the information system's underlying events adhere to the assumption of exactly one object per event, traditional process mining does not encounter convergence, deficiency, or divergence problems. This assumption holds true for some information systems, like ticketing or case management systems [12]. However, for the vast majority of information systems this does not hold true, i.e., quality problems are to be expected when extracting traditional cases, i.e., sequences.

6.3 Implications

The most important implication of our paper is that quality issues are unavoidable if there are conflicting requirements for connected components. Depending on the planned process analysis, this could have significant effects on the quality of the results. Based on this, the second implication of our work is that many real-life applications of object-centric process mining will have to make a trade-off between different quality issues using different extraction techniques.

As we currently can only make qualitative statements (cf. Table 1), a quantitative evaluation of quality issues for different extraction techniques is necessary. Furthermore, an overview of which process analysis tasks are negatively affected by which quality issues is necessary. Based on the trade-off spectrum between different quality issues, new extraction techniques that inhibit unoccupied parts of the spectrum can be proposed. These developments will bring transparency and capabilities to dealing with quality issues in object-centric process mining.

We derive secondary implications from aligning connected-components extraction with the traditional case notion: New extraction techniques should provide the traditional case concept when applied to traditional event logs and adaptations of traditional process mining algorithms to the object-centric setting should be backward compatible, i.e., if the object-centric event log is a traditional event log, the results of the algorithm should be consistent with the traditional process mining algorithm. This ensures the consistency of the process mining field when moving towards object-centricity. Examples of such backward-compatible process mining algorithms are the discovery of object-centric Petri nets [3], which yields a standard Petri net when fed with a traditional event log, object-centric variants [5], which produce standard variants when fed with a traditional event log, or object-centric features [4] which would return standard features when fed with a traditional event log. This ensures the consistency of the process mining field.

7 Conclusion

This paper addressed the issue of quality problems in process mining when extracting cases. We proved that only case extraction based on connected components is free of convergence, deficiency, or divergence issues. For process analysis tasks that are negatively affected by quality issues, connected components can yield the only case extraction that mitigates the negative impacts of convergence, divergence, and deficiency. If there are other conditions that the extracted cases should fulfill and these conditions conflict with connected-components extraction, mitigating quality problems is infeasible. In that situation, case extraction is a trade-off between different quality issues. Furthermore, we have discussed that connected-components extraction aligns with the traditional case notion of traditional event logs. This necessitates object-centric adaptations of traditional process mining algorithms to be backward compatible, ensuring the consistency of the process mining field.

For future work, there should be an in-depth analysis of the impact of different quality issues on different process analysis tasks, as well as a quantification of quality issues for different case extraction techniques. These contributions would deliver transparency to anyone applying case extraction to conduct process analysis tasks by understanding the presence of quality issues, their effect on the quality of results, and the trade-offs that can be achieved by utilizing different case extraction techniques.

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016). https://doi.org/10.1007/978-3-662-49851-4
2. van der Aalst, W.M.P.: Object-centric process mining: Dealing with divergence and convergence in event data. In: SEFM. pp. 3–25. Springer (2019). https://doi.org/10.1007/978-3-030-30446-1_1
3. van der Aalst, W.M.P., Berti, A.: Discovering object-centric petri nets. Fundam. Informaticae **175**(1-4), 1–40 (2020). https://doi.org/10.3233/FI-2020-1946
4. Adams, J.N., Park, G., Levich, S., Schuster, D., van der Aalst, W.M.P.: A framework for extracting and encoding features from object-centric event data. In: ICSOC. pp. 36–53. Springer (2022). https://doi.org/10.1007/978-3-031-20984-0_3
5. Adams, J.N., Schuster, D., Schmitz, S., Schuh, G., van der Aalst, W.M.P.: Defining cases and variants for object-centric event data. In: ICPM. pp. 128–135. IEEE (2022). https://doi.org/10.1109/ICPM57379.2022.9980730
6. Adams, J.N., van Zelst, S.J., Rose, T., van der Aalst, W.M.P.: Explainable concept drift in process mining. Inf. Syst. **114**, 102177 (2023). https://doi.org/10.1016/j.is.2023.102177
7. Calvanese, D., Montali, M., Syamsiyah, A., van der Aalst, W.M.P.: Ontology-driven extraction of event logs from relational databases. In: BPM Workshops. pp. 140–153. Springer (2015). https://doi.org/10.1007/978-3-319-42887-1_12
8. Fahland, D.: Process mining over multiple behavioral dimensions with event knowledge graphs. In: Process Mining Handbook, pp. 274–319. Springer (2022). https://doi.org/10.1007/978-3-031-08848-3_9
9. Gerke, K., Mendling, J., Tarmyshov, K.: Case construction for mining supply chain processes. In: BIS. pp. 181–192. Springer (2009). https://doi.org/10.1007/978-3-642-01190-0_16
10. Leemans, S.J.J., van Zelst, S.J., Lu, X.: Partial-order-based process mining: a survey and outlook. Knowl. Inf. Syst. **65**(1), 1–29 (2023). https://doi.org/10.1007/s10115-022-01777-3
11. Lu, X., Nagelkerke, M., van de Wiel, D., Fahland, D.: Discovering interacting artifacts from ERP systems. IEEE Trans. Serv. Comput. **8**(6), 861–873 (2015). https://doi.org/10.1109/TSC.2015.2474358
12. Weerdt, J.D., Wynn, M.T.: Foundations of process event data. In: Process Mining Handbook, pp. 193–211. Springer (2022). https://doi.org/10.1007/978-3-031-08848-3_6