Extended investigation on SOTA (Van Hulzen et al. 2021)

We carried out an extended investigation to study the comparison of our proposed solution, i.e., Simulated Annealing (SA)-based method, to a state-of-the-art solution in the process mining literature.

SOTA: Activity instance archetypes in ResProMin

We first investigated the scope of comparable SOTA methods. As explained in Section II (Related Work), the most relevant method is the **identification of activity instance archetypes** in the ResProMin framework (Van Hulzen et al. 2021, cited as Ref. [9] in the manuscript).

The ResProMin framework (Van Hulzen et al. 2021) aims at identifying resource profiles. It uses activity instance archetypes as a key construct. However, as we focus on work specialization as the grouping of tasks/activities, we choose activity instance archetypes identification as the comparable SOTA method, but not including the identification of resource profiles. See Figure 1 for the idea.

Specifically, a set of activity instance archetypes prescribes a clustering of events, which is comparable to a set of execution contexts generated from our solution.

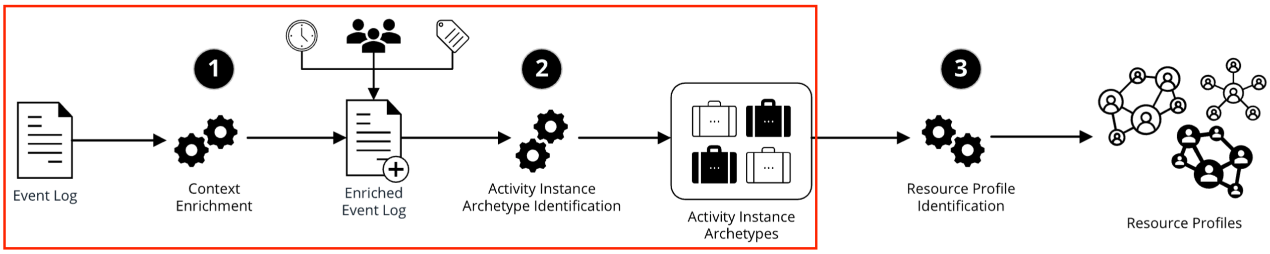

Figure 1: An illustration of the ResProMin framework (Van Hulzen et al. 2021), annotated based on Figure 1 in the original paper. The red rectangle represents the scope of their solution as comparable SOTA in our work, as we focus on work specialization as the grouping of tasks.

Conceptual difference and its implication

There are fundamental differences between *execution contexts* in our work and *activity instance archetypes* in (Van Hulzen et al. 2021) as two representations to model work specialization. Section II (Related Work) provides the detail explanation. In summary, a major difference is that a set of execution contexts are defined based on "types". They can be used to organize events in an OLAP-like view and can be explicitly linked to one another. A set of activity instance archetypes can be used to organize events based on probabilities based on finite mixture models, but the relations between archetypes are not defined.

Given the conceptual difference, in order to enable a comparison between our method and the SOTA, it is necessary to adapt either SOTA or our method.

Comparison based on impurity and dispersal (using measures in our work)

The first option is to adapt activity instance archetypes (SOTA). Specifically, to compute impurity and dispersal of a set of activity instance archetypes, each "archetype" (i.e., a cluster of events) need to be assigned with some artificial case type name, activity type name, and time type name. This can be achieved by constructing for each archetype an execution context with its unique case type name, activity type name, and time type name. For example, if there are three activity instance archetypes (three clusters of events), then we would have three execution contexts as follows:

Archetype	Execution Context (EC)	EC case type	EC activity type	EC time type
1	(C1, A1, T1)	C1	A1	T1
2	(C2, A2, T2)	C2	A2	T2
3	(C3, A3, T3)	C3	A3	T3

Note that however, as previously mentioned in the conceptual difference, activity instance archetypes are based on finite mixture models, and it is therefore not possible to guarantee exact partitioning of case identifiers, activity names, and timestamps as required by Definition 3 in the manuscript. For instance, {A1, A2, A3} in the example above do not guarantee partitioning of the possible values of activity names in an event log into three sets. Still, this adaption would allow computing impurity and dispersal of activity instance archetypes.

We performed the following additional experiments where activity instance archetypes are identified from an input event log and evaluated after the abovementioned adaptation.

Implementation of SOTA

We referred to the original software implementation of ResProMin available online: https://doi.org/10.5281/zenodo.4606757 The implementation is in R and uses the R-package *flexmix* (documentation available at: https://cran.r-project.org/web/packages/flexmix/flexmix.pdf).

Extended experiment datasets and method configuration

In (Van Hulzen et al. 2021), the identification of activity instance archetypes (SOTA) is only demonstrated on the BPIC 2015 dataset, with each municipality recorded in an individual event log. In this extended experiment, we followed the experiment design in our manuscript (see Section V) and attempted to test the SOTA method on five event logs.

Two aspects need to be considered for the configuration of the SOTA method.

1. Selecting event attributes for context enrichment: We followed the same selections as how we had selected event attributes when evaluating our method. This is reported in detail in Section V of our manuscript and well as the corresponding experiment documentation (https://royjy.me/to/learn-co).

2. Configuring parameters for fitting mixture models: As (Van Hulzen et al. 2021) does not contain experiment results on dataset other than BPIC 2015 or provide suggestions about the configuration of parameters according to the dataset characteristics, we chose to use the parameter settings as-is (cf. Section 4.1 in Van Hulzen et al. 2021 and the source code file "bpic_15_cluster.R" in their online appendix).
   - The number of components `k` is set to be tested as a range `[2, 10]`.
   - The number of initializations `nrep` is set to `5`.
   - The `tolerance` threshold for the minimum change of log-likelihood during Expectation-Maximization is set to `1e-06`.
   - The maximum number of steps during Expectation-Maximization is set to `500`.

In addition, we attempted to run the identification of activity instance archetypes (SOTA) `20` times per each of the five event logs. This is to avoid arbitrariness in the results due to the use of random seeds.

Experiment environment

We carried out the extended experiment on the Aquarius ("Aqua") HPC at Queensland University of Technology, which consists of the AMD EPYC 9684X CPUs for CPU jobs (https://docs.eres.qut.edu.au/about-aqua). Note that the software implementation of SOTA uses only single CPU core in computing.

We ran the SOTA method with a wall time limit of 24 hours. In other words, if no result was obtained within a 24-hour period for a dataset, we did not perform further experiment on that dataset.

Extended experiment results and discussion

We found that experiments on the three large event logs (BPIC 2015, BPIC 2017, BPIC 2018; all with more than 250,000 events after preprocessing. Cf. Table II in our manuscript) all exceeded the wall time limit. The HPC records are exported and stored under folder **oversized_logs/**.
We report on results obtained from the two smaller event logs (sepsis and WABO).

Log sepsis

Following the SOTA paper (Van Hulzen et al. 2021), we determined the selection of number of components `k` based on the elbow method over BIC (mean value over 20 runs) as well as the minimum number of events in a cluster (see Figure 2).

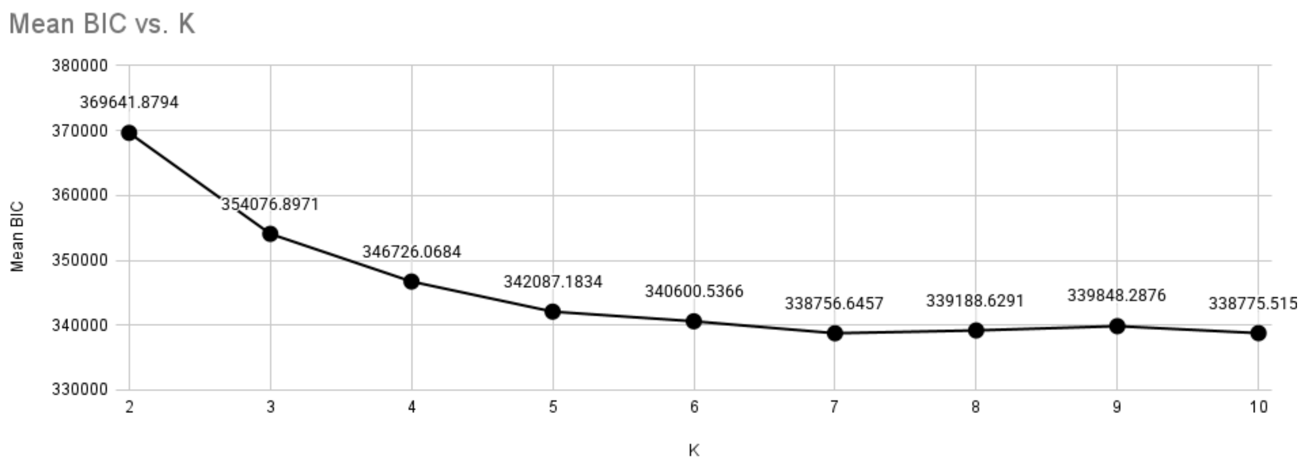

*Figure 2: Determining the suitable number of components `k` as `5` for log sepsis using the elbow method. Note that it was also ensured that the minimum number of events in any cluster is at least 5% of the size of the event log (Van Hulzen et al. 2021).

The best suitable `k` was determined as `5`. Figure 3 show the comparison between the SOTA method and our proposed SA-based method, in terms of quality measures: impurity, dispersal, and the quality score. It can be seen that the SA-based method has lower impurity and dispersal, and thus a higher quality score.

Figure 3: Comparison between SOTA and our proposed SA-based method on log sepsis, based on impurity (left), dispersal (middle), and the quality score (right).

In addition:

- the mean CPU time taken for each run of the SOTA method is `644.25` seconds;
- the mean CPU time for each run of our proposed SA-based method is `2570.20` seconds.

Log WABO

For log WABO, we only managed to obtain valid outputs for the cases of `k=2` and `k=3`. For all other `k` values, we encountered the NA log-likelihood problem across all 20 runs, each with 5 random initializations. As there is no further information in the SOTA paper or its software implementation or the original R-package *flexmix* documentation, we decided to use include both the `k=2` and `k=3` results for comparison, instead of choosing one specific `k` value. The HPC records for all other `k` values are exported and stored under folder **invalid_outputs/**.

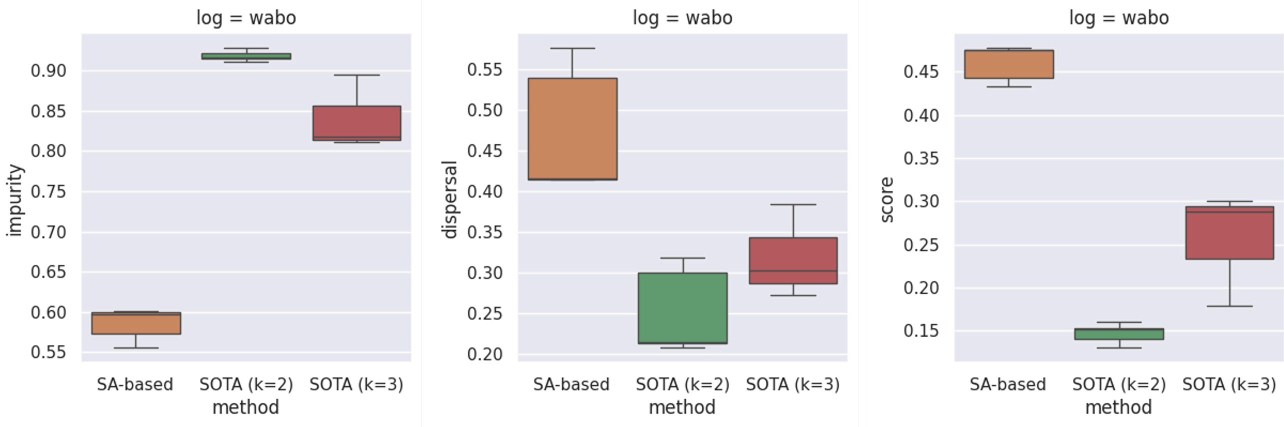

Figure 4: Comparison between SOTA and our proposed SA-based method on log WABO, based on impurity (left), dispersal (middle), and the quality score (right).

It can be seen that the SA-based method has lower impurity. In terms of dispersal, SA-based is higher (worse), however this is due to the small size of the execution contexts converted from the SOTA output (size `2` and `3`, respectively). Overall, SA-based has a higher quality score.

In addition:

- the mean CPU time taken for each run of the SOTA method is measured as `982.95` seconds (for `k=2`) and `549.70` seconds (for `k=3`);
- the mean CPU time for each run of our proposed SA-based method is `131.90` seconds.

Comparison based on mixture density function (using measure in SOTA)

It is not possible to "convert" a set of execution contexts into a set of activity instance archetypes, since those archetypes are defined based on finite mixture models that cannot be directly created from a set of execution contexts. Therefore, it was not possible to evaluate our method based on the metrics employed in (Van Hulzen et al. 2021)

Reference:

- Van Hulzen, G., Martin, N., & Depaire, B. (2021). Looking Beyond Activity Labels: Mining Context-Aware Resource Profiles Using Activity Instance Archetypes. In A. Polyvyanyy, M. T. Wynn, A. Van Looy, & M. Reichert (Eds.), *Business Process Management Forum - BPM Forum 2021, Rome, Italy, September 06-10, 2021, Proceedings* (pp. 230–245). Springer.