Mining Partially-Ordered Episode Rules With the Head Support

Philippe Fournier-Viger[1], Yangming Chen[1], Farid Nouioua[2], Youxi Wu[3]

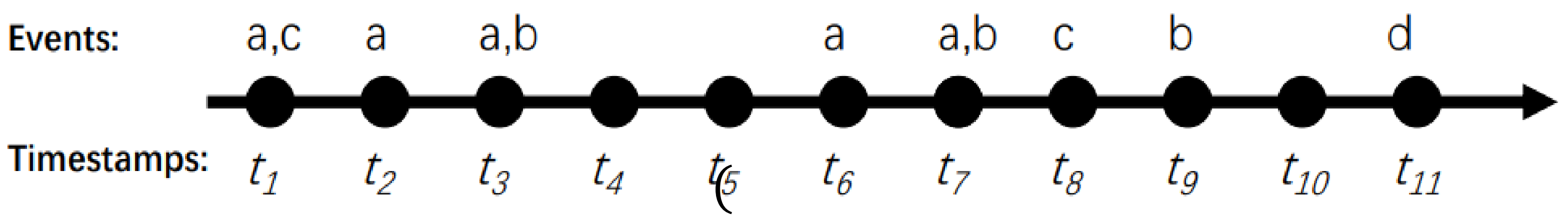
1 Harbin Institute of Technology (Shenzhen), China
2 University of Bordj Bou Arreridj, Algeria
3 Hebei University of Technology, Tianjin, China

DAWAK 2021

1 Introduction

Frequent Episode Mining

Input: a sequence of events with timestamps



Output:

Each frequent episode such that :

minimal occurrence count ≥ minSup

For example:

episode <{a,b},{c}>

Episode types:

parallel episodes, serial episodes, complex episodes

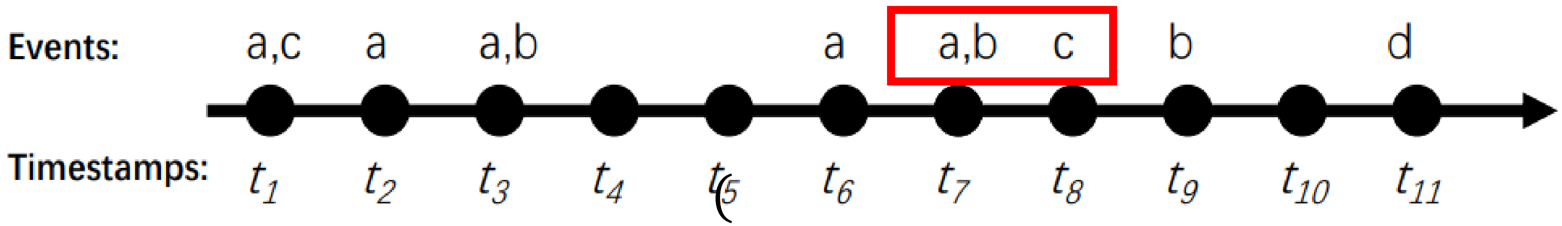
Algorithms:

WINEPI, MINEPI, EMMA, MINEPI+, ...

1 Introduction

Frequent Episode Mining

Input: a sequence of events with timestamps



Output:

Each frequent episode such that :

$minimal\ occurrence\ count \geq minSup$

For example:

episode <{a,b},{c}>

Episode types:

parallel episodes, serial episodes, complex episodes

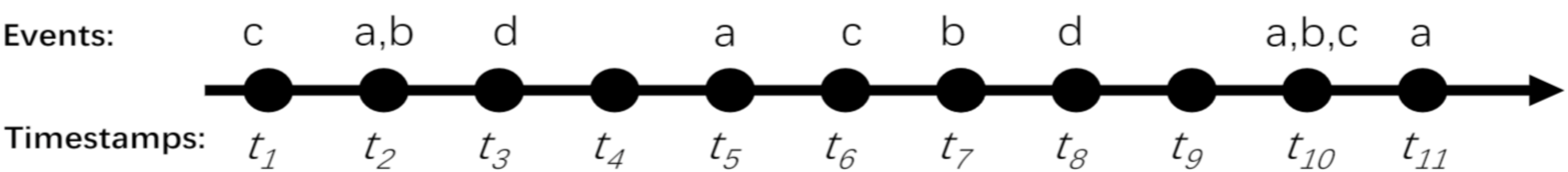
Algorithms:

WINEPI, MINEPI, EMMA, MINEPI+, ...

1 Introduction

Episode Rule Mining

Input: a sequence of events with timestamps



Output:

Each episode rule of the form $X \rightarrow Y$ such that:

(Supp(X) $\geq minSup$)

(conf(X→Y) = $\frac{\text{Supp}(X \cap Y)}{Supp(X)} \geq minConf$)

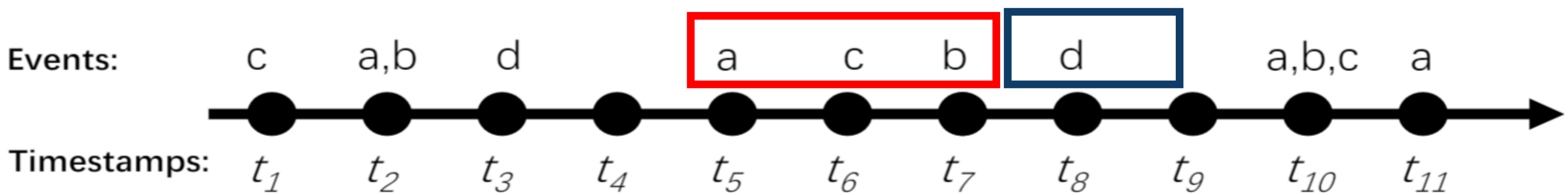
For example:

$\langle\{a\},\{c\},\{b\}\rangle \rightarrow \langle\{d\}\rangle$

1 Introduction

Episode Rule Mining

Input: a sequence of events with timestamps



Output:

Each episode rule of the form $X \rightarrow Y$ such that:

(Supp(X) $\geq minSup$)

(conf(X→Y) = $\frac{\text{Supp}(X \cap Y)}{Supp(X)} \geq minConf$)

For example:

⟨{a},{c},{b}⟩ → ⟨{d}⟩

1 Introduction

Previous Work

- **Utility-based episode rules**
  - Discovering episodes rule having a high importance in a sequence of discrete events with weights and quantities

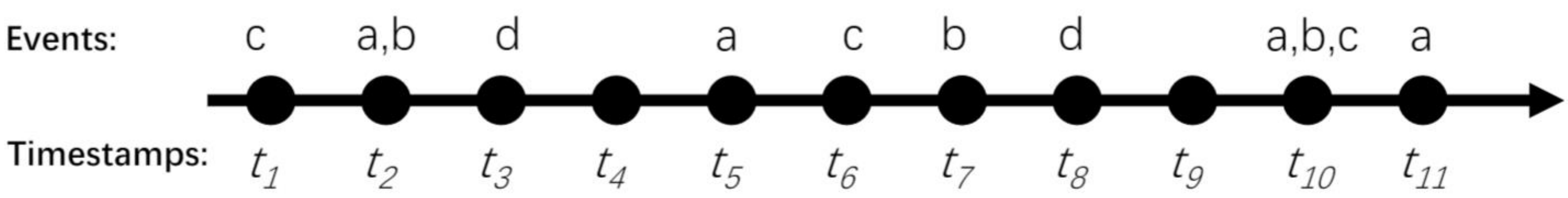
- **Precise-Positioning Episode Rules**
  (Ao et al., TKDE 2018, ICDE 2017)
  - the elapsed time between the antecedent and the consequent is fixed

To address these issues

- **Distant and essential episode rules**
  - the antecedent is as small as possible both in terms of number of events and of occurrence duration,
  - consequent is temporally distant from the antecedent.

2 Definitions

Non redundant occurrences



For Occ({a, b, c}) $=\{[t_1, t_2],[t_5, t_7],[t_{10}, t_{10}]\}$,

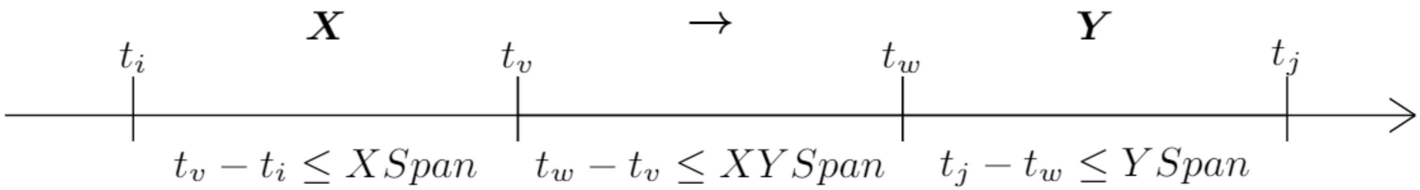
$[t_{10}, t_{11}]$ is overleaping with $[t_{10}, t_{10}]$, so $[t_{10}, t_{11}]$ is redundant to Occ({a, b, c})

$\{[t_1, t_2],[t_5, t_7],[t_{10}, t_{10}]\}$, is the set of non redundant occurrence of event set {a, b, c}

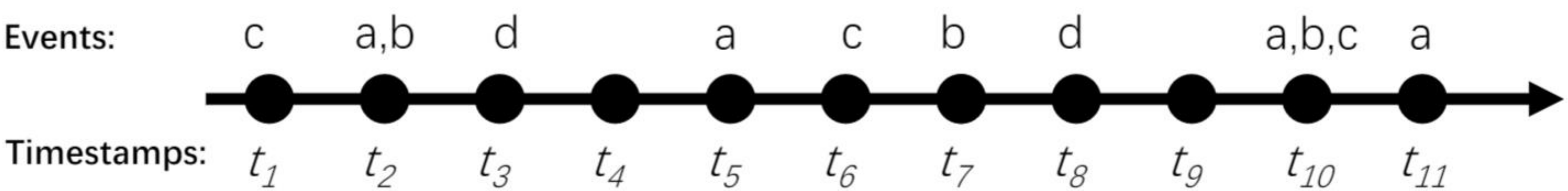
2 Definitions

Partially-Ordered Episode Rules（POER）

X - > Y



For example:



minsup= 3, minconf= 0.6, XSpan= 3, XYSpan= 1, YSpan= 1

Occurrences of {a, b, c} are Occ({a, b, c}) ={[t1, t2],[t5, t7],[t10, t10]}

Occurrences of rule R:{a, b, c} → {d} are occ(R) = {[t1, t3],[t5, t8]}

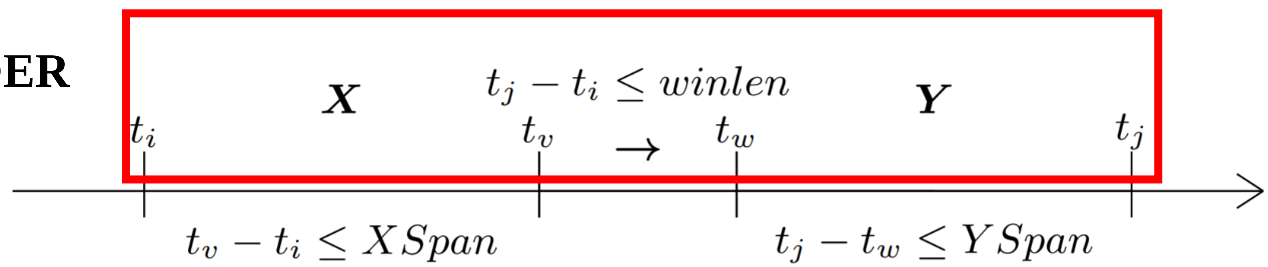
Efficient algorithm: POERM (ACIIDS 2021)

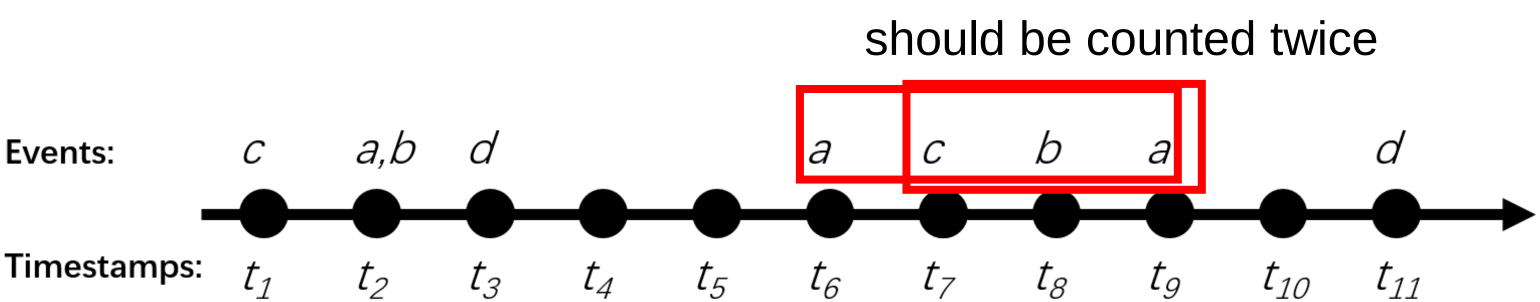
2. Definitions

POER with Head support

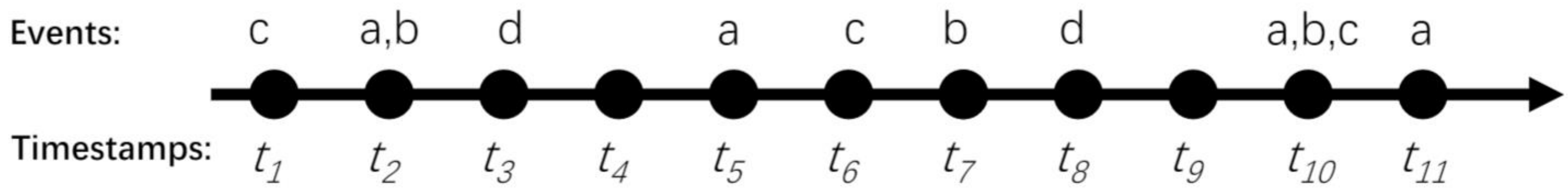
Observation: selecting POERs by counting their minimal occurrences can lead to undesirable results.
Solution: use the **Head Frequency** to calculate the support of POERs.

Head Frequency for POER





3. The POERMH algorithm



STEP 1: Find the frequent rule antecedents

minsup= 3

↓

Frequent 1-event sets : {a} , {b}, {c}

↓

extends i-event sets into i+1-event sets

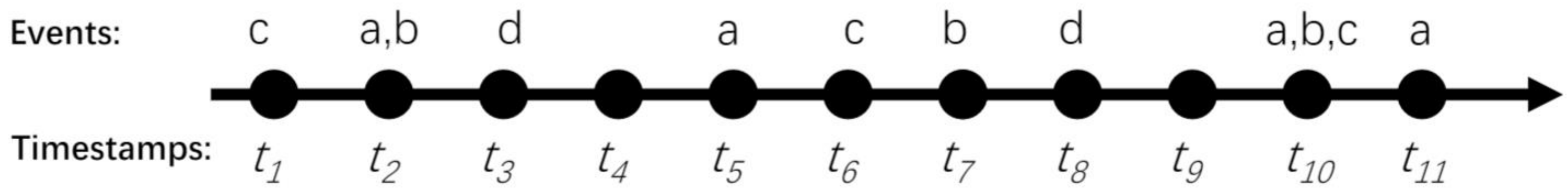
↓

Apply the quick sort algorithm to sort positions by ascending start point

↓

Count the Head Frequency and filter out the infrequent episodes

3. The POERMH algorithm



STEP 2: Find consequents to make rules

Frequent event sets : <{a} , {b}, {c},{a, b}, {a, c}, {b, c}, {a, b, c}>

Frequent 1-event consequent rule

Extends i-event consequent rule into i+1-event consequent

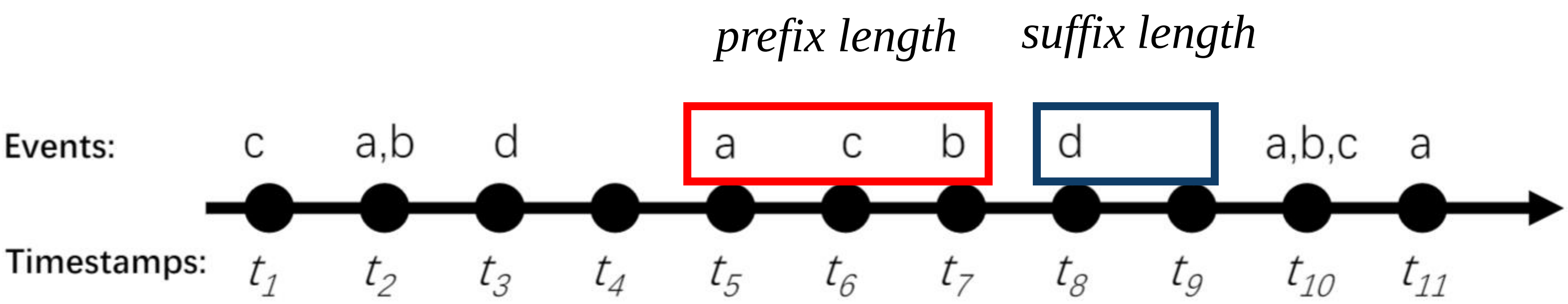
For each occurrences [pos.start, pos.end] of i-event sets X

Search for [pos.end + 1, winlen−pos.start] to extend it

4. Experiment

Sequence Prediction

Task: Predict an event that will appear in the suffix of a test sequence using the information from the prefix.



$$\text{accuracy} = \frac{\text{good predictions}}{\textit{the number of prediction opportunities}}$$

$$\text{matching rate} = \frac{\text{good predictions} + \textit{bad}\ \text{predictions}}{\textit{the number of prediction opportunities}}$$

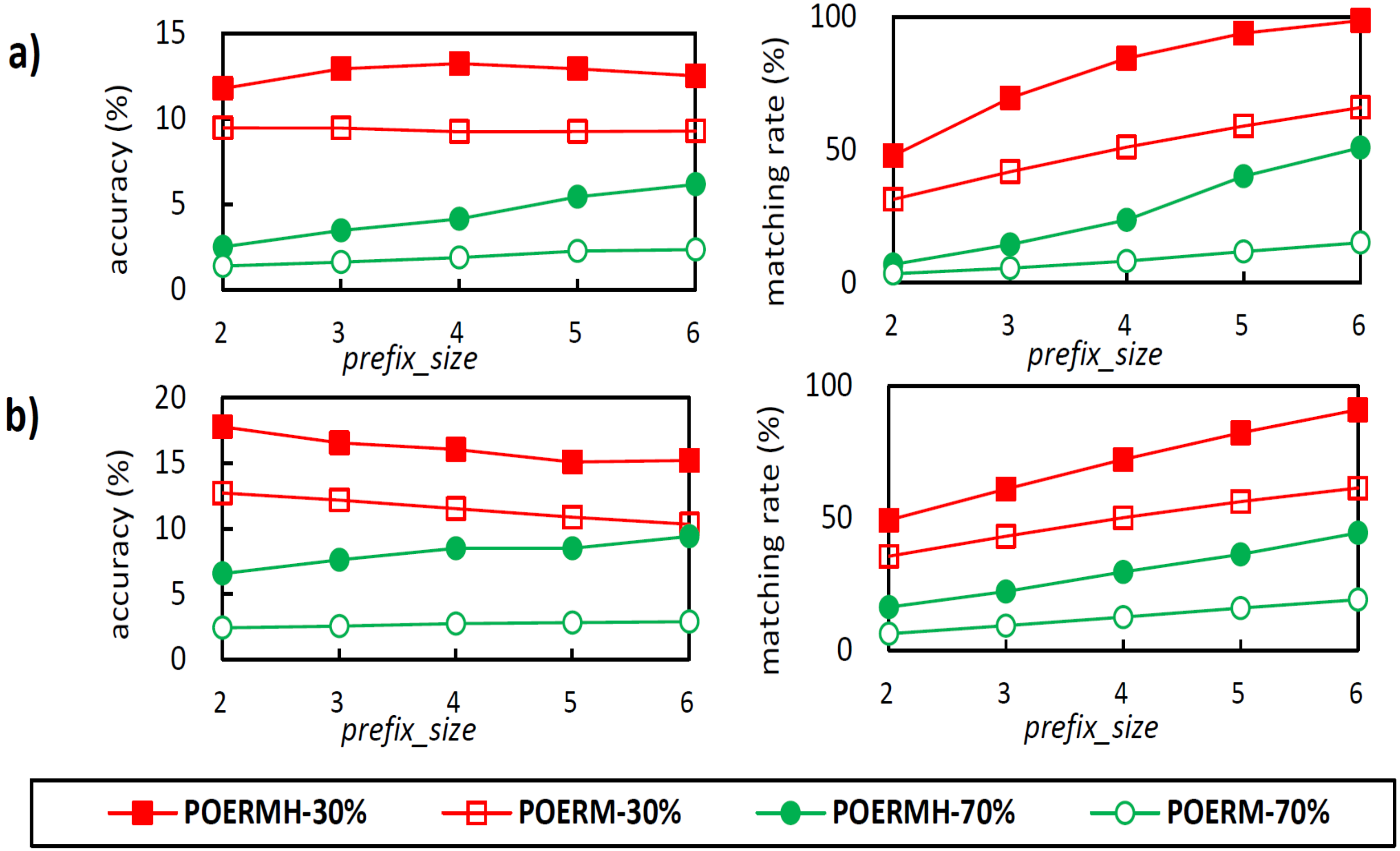
4 Experimental Evaluation

Dataset and Default Values

Dataset	# Timestamps	# Events
Bible	649,024	13,905
Leviathan	153,682	9025

Default Values	minsup	minconf	suxfixSize
Bible	120	0.3	1
Leviathan	80	0.3	2

4. Experimental Evaluation



Observation: POERM-H is more accurate and has a higher matching rate than POERM

5. Conclusion

- **Contributions:**
  - A new algorithm to find **partially-ordered episode rules** using the **head support,** named **POERMH**
  - **Experiments** have shown that the POERMH has good sequence prediction performance compared to the benchmark **POERM** algorithm.
- **Future work:**
  - Consider adding optimizations to POERMH
  - Considering more complex data types

- **Open-source code, datasets:**
  - SPMF data mining library (over 200 algorithms)

Thanks for listening!

Our recent papers on episode rule mining:

- Chen, Y., Fournier-Viger, P., Nouioua, F., Wu, Y. (2021). **Mining Partially-Ordered Episode Rules with the Head Support**. Proc. 23rd Intern. Conf. on Data Warehousing and Knowledge Discovery (DAWAK 2021), Springer, LNCS, 7 pages **[POERMH algorithm]**
- Fournier-Viger, P., Chen, Y., Nouioua, F., Lin, J. C.-W. (2021). **Mining Partially-Ordered Episode Rules in an Event Sequence.** Proc. of the 13th Asian Conference on Intelligent Information and Database Systems (ACIIDS 2021), Springer LNAI, pp 3-15 **[POERM algorithm]**
- Chen, Y., Fournier-Viger, P., Nouioua, F., Wu, Y.. (2021). **Sequence Prediction using Partially-Ordered Episode Rules**. Proc. 4th International Workshop on Utility-Driven Mining (UDML 2021), in conjunction with the ICDM 2021 conference, IEEE ICDM workshop proceedings, to appear. **[POERM and POERMH for sequence prediction]**
- Ouarem, O., Nouioua, F., Fournier-Viger, P. (2021). **Mining Episode Rules From Event Sequences Under Non-Overlapping Frequency**. Proc. 34th Intern. Conf. on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA AIE 2021), Springer LNAI, pp. 73-85. **[NONEPI algorithm]**