Resource and Activity Clustering Based on a Hierarchical Cell Formation Algorithm

Landelin Delcoucq, Thomas Dupiereux-Fettweis, Fabian Lecron, Philippe Fortemps

University of Mons (UMONS), Belgium

Abstract

In this paper, we focus on the resource perspective of process mining and more precisely on the clustering of resources sharing the same behaviors. This problematic was addressed through the use of a well-known facility layout method: cell formation. We propose an algorithm combining the resource perspective and cell formation approach to make the best use of their respective features. We wish to identify both subgroups of resources that perform similar activities and subgroups of activities performed by common resources. This new hierarchical approach provides new insights into the clustering problematic because of its bi-dimensional clustering. Experiments are considered on synthetic and real data.

1. Introduction[1]

Process mining is the science developed to model, analyze and enhance [1],[2] processes, from their real execution, through particular perspectives. The case perspective and the organizational perspective are two of the most commonly used. The first focuses on the actions performed (*the activities*) through the process from *the case* viewpoint, i.e. the point of view of its purpose. This perspective represents, for example, all the medical examinations undergone by a patient, all the forms filled out by a customer or all the steps required in the manufacturing of a product. The second focuses on a more human-centered point of view [3],[4] which aims to consider the *resources*, i.e. to describe what happens behind the process itself by considering all those who are involved in the actions. This perspective highlights how the organization that harbors the processes is structured and what the relationship between the actors is.

One of the main questions related to process mining concerns the clustering in the broadest sense of the term. The case perspective handles this through the extraction of sub-processes and the mining of activity clusters. These sub-processes are frequent groups of activities, ordered similarly, appearing frequently in the process. They can be discovered using trees combining activities and operators [5] or heat maps based on intervals of activities called sessions [6]. Activity clusters [7] are closer to common clustering ap-

[1] This version of the article has been accepted for publication, after peer review and is subject to Springer Nature's AM terms of use, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/10.1007/s10489-022-03457-9

proaches, grouping together activities using a metric based on the proximity between them. The purpose of these approaches is to aggregate activities sharing common features (intrinsic features, relationship between them, performed by common resources, etc) to provide modeling with a higher level of granularity. This allows for a better understanding and an easier visualization of complex processes. The organizational perspective focuses on highlighting social networks of resources by identifying the links between the resources and grouping these with common links.The key point of these approaches is to define appropriately what a link is. This link can be a metric [4], such as handover-of-work counting the successions between the resources in executions of the process, or a more complex approach [8] combining the successions between the resources (being applicable in the optimization of flows between staff members in an emergency room), the frequency of these successions and the distance between these successions. Such a network provides relevant information about the key resources and the interactions between them. Another focus of this perspective is trace clustering where resources, represented by a set of features (frequency of the activities [9], sub-processes [10], time, etc), are grouped into homogeneous clusters, that could be build iteratively [11], with the purpose of identifying different communities of resources. Trace clustering has many applications in the medical field, for example to highlight treatments by grouping patients performing the same process [9] or to optimize the organization of services by grouping together personnel acting in the same way [10].

The first perspective focuses on the activities whereas the second focuses on the resources, two distinct elements of process mining. However, clustering, whether focused on the activities or on the resources, is part of the main question related to both perspectives. The clustering is used to group people acting similarly or activities performed by the same people with the purpose to save resources (money, space, energy,...). Those clusterings allow to optimize the people related to the process or the process itself. A choice has to be made between the human optimization and the production optimization. Activities are similar if they are performed by similar resources and resources are similar if they perform similar activities. The purpose of this paper is to propose an approach combining the two perspectives by simultaneously clustering the activities and the resources. The main idea being that activities are similar if they are performed by the same resources and that resources performing the same activities are similar too. By combining both clustering, our approach allows a human-based optimization and a production-based optimization of the process simultaneously.

Let's assume an example considering a sales process representing the activities performed by sales assistants. There are two kinds of salespeople characterized by different sales techniques and, obviously, two final outputs: the sale is or is not completed. To be more efficient, the process owner would like to know, as soon as possible in the execution of the process, if the output will be positive and what the most effective sales technique is. Based on a half execution of the process, current approaches are able to differentiate processes with positive outputs from processes with negative outputs based on the sequence of activities performed. However, those approaches do not allow a direct conclusion of which kind of sales technique is linked to the outputs. Our proposal provides a bi-dimensional cluster linking the outputs of the process and the different kinds of salespeople. In addition to the results provided by current approaches or classical classifying methods, our proposal provides a better understanding of these results by linking them to their cause.

Our contribution, through this paper, is to:

1. Adapt the theoretical concepts of Cell Formation methods into a new methodology designed to link the case perspective with the resource perspective in process mining. The proposed methodology allows a new and improved understanding of the process more suited than other bi-dimensional clustering approaches based mainly on attributes and features.
2. Present how to implement the methodology and how to optimize it.
3. Validate the methodology using both synthetic and real-life datasets by comparing obtained results to a ground-truth.

The remainder of the paper is organized as follows. The next section explains the proposed approach and its key points through a running example. Section 3 presents the results of the approach on three datasets and analyzes the key parameters. Finally, last section concludes the paper by talking about future perspectives.

2. Hierarchical Cell Formation Clustering Approach

The proposed approach and the concepts used will be explained using a running example. The running example is described in Subsection 2.1. Subsection 2.2 describes our proposal based on a cell formation algorithm. This Subsection explains our approach and is divided into a first part on the normalization of the input data, a second part on the cell formation algorithm itself and a last part describing the hierarchical process of our method.

It is expected that our algorithm can create two-dimensional clusters efficiently and quickly. To meet these expectations, the algorithm have to:

- be robust to heterogeneous frequency distributions.
- be able to distinguish resources with similar but different behaviors.
- offer the possibility to provide different levels of granularity.

To fulfill all these criteria, we have based ourselves on recognized algorithms and methods that are widely used and have a strong theoretical foundation. As we will describe in more detail in this section:

- the DCA algorithm is efficient and fast.
- the hierarchical approach allows to control the granularity of the results.
- its combination with a normalization at each step allows to be both robust and sensitive to the frequency distribution.

The use of these confers a theoretical legitimacy to our algorithm which will be proven through the experiments presented in this section.

There are other approaches allowing bi-clustering with solid theoretical foundations based on other techniques and presenting high performances [12],[13],[14]. But in the context of process mining, with the expected specificities described above, the proposed approach is the only one offering the desired flexibility and quality.

2.1. Running Example

This section is based on a process representing a repair procedure, composed of eight activities, for which the logs are provided by ProM Tutorial [15]. Three kinds of resources (solver, tester and system) are involved in this process. It represents the repair of a defective product. After the registration of the product, it is analyzed. A repair sequence is then applied consisting of a choice between a simple or a complex repair. Whatever the kind of repair, the product is again tested, if this test fails, the repair has to be re-performed by going back to the analysis of the defect. In parallel of this repair procedure, the user is informed and both this information and the good completion of the repair has to be performed before archiving it. This process model the three main situations represented by Petri nets: the choice (between the kinds of repair), the concurrency (between the repair procedure and the information procedure) and the loop (if the repair procedure is restarted). The whole process is represented on Figure 1 using a Petri net. Through this example, we will show how our approach is able to identify the three different kinds of resources and the activities that make them different from each other.

2.2. Hierarchical Cell Formation Algorithm

This Subsection shows an overview of our proposal followed by a more detailed description of its parts as our approach is a combination of three elements: an adapted normalization, a cell formation algorithm and a hierarchical clustering approach. Figure 2 describes the global conduct of the algorithm.

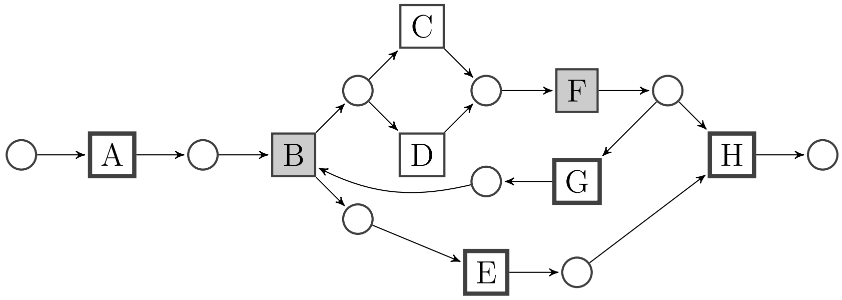

Figure 1: Running example process (A: register, B: analyze defect, C: repair simple, D: repair complex, E: inform user, F: test repair, G: restart repair, H: archive repair). The grey activities are performed by testers, the white activities are performed by solver and the activities in bold are performed by systems.

The input data are a set of activities performed by a set of resources extracted directly from the logs. The first step of our approach provides a two-dimensional matrix where the rows represent the resources and the columns the activities performed. The cells of this matrix are the frequencies of occurrence of a particular activity by a particular resource i.e. the cell (i,j) indicates how many times the resource i performs the activity j. The cell formation algorithms are designed to deal with binary matrices, therefore a normalization has to be used. The second step is then to normalize the frequencies by ensuring the best representation of the difference between them. The third step performs the cell formation itself. Thereupon, the clusters are extracted and the original frequencies are restored. The last step is to apply a top-down hierarchical procedure and go back the normalization step. This procedure consists of extracting clusters which are the results of the previous steps and reapplying all the steps to these clusters. This allows more accurate clusters to be obtained at each iteration.

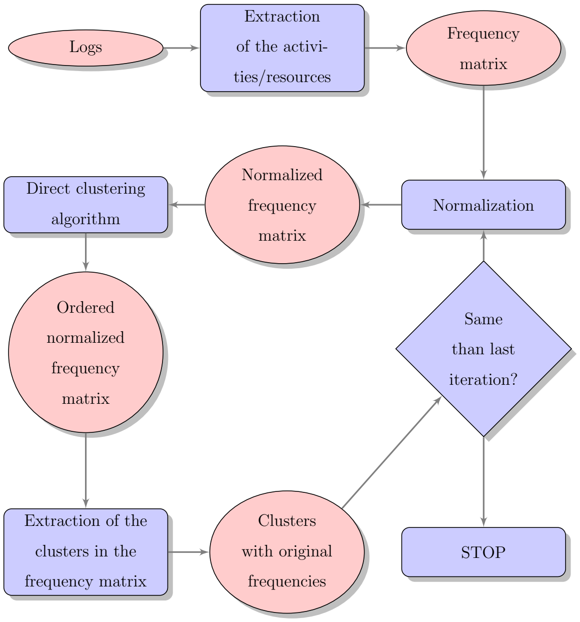
Logs
Extraction of the activities/resources
Frequency matrix
Normalization
Normalized frequency matrix
Direct clustering algorithm
Ordered normalized frequency matrix
Extraction of the clusters in the frequency matrix
Clusters with original frequencies
Same than last iteration?
STOP

Figure 2: Flow chart of the proposed approach

2.2.1. Normalization

Classical cell formation algorithms [16, 17] are designed to be used in manufacturing layout. Their original purpose is to identify which machines are involved in the production of which parts to regroup them and create efficient production units. They are only focused on the presence or not of the parts/machines in a particular production process. As such, they take as input a binary matrix where the columns represent the parts and the rows the machines. The value *1* is placed in the cell (i,j) if machine i is in the production process of part j, otherwise, the cell takes the value *0*. Here, as exposed on Table 1, a cell represents not only if a resource performs an activity but how many times the activity is performed. Accordingly, our approach has to propose a normalization allowing it to take into account the

Table 1: Frequency matrix of the running example

Resource	A	B	C	D	E	F	G	H
System 1	20	0	0	0	15	0	10	11
Tester 1	0	5	0	0	0	4	0	0
Solver 1	0	0	4	6	0	0	0	0
System 2	12	0	0	0	30	0	20	15
Tester 2	0	7	0	0	0	3	0	0
Solver 2	0	0	1	2	0	0	0	0

different frequencies.

The distribution of frequencies is highly biased by the large number of zeros representing the non-completion of activities, meaning that a normalization using the mean of all the frequencies is then irrelevant. The chosen normalization is based on a proportion of *1* to ensure the best trade-off between a fast convergence and a minimization of the loss of information. Our approach transforms a percentile of the values into 0 and all the others in 1 as depicted in Table 2. In this table, the eight highest values were transformed into 1 and the forty remaining into 0, this corresponds to the 5/6 th percentile. Subsection 3.3 explains how to determine the optimal value of this percentile.

2.2.2. Cell Formation

Because of the hierarchical top-down approach used, the algorithm is performed multiple times. Indeed, as detailed in the next Subsection, at

Table 2: Normalized frequency matrix of the running example

Resource	A	B	C	D	E	F	G	H
System 1	1	0	0	0	1	0	1	1
Tester 1	0	0	0	0	0	0	0	0
Solver 1	0	0	0	0	0	0	0	0
System 2	1	0	0	0	1	0	1	1
Tester 2	0	0	0	0	0	0	0	0
Solver 2	0	0	0	0	0	0	0	0

each iteration, the frequency matrices are divided into two and then the number of executions of the cell-formation algorithm is doubled. Therefore, a simple and time/memory not-consuming algorithm is more suitable than a complex and more accurate algorithm. The chosen algorithm is the *Direct Clustering Algorithm* (DCA) [16]. This algorithm is based on a succession of permutations of the rows and columns to create sub-matrices of 1.

The steps of the *DCA* are as follows:

1. Computing the rank (RR) of each row by summing all the positive entries of this row.
2. Rearranging the rows of the matrix (top to bottom) in descending order of the RR.
3. Computing the rank (RC) of each column by summing all the positive entries of this column.
4. Rearranging the columns of the matrix (left to right) in ascending order

Resource	A	E	G	H	B	C	D	F
System 1	1	1	1	1	0	0	0	0
System 2	1	1	1	1	0	0	0	0
Tester 1	0	0	0	0	0	0	0	0
Tester 2	0	0	0	0	0	0	0	0
Solver 1	0	0	0	0	0	0	0	0
Solver 2	0	0	0	0	0	0	0	0

Figure 3: Normalized frequency matrix of the running example

of RC.

5. Starting with the first column, transferring to the top the rows which have a positive entry for this column. Repeating this for the next column until all the rows are rearranged.
6. Starting with the first row, transferring to the left-most position the columns which have a positive entry for this row. Repeating this for the next row until all the columns are rearranged.
7. If there are no changes between step 5 and step 6: STOP otherwise go to step 4.

Figure 3 presents the results of the algorithm based on the normalized frequency matrix. It highlights a sub-matrix containing the activities A, E, G and H and the resources System 1 and System 2. It may be inferred that both of these resources mainly perform these four activities.

2.2.3. Hierarchical Approach

The last step is the hierarchical top-down procedure where the clusters created by the cell formation algorithm are extracted from the initial matrix and each of them goes back to the normalization step as described in Figure 4. This Figure represents the first iteration of the hierarchical approach.

The key point of this approach is to determine how to extract the submatrices from the outputs of the cell-formation algorithm. A cut is performed to divide the results of the cell formation in two. At each iteration of the algorithm, the cell-formation is then performed $2^{iteration-1}$ times.

This cut is again a trade-off between a fast convergence and a minimization of the loss of information which, will be discussed in Subsection 3.4.

2.3. Complexity

Each of the previously exposed step of the approach has its own complexity. The complexity of the normalization, regardless the hierarchical step, for an initial input matrix composed of R rows and C columns, is $\mathcal{O}$(C*R). The complexity of the direct clustering algorithm itself is $\mathcal{O}(RC^2+R^2C)$. The complexity of the hierarchical step is related to C and R. The algorithm can be performed at most $min(R, C)$, the global complexity of the algorithm is then $\mathcal{O}(R^2C^2)$.

The experiments were carried out on a processor Intel Core i7-7600U CPU 2.80GHz and the following results are the mean times after 20 runs of the algorithm for each dataset:

- Hospital Dataset: 112 seconds
- Review Dataset: 33 seconds

Resource	A	E	G	H	B	C	D	F
System 1	1	1	1	1	0	0	0	0
System 2	1	1	1	1	0	0	0	0
Tester 1	0	0	0	0	0	0	0	0
Tester 2	0	0	0	0	0	0	0	0
Solver 1	0	0	0	0	0	0	0	0
Solver 2	0	0	0	0	0	0	0	0

Resource	A	E	G	H
System 1	1	1	1	1
System 2	1	1	1	1

Resource	A	E	G	H
System 1	20	15	10	11
System 2	12	30	20	15

Resource	B	C	D	F
Tester 1	0	0	0	0
Tester 2	0	0	0	0
Solver 1	0	0	0	0
Solver 2	0	0	0	0

Resource	B	C	D	F
Tester 1	5	0	0	4
Tester 2	7	0	0	3
Solver 1	0	4	6	0
Solver 2	0	1	2	0

Figure 4: First iteration of the hierarchical ordered normalized frequency matrix of the running example

- Repair Dataset: 38 seconds

As expected, the execution times are are not related to the size of the datasets but to the number of resources/activities.

3. Experiments and Discussions

This section exposes the results of our approach and the setup of its key parameters. The next subsection describes the three datasets used. Subsection 3.2 presents how the results are evaluated. Subsections 3.3 and 3.4 discuss the impact of two key parameters and how to determine their optimal values. Finally, Subsection 3.5 presents a qualitative and quantitative analysis of the results.

3.1. Presentation of the datasets

To validate our approach, three datasets will be used. The first is a real-life medical dataset whereas the two last are synthetic datasets [15] which were not designed to be used in the organizational perspective.

The three datasets describe the following processes:

- Hospital Dataset: this dataset comes from the radiation oncology department of a Belgian hospital over two years. The dataset consists of 3,058 treatments containing 29,019 activities performed by 30 resources. There are three kinds of resources performing different activities: 15 nurses, 9 physicists and 6 doctors. The process of this radiation oncology department has clear procedures and well-defined activities. There are 11 activities describing the preparation of the radiotherapy itself. There is a simulation stage followed by a computation stage for the parameters and several activities concerning the coordination between the resources.

- Review Dataset [15]: this dataset comes from a scientific paper review process. The dataset consists of 100 submission containing 2297 activities performed by 11 resources. The resources can be separated into four classes of 2, 7, 1 and 1 people.

- Repair Dataset [15]: this dataset comes from the process of the running example. The dataset consists of 1,104 instances containing 11,855 activities performed by 13 resources. There are 3 main kinds of resources: 6 testers, 6 solvers and 1 system. The solvers can be divided into two different subgroups of 3 solvers performing close but different kinds of activities.

3.2. Metrics

The results of the algorithm have to be evaluated in terms of correctness and efficiency. The algorithm has to be efficient and converge as soon as possible to allow a clear representation of the results. On the other hand, the algorithm has to be able to distinguish close clusters of resources/activities and provide results as correct as possible.

Contrary to resources, the ground-truth on the activities is not available. As a consequence, only the precision and recall concerning the resources (*r-precision* and *r-recall*) could be computed. The *r-precision* related to a class of resource i is defined as:

$$r - precision(i) = \frac{n_{ii}}{n_{.i}}$$

where n_{ij} is the number of resources coming from the class i in cluster j. $n_{.i}$ represents all the resources in cluster i. The *r-recall* related to a class of resources i is defined as:

$$r - recall(i) = \frac{n_{ii}}{n_{i.}}$$

where $n_{i.}$ represents all the resources coming from class i. The global value for a particular clustering is obtained by averaging all the class values. Based on these two values, the F-measure is computed by:

$$F-Measure = \frac{2*r-precision*r-recall}{r-precision+r-recall}.$$

To be independent of the efficiency, the F-Measure is calculated for each step (F-Measure$_s$) and the best value is kept as final result. This value represents an indicator of the correctness of the resources regardless to the efficiency of the algorithm. The correctness of the clustering of the activities can only be evaluated qualitatively, the ground-truth not being available.

To evaluate the efficiency of the algorithm, the number of steps to reach the best correctness will be computed (n_{corr}) i.e. the number of steps at which the F-Measure is computed. This number allows us to measure how fast the algorithm reaches the best result from a correctness point of view. The global number of steps (n_{steps}) before the completion of the algorithm is also computed.

3.3. Normalization Parameter

The normalization of occurrences consists of replacing a percentile (perc$_{norm}$) of the values by 0 and the others by 1. To determine the best value of perc$_{norm}$, the algorithm was performed multiple times on all the datasets with different values of the parameter while keeping the other parameters unchanged.

Figure 5 shows the evolution of the global F-Measure and of the F-Measure$_1$ at the first step regarding to the perc$_{norm}$ for the hospital dataset. We see that under a percentile, the values are constant due to the fact that there are a lot of zeros in the matrix. If perc$_{norm}$ is lower than the percentage of zero cells of the matrix, all the positive cells are set to 1 and perc$_{norm}$ is no longer relevant.

A high value of perc$_{norm}$ induces a decrease of the correctness of the results. This can be explained by the fact that two completely different resources or activities cannot be differentiated because the normalization is too strong and almost all the frequencies are set to 0.

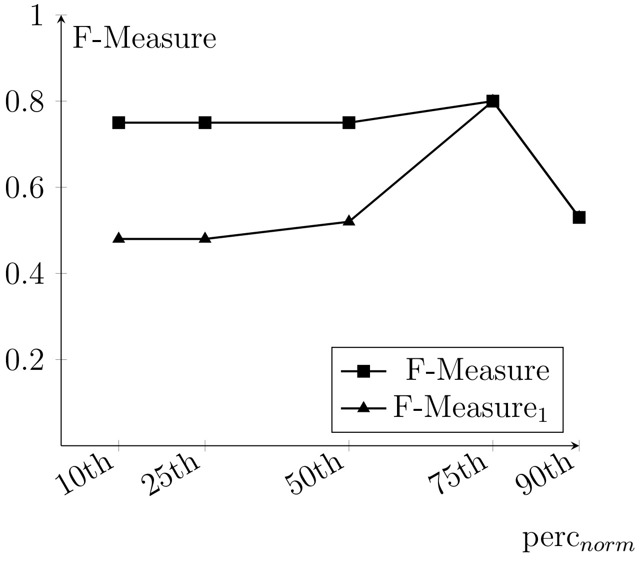

Figure 5: F-Measure and F-Measure$_1$ for the hospital dataset

Figure 6 presents the number of steps performed before reaching the best F-Measure and the total number of performed steps for the hospital. n_{corr} and n_{steps} decrease with the increase of perc$_{norm}$, indeed, a higher perc$_{norm}$ reduces the differentiation between the frequencies which are, in most cases, equal to zero. Once more, under a percentile, the results are equal.

The results for both the review and repair datasets are not relevant be-

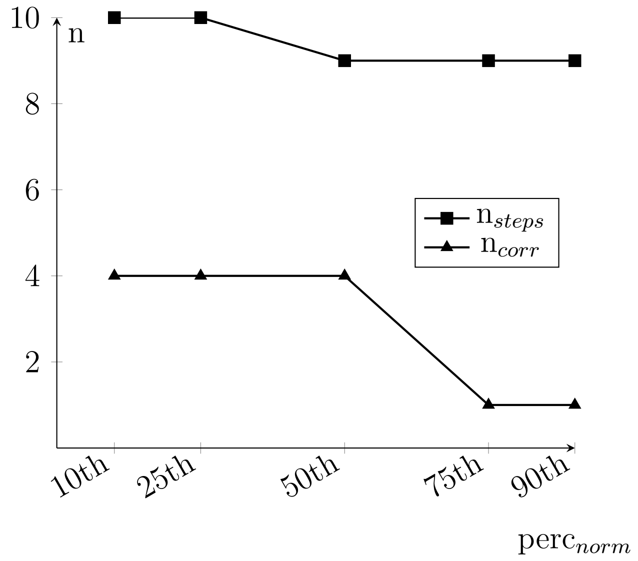

Figure 6: n$_{steps}$ and n$_{corr}$ for the hospital dataset

cause the algorithm provides a perfect F-Measure at the same step regardless of the perc$_{norm}$ in a range between the 10th and the 90th percentile. Theses datasets are not complex enough to provide relevant information on the perc$_{norm}$. The conclusion of the previous results is that the chosen perc$_{norm}$ is the 75th percentile which is a high enough value to avoid all the positive values being set to 1 and low enough to avoid that completely different resources were represented by the same values. This result is dependent on the datasets used, therefore one of the purposes of this section is to identify the way forward to compute perc$_{norm}$. A more robust way to determine the perc$_{norm}$ is to consider the 50th percentile of the non-zero value. This allows us to have results independent of the sparseness of the input matrix.

3.4. Separation Parameter

The separation of a matrix into two submatrices is not, in most cases, as obvious as described on Figure 3. Figure 7 exposes a more complex but realistic example where the separation of the initial matrix is harder to achieve than the case in Figure 3.

Resource	A	E	G	H	B	C
Resource 1	1	1	1	1	0	0
Resource 2	1	1	1	1	0	0
Resource 3	0	0	1	1	1	0
Resource 4	0	0	1	1	1	0
Resource 5	0	0	0	0	1	1
Resource 6	0	0	0	0	1	1

Figure 7: Normalized frequency matrix of a complex example

Our algorithm is based on a separation using a threshold corresponding to the number of different cells. For each row and for each column, from the top-left corner to the right-bottom corner of the matrix, the number of different cells is computed, and when the threshold is reached, the separation is performed. A threshold of three for the rows and columns in the example of Figure 7 will extract the resources 1 and 2 and the activities A, E , G and H. This threshold can also be expressed in terms of percentage applied to the global number of rows/columns rather than directly as a number of rows/columns. A threshold on both rows and columns of 50% applied to the example of Figure 7 will provide the same results as previously. This formulation allows the algorithm to be scalable.

Based on our datasets, we have to determine the best value of this threshold. Multiple values of this perc$_{sep}$ on the three datasets with perc$_{norm}$ equal to the 75th percentile were applied. As described in Figure 8, extreme values provide bad results. When the value of perc$_{sep}$ is too low, this does not allow the algorithm to differentiate close rows/columns, only perfectly similar rows/columns are clustered together. When the value of perc$_{sep}$ is too high, this leads to a situation where all the rows/columns look similar and this makes the algorithm inefficient.

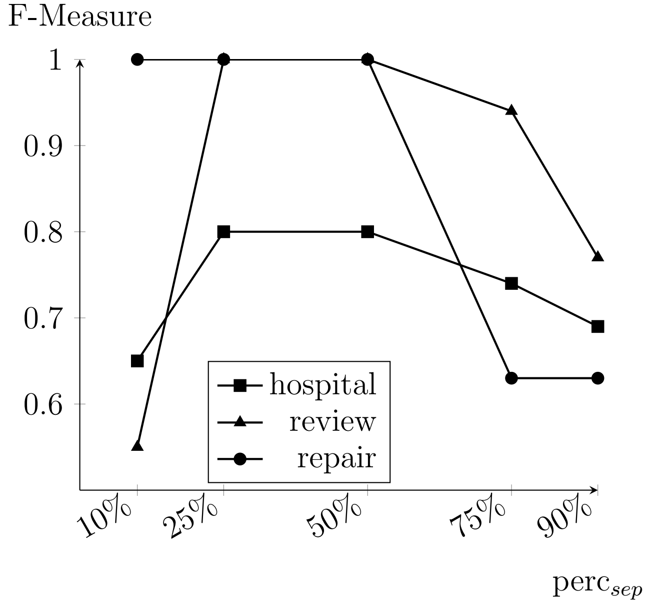

Figure 8: F-Measure for the three datasets

Perc$_{sep}$ also has a logical direct impact on n$_{steps}$, as shown in Figure 9. The decrease of n$_{steps}$ follows the increase of perc$_{norm}$ because the more the algorithm detect rows/columns which seem different the higher the number of iterations to complete the algorithm is.

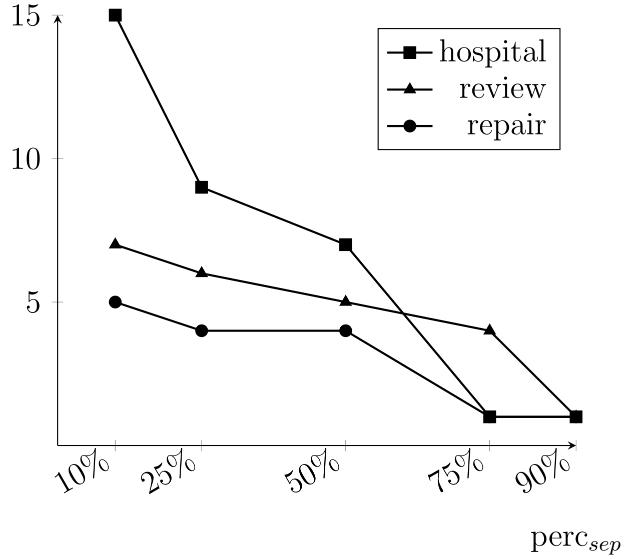

Figure 9: n$_{steps}$ for the three datasets

The chosen value for perc$_{sep}$ is 50% which is the best trade-off between the efficiency of the algorithm and its correctness.

3.5. Results

Considering perc$_{norm}$ = 50th percentile when the non-zero values and perc$_{sep}$ is 50%, this subsection presents the results of the clustering for the three datasets and analyzes them qualitatively and quantitatively.

3.5.1. Repair Dataset

Figure 10 shows that the final result of our approach is similar to the theoretical result extracted from the running example. The F-Measure is equal to 1 after 4 iterations, the first iteration distinguishes the system and the humans workers. The second iteration is able to separate the solvers from the testers. The third iteration identifies, for the solvers, the two different kinds among the humans workers. The fourth iteration being equal to the previous one for the four clusters, the algorithm stops.

Resource	A	E	G	H	B	F	C	D
System 1	X	X	X	X				
Tester 1					X			
Tester 2					X			
Tester 3					X			
Tester 4						X		
Tester 5						X		
Tester 6						X		
Solver 1							X	
Solver 2							X	
Solver 3							X	
Solver 4								X
Solver 5								X
Solver 6								X

Figure 10: Final result of the algorithm on the Repair Dataset

Referring to the caption of Figure 1, the validation of the results concerning the clustering of the activities can be carried out. We can extrapolate that an F-Measure computed using the activities and not the resources will also be equal to 1. Although this dataset is relatively simple, our approach correctly clusters both resources and activities with high efficiency.

3.5.2. Review Dataset

The F-Measure of the results of this dataset, is also equal to 1 which means that, from the resource point of view, the ground-truth is correctly identified. There are four clusters and, from the activities point of view, they regroup:

- for the 1st cluster: get reviews 1, get reviews 2 , get reviews 3 and get reviews X.
- for the 2nd cluster: time-out 1, time-out 2, time-out 3 and time-out X.
- for the 3nd cluster: invite reviewers, invite additional reviewers, collect, accept and reject.
- for the 4nd cluster: decide

Although the ground-truth is not available for the activities, this list indicates that the clusters are consistent. This dataset highlights the improvement brought by our proposal. The 11 resources are identified only by their names (Mike, Ann, Pete, etc). The classical approaches described in the introduction are able to identify the four clusters of resources and the four clusters of activities, but our method is able to conclude that Wil decides or that Mike and Anne invite the reviewers, collect, accept and reject. It was possible to identify that Mike and Ann work on the same activities but now, among all the activities they perform, it is possible to extract who in particular and to identify which activities distinguish them from other resources.

3.5.3. Hospital Dataset

This dataset is the most complex because, even though the resources are correctly identified, several activities are performed by resources coming from different kinds of resources. For example, if an activity A is performed equally by nurses and doctors, even if the algorithm is able to correctly identify the cluster of the doctors and the cluster of the nurses, activity A will be attributed to one of them and will decrease the quality of the results. Another possibility is that the created cluster regroups all the resources performing activity A, this situation leads to a decrease in the F-Measure and explains why it is equal to 0.8.

This F-Measure is slightly lower than the results of the classical approaches considering the identification of resource clusters but, again, our approach proposes additional information by summing clustering as well as the relevant activities linked to these resources.

3.6. Discussion

The previous paragraphs highlight the key points of our method and attest is efficiency from a quantitative point of view on multiple complex datasets.

Based on these paragraphs, we can conclude that our method :

- allows to cluster resources and activities simultaneously providing results close to the ground-truth.

- is robust to outliers (infrequent performed activities) being based on multiple normalization of the inputs.
- provides an interpretation of the results improved by using different levels of granularity allowed by the hierarchical approach.

In addition to this, one of our main purpose is to improve the effectiveness of the information obtained from a qualitative point of view. Even if the know-how of an expert domain is required for each different datasets, several trivial improvements could be observed as for the review dataset where our approach identifies clearly and simultaneously the different roles and tasks of the process.

Table 3: Comparison between several clusterings for the hospital dataset

Clustering	F-Measure
DCA hierarchical approach	0.8
Euclidean distance	0.52
Manhattan distance	0.5
Pearson distance	0.53
Chebyshev distance	0.4

Table 3 is comparing our approach to four well-known distances used for the clustering with a classical algorithm (K-Means), using the F-Measure. This comparison is based on the ground-truth concerning the activities, the only available, of the hospital dataset. Moreover to improve the effectiveness of the information obtained by a two dimensional clustering, the quality of the clustering in a single way is also clearly improved.

4. Conclusion

The purpose of this paper was to be able to connect the organizational perspective and the case perspective through the clustering issue by developing an approach simultaneously clustering the resources and the activities.

Based on a realistic example, the key points of the developed algorithm were introduced: the extraction of the frequencies from the logs, the normalization of the data, the cell formation algorithm and the top-down hierarchical procedure. Each of these concepts being commonly frequently used and with multiple variants, a major contribution of this paper was to combine them, to choose and motivate the most suitable implementation and to effectively set them up.

The results show that from a single point of view (the resources or the activities) our approach is equal or relatively close to the existing approaches but, combining both points of view provides a significantly higher explanatory power.

Combining two perspectives enhances the analysis of the processes, therefore, future research taking into account additional perspectives in the clustering will be envisaged.

Applying our proposal to three datasets coming from different fields, with different features and with different levels of complexity confirms than our approach highly improves the understanding of the considered processes.

References

[1] W. Van Der Aalst, Process mining: discovery, conformance and enhancement of business processes, volume 2, Springer, 2011.

[2] W. Van der Aalst, T. Weijters, L. Maruster, Workflow mining: Discovering process models from event logs, IEEE Transactions on Knowledge & Data Engineering (2004) 1128–1142.

[3] M. Song, W. Van der Aalst, Towards comprehensive support for organizational mining, Decision Support Systems 46 (2008) 300–317.

[4] W. M. Van Der Aalst, H. A. Reijers, M. Song, Discovering social networks from event logs, Computer Supported Cooperative Work (CSCW) 14 (2005) 549–593.

[5] N. Tax, N. Sidorova, R. Haakma, W. M. van der Aalst, Mining local process models, Journal of Innovation in Digital Ecosystems 3 (2016) 183–196.

[6] M. de Leoni, S. Dündar, Event-log abstraction using batch session identification and clustering, in: Proceedings of the 35th Annual ACM Symposium on Applied Computing, SAC '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 36–44. URL: https://doi.org/10.1145/3341105.3373861. doi:10.1145/3341105.3373861.

[7] C. Günther, W. Aalst, Mining activity clusters from low-level event logs, Cirp Annals-manufacturing Technology - CIRP ANN-MANUF TECHNOL (2006).

[8] C. Alvarez, E. Rojas, M. Arias, J. Munoz-Gama, M. Sepúlveda, V. Herskovic, D. Capurro, Discovering role interaction models in the emergency room using process mining, Journal of biomedical informatics 78 (2018) 60–77.

[9] M. Song, C. W. Günther, W. M. Van der Aalst, Trace clustering in process mining, in: International Conference on Business Process Management, Springer, 2008, pp. 109–120.

[10] L. Delcoucq, F. Lecron, P. Fortemps, W. M. P. van der Aalst, Resource-centric process mining: Clustering using local process models, in: Proceedings of the 35th Annual ACM Symposium on Applied Computing, SAC '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 45–52. URL: https://doi.org/10.1145/3341105.3373864. doi:10.1145/3341105.3373864.

[11] A. K. A. De Medeiros, A. Guzzo, G. Greco, W. Van Der Aalst, A. Weijters, B. F. Van Dongen, D. Saccà, Process mining based on clustering: A quest for precision, in: International Conference on Business Process Management, Springer, 2007, pp. 17–29.

[12] W. Weng, W. Zhou, J. Chen, H. Peng, H. Cai, Enhancing multi-view clustering through common subspace integration by considering both global similarities and local structures, Neurocomputing 378 (2020) 375–386. URL: https://www.sciencedirect.com/science/article/pii/S0925231219313888. doi:https://doi.org/10.1016/j.neucom.2019.10.014.

[13] H. Wang, G. Han, B. Zhang, G. Tao, H. Cai, Multi-view learning a decomposable affinity matrix via tensor self-representation on grassmann manifold, IEEE Transactions on Image Processing 30 (2021) 8396–8409. doi:10.1109/TIP.2021.3114995.

[14] H. Peng, Y. Hu, J. Chen, W. Haiyan, Y. Li, H. Cai, Integrating tensor similarity to enhance clustering performance, IEEE Transactions on Pattern Analysis and Machine Intelligence (2020) 1–1. doi:10.1109/TPAMI.2020.3040306.

[15] H. E. Verbeek, R. J. C. Bose, Prom 6 tutorial, Technical report, Tech. Rep. (2010).

[16] H. M. Chan, D. Milner, Direct clustering algorithm for group formation in cellular manufacture, Journal of Manufacturing systems 1 (1982) 65–75.

[17] J. R. KING, Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm, International Journal of Production Research 18 (1980) 213–232. URL: https://doi.org/10.1080/00207548008919662. doi:10.1080/00207548008919662. arXiv:https://doi.org/10.1080/00207548008919662.