Similarity Analysis of Time Interval Data Sets Regarding Time Shifts and Rescaling

Marc Haßler, Sabina Jeschke, and Tobias Meisen

Institute of Information Management in Mechanical Engineering,
RWTH Aachen University, Germany
marc.hassler@ima.rwth-aachen.de,
home page: https://www.ima-zlw-ifu.rwth-aachen.de

Abstract. Comparing things like objects, tasks, texts or audio is a common task in computer science. To do so, first a definition for similarity is required. In many fields of application, common and generic distance measures like the Minkowski distance or more specific measures like Dynamic Time Warping to compare temporal sequences are already defined and used. Based on our state of knowledge, there is no applicable measurement for calculating the similarity between time interval data sets in a manlike understanding.
In this paper, we present a novel method to compare time interval data sets while using an adapted distance measurement. With our approach we look at the data sets as the disjoint parts of a bigraph, such that we can use methods from graph theory. In particular, our solution provides the opportunity to take dynamic changes (like rescaling or time-shifting) into account and thus allows the comparison of real data in humanoid fashion. Hence, it allows to compare real data with e.g. scale models.

Keywords: time interval data set, TIDA, similarity analysis, graph theory, temporal displacement

1 Introduction and Motivation

Nowadays, process optimization is an essential feature in many areas of manufacturing [1]. The stability of these optimized processes is particularly important because continuous deviance could lead to aberration within the process management regarding its optimized parameters. This may result in unwanted time delays and additional costs as, due to the increasingly frequent on-demand production, there are no products in storage [2]. Today's optimized and timed procedures are more susceptible to irregularity, as a result of which, in addition to the optimization, the deviations themselves become more and more superficial. Since these deviations cannot always be avoided, a strategy for faster responses must be available.

At the moment, workers recognize deviations based on their experience and start appropriate counter measures. This procedure resembles a similarity analysis regarding past processes. Recognitions like these are not often part of the

computer-aided similarity analysis because at the moment there are only few limited possibilities to quantify interval similarities. Examples for specific similarity analysis already exist within certain scientific research areas. Those methods include the area of text or image processing, where similarity analysis is used to optimize search algorithms [3], within biology to compare genes or gene groups ([4] and [5]) or as a tool of audio recognition methods [6]. To our knowledge, basic considerations of similarities regarding time intervals are missing up to now. However, research regarding time interval data sets gained importance over the past years ([7] - [12]). While the similarity in the mentioned publications was derived from a sequence analysis and studies the existing data sets as a whole, we deduce the similarity of the data set from the individual similarities between the underlying intervals in our approach.

Therefore, we concentrate solely on the time intervals and at first construct a similarity measure to compare intervals with each other. This method allows for a detailed view on specific characteristics of the records saved in the time interval data set and a comparison even under big time offsets is possible. With this approach we are able to measure the similarity of two data sets with well known methods from graph theory [13]. To achieve our goal, we interpret the comparative data sets as the disjoint parts of an bigraph where each time interval is represented by a node within these parts and the weight of each edge represents the similarity measure of the corresponding intervals.

2 Related Work

One work regarding time interval data sets [7] compares two data sets in relation to the correlation of the intervals within the respective data set. Within this method, a difference regarding the interval length is not considered as long as it does not effect the correlation between the two intervals. The authors introduce seven interval correlations, which they used for their comparison (cf. figure 1).

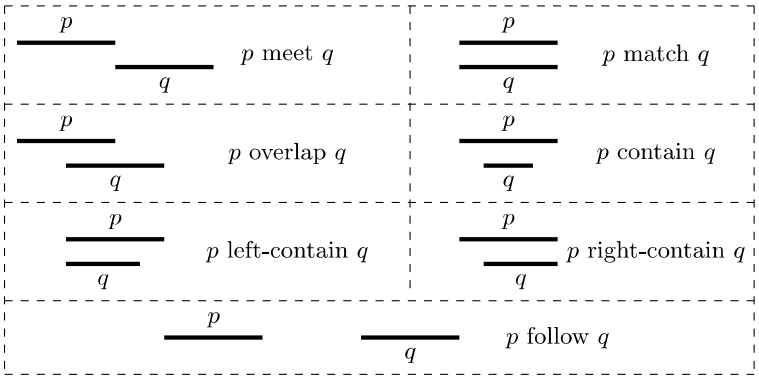

Fig. 1. Interval relation within a data set defined by Kostakis et al. [7]

Further authors [8] differentiate the similarity analysis into three distances, which are later combined to form the similarity measure. These distances are determined at a specific time t and are the following:

1) 'temporal order distance' compares the number of active intervals at time t.
2) 'temporal measure distance' matches the 'value' of all intervals at time t.
3) 'temporal relation distance' analyzes the relation of all intervals at time t.

This approach takes into account the lengths of the individual intervals, but only considers the data set for each evaluation at a certain point in time. Therefore, even small time shifts in one of the datasets are fully changing the outcome of the analysis.

The previously mentioned methods can be described as static comparisons, as depicted in Figure 2, yet global changes (like temporal displacements) are not regarded. Here, our method has a decisive advantage as we are able to allow global changes to be incorporated into the model by matching individual intervals.

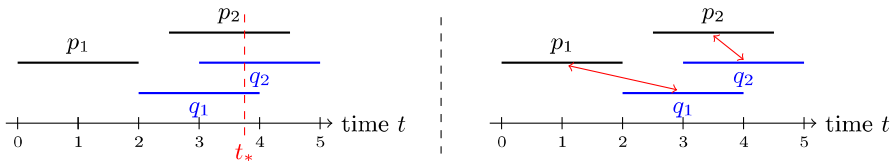

Fig. 2. *left:* time-based view on data sets by Meisen [8]; *right:* our interval approach

3 Similarities Between Time Intervals

In order to make sure that two time intervals are comparable, we take a closer look at the construction of these intervals. They consist of a start point and an end point and any amount of metadata, such as device class or hourly cost to run e.g. a specific process. In this paper, we assume that the metadata is available in mathematical form and is thus comparable (cf. chapter 3.2). We consider the following form for an interval p:

$$p := (s_p, e_p, M_{p_i} \mid i \in \mathbb{N})$$
$$\text{or in } \textit{short form } p := (s_p, e_p)$$

where

$$s_p := \text{start point of the interval}$$
$$e_p := \text{end point of the interval}$$
$$M_{p_i} := \text{i-th metadata of the interval}$$

The analysis is divided into three parts. At first, the geometrical data of each interval, such as length or position on the time axis, is compared to generate geometrical distances between two intervals. In the second part, the metadata as well as the possibility to address deadlines or earliest starting time is added into the interval similarity. In the end, all of these information define a similarity measure for two time intervals.

3.1 Geometrical Analysis

In the first step, we use the information for each interval to generate several distances with the possibility to evaluate each characteristic differently. For two intervals $p = (s_p, e_p)$ and $q = (s_q, e_q)$ as well as a norm $||\cdot||$, we conclude the following geometrical attributes.

1) Start point distance:

$$D_S(p,q) := \frac{||s_p - s_q||}{||\max\{e_p, e_q\} - \min\{s_p, s_q\}||} \tag{1}$$

2) End point distance:

$$D_E(p,q) := \frac{||e_p - e_q||}{||\max\{e_p, e_q\} - \min\{s_p, s_q\}||} \tag{2}$$

3) Lengths distance:

$$D_L(p,q) := 1 - \frac{\min\{||e_p - s_p||, ||e_q - s_q||\}}{\max\{||e_p - s_p||, ||e_q - s_q||\}} \tag{3}$$

4) Overlap:

$$D_O(p,q) := 1 - \frac{||p \cap q||}{\min\{||e_p - s_p||, ||e_q - s_q||\}} \tag{4}$$

with the interval

$$p \cap q = \begin{cases} (\max\{s_p, s_q\}, \min\{e_p, e_q\}) & \text{for } \max\{s_p, s_q\} < \min\{e_p, e_q\} \\ 0 & \text{else} \end{cases}$$

5) Gap:

$$D_G(p,q) := \begin{cases} \frac{\min\{||s_q - e_p||, ||s_p - e_q||\}}{||\max\{e_p, e_q\} - \min\{s_p, s_q\}||} & \text{for } ||p \cap q|| = 0 \\ 0 & \text{else} \end{cases} \tag{5}$$

Example 1. To visualize the geometrical attributes, we take a closer look at the intervals $p := (0, 10)$ and $q := (3, 7)$ and calculate their attributes:

1) $||p|| = 10$, $||q|| = 4$ and $||\max\{e_p, e_q\} - \min\{s_p, s_q\}|| = 10$
2) $p \cap q = (3, 7)$ and therefore $||p \cap q|| = 4$
3) $||s_p - s_q|| = 3$, $||e_p - e_q|| = 3$ and $D_G(p,q) = 0$

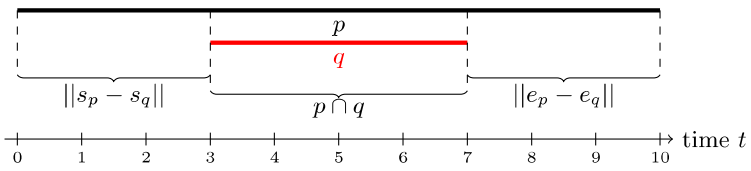

Fig. 3. Visualization of two intervals with their geometrical attributes

3.2 Metadata and Dealing With Deadlines

Like stated in the beginning, it is assumed that the metadata related to the considered time intervals p and q are in mathematically comparable form. That means that for every metadata i there is a continuous distance D_{M_i} with $0 < D_{M_i}(p,q) < 1$ available. The metadata is used to identify, if two intervals are comparable or not. If e.g. machine classes are considered, it measures whether the machines used within the intervals have equivalent functions and are therefore comparable.

A termination criterion regarding interval deadlines is also added, which means that, if interval p is compared with q, we want to make sure that interval q does not end after p has ended. The same goes for a start condition. Therefore, we defined the following two distances.

$$D_{END}(p,q) := \begin{cases} \min\{1, ||e_q - e_p||\} & \text{for } e_q > e_p \\ 0 & \text{else} \end{cases} \tag{6}$$

$$D_{START}(p,q) := \begin{cases} \min\{1, ||s_p - s_q||\} & \text{for } s_p > s_q \\ 0 & \text{else} \end{cases} \tag{7}$$

3.3 Similarity of Two Time Intervals

With the introduced distances, a distance measure for two time intervals is defined, where every characteristic is individually weighted. This measure is then used in chapter 4 two calculate the similarity between two data sets.

Definition 1 (distance between time intervals). *For two intervals p and q, the distance between them is measured by calculating the weighted sum of distances:*

$$S(p,q) := \sum_{i \in I} \lambda_i \cdot D_i(p,q) \tag{8}$$

Thus, the more similar the two intervals p and q are to each other, the smaller the value of $S(p,q)$ is. In the next step, two time interval data sets are compared and the similarity using this approach is evaluated.

4 Similarity Analysis Regarding Time Interval Data Sets

In this section, two interval data sets P and Q are compared. At first, the same cardinality for both P and Q is assumed, that means the number of intervals in each data set is the same. In Chapter 4.2, a procedure for dealing with different cardinalities is introduced. Furthermore, the intervals in P are specified with p_i and q_i for Q. For the remainder of the paper, data sets are considered as disjoint partial sets of a complete, weighted bipartite graph (cf. figure 4), in which the edge weight between two nodes corresponds to the interval similarity measure S.

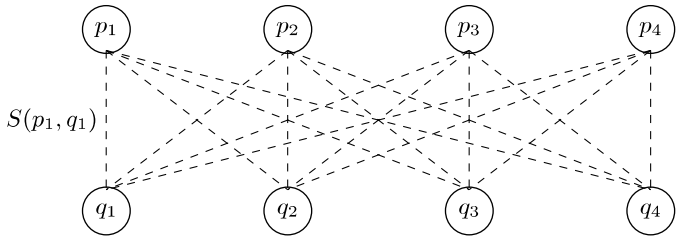

Fig. 4. Representation as a bipartite graph

Hence, the similarity of time interval data sets (STIDes) is equivalent to a perfect matching with minimal weight within our constructed bipartite graph.

Definition 2 (STIDes approach). *Let P and Q be two time interval data sets, $p_i \in P$, $q_i \in Q$ and $|P| = |Q| = n$. Furthermore, Π is the set of permutations of a set with n elements and $\pi \in \Pi$. The similarity between P and Q is determined by the following distance measure*

$$S(P,Q) := \min_{\pi} \left\{ \sum_{i=1}^{n} S(p_i, q_{\pi(i)}) \right\}_{\pi \in \Pi} \tag{9}$$

Such minimization problems in bipartite graphs can be solved within polynomial time by using for example the Hungarian algorithm [13]. Our approach is therefore capable of calculating a similarity measure within polynomial time while being able to prioritize certain characteristics and measure similarities even with existing time shift. In the next part, we expand this static approach for a dynamic similarity search, which also includes rescaling and shifting possibilities.

4.1 Dynamic Changes Within One Data Set

Until now, the previous static approach has difficulties in determining realistic similarities as soon as one of the time interval sets has big temporal shifts. In Figure 5 we recognize, that the pattern of p_1 and p_2 is the same as the pattern of

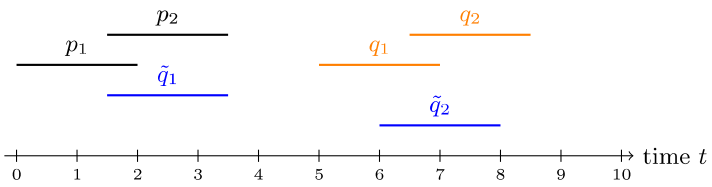

Fig. 5. Similarity regarding big temporal shifts

q_1 and q_2. Our static approach would determine $\tilde{Q} = \{\tilde{q}_1, \tilde{q}_2\}$ as more similar to $P = \{p_1, p_2\}$. A comparison with true-to-scale model data sets is not provided in the basic configuration either, e.g. in Figure 6 the Set $Q = \{q_1, q_2\}$ is exactly like $P = \{p_1, p_2\}$, only compressed by factor $\frac{1}{2}$. The static algorithm would choose $\tilde{Q} = \{\tilde{q}_1, \tilde{q}_2\}$ like before. However the construction of the interval distances allows an extension of the desired properties. Therefore, we define two kinds of operations.

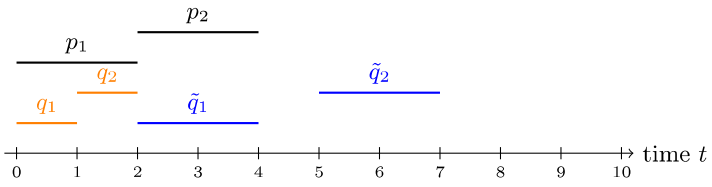

Fig. 6. Similarity regarding true-to-scale model data sets

Definition 3. *Let $p = (s_p, e_p)$ be an time interval in short form. Furthermore, let $v \in \mathbb{R}$ be a shift parameter and $s \in \mathbb{R}_+$ a scaling factor. The functions*

$$p + v := (s_p + v, e_p + v) \tag{10}$$
$$s \cdot p := (s \cdot s_p, s \cdot e_p) \tag{11}$$

map an interval onto a new interval, hence we can integrate these functions into our similarity measure.

For the similarity analysis of our data sets, this means that we have to solve the following minimization problems:

Definition 4. *Let P and Q be two time interval data sets, $p_i \in P$, $q_i \in Q$ and $|P| = |Q| = n$. Furthermore, let Π be the set of permutations of a set with n elements, $\pi \in \Pi$, $v \in \mathbb{R}$ a shift parameter and $s \in \mathbb{R}_+$ a scaling factor. The degree of similarity taking into account global displacement (12) or global scaling*

(13) can then be calculated with

$$S(P, Q + v) := \min_{\pi, v} \left\{ \sum_{i=1}^{n} S(p_i, q_{\pi(i)} + v) \right\}_{\pi \in \Pi, v \in \mathbb{R}} \tag{12}$$

$$S(P, s \cdot Q) := \min_{\pi, s} \left\{ \sum_{i=1}^{n} S(p_i, s \cdot q_{\pi(i)}) \right\}_{\pi \in \Pi, s \in \mathbb{R}_+} \tag{13}$$

Before dealing with an efficient solver of the above minimization problems, the solubility must be ensured. Therefore, the following is stated.

Lemma 1 (Existence of the Minimum).
Let the conditions of definition 4 be satisfied. The following functions are then continuous with a global minimum.

$$F_1(v) := \min_{\pi} \left\{ \sum_{i=1}^{n} S(p_i, q_{\pi(i)} + v) \right\}_{\pi \in \Pi} \tag{14}$$

$$F_2(s) := \min_{\pi} \left\{ \sum_{i=1}^{n} S(p_i, s \cdot q_{\pi(i)}) \right\}_{\pi \in \Pi} \tag{15}$$

Proof. For $\pi_k \in \Pi$ we define

$$f^1_{\pi_k}(v) := \sum_{i}^{n} S(p_i, q_{\pi_k(i)} + v) \tag{16}$$

Analogous we define $f^2_{\pi_k}(s)$.

Continuity:
We concentrate on the functions f^1_*, f^2_* are valid analogously.

1) Let $\Pi = \{\pi_1\}$. $F_1(v) = f^1_{\pi_1}(v)$ is then continuous because it is a sum of continuous distance measures $D_*(p_i, q_{\pi_1(i)} + v)$.
2) Let $\Pi = \{\pi_1, \pi_2\}$.

$$F_1(v) = \min\left\{f^1_{\pi_1}(v), f^1_{\pi_2}(v)\right\} = \frac{f^1_{\pi_1}(v) + f^1_{\pi_2}(v) - |f^1_{\pi_1}(v) - f^1_{\pi_2|}(v)}{2} \tag{17}$$

   is then continuous as a combination of continuous functions.
3) Let $F_1(v)$ be continuous for $|\Pi| = n$, then $|\Pi| = n + 1$ holds:

$$F_1(v) = \min\left\{f^1_{\pi_1}(v), \ldots, f^1_{\pi_{n+1}}(v)\right\} \tag{18}$$

$$= \min\left\{f^1_{\pi_1}(v), \ldots, f^1_{\pi_{n-1}}(v), \min\left\{f^1_{\pi_n}(v), f^1_{\pi_{n+1}}(v)\right\}\right\} \tag{19}$$

   and therefore $F_1(v)$ is continuous for $|\Pi| = n + 1$

That means $F_1(v)$ and $F_2(s)$ are continuous functions.

Existence of a minimum:
To show the existence of a minimum, the well known extreme value theorem of Weierstrass[1] is used. The continuity of the functions $F_1(v)$ and $F_2(v)$ was already shown. The last step is to show that there exists an interval $[v_u, v_o]$ (or $[s_u, s_o]$) for which the values of the function $F_1(v)$ (or $F_2(s)$) outside the interval are greater than at least one within. These intervals for both functions are now constructed.
For $F_1(v)$ we define

$$v_u = -|| \max_i \left(e_{q_i} \mid q_i \in \ Q\right) - \min_j \left(s_{p_j} \mid p_j \in P\right) || \tag{20}$$

as well as

$$v_o = || \max_i \left(e_{p_i} \mid p_i \in \ P\right) - \min_j \left(s_{q_j} \mid q_j \in Q\right) || \tag{21}$$

Because of the construction of the geometrical distances, that means for every $v > v_o$ (or $v < v_u$):

$$F_1(v) \geq F_1(v_o) \text{ (or } F_1(v) \geq F_1(v_u) \text{)} \tag{22}$$

For $F_2(s)$ we define

$$s_u = \min \left\{ \frac{\min \{||p_i||\}}{\max \{||q_i||\}}, \frac{\min_i \left(s_{p_i} \mid p_i \in P\right)}{\max_j \left(e_{q_j} \mid q_j \in \ Q\right)} \right\} \tag{23}$$

as well as

$$s_o = \max \left\{ \frac{\max \{||p_i||\}}{\min \{||q_i||\}}, \frac{max_i \left(e_{p_i} \mid p_i \in P\right)}{min_j \left(s_{q_j} \mid q_j \in \ Q\right)} \right\} \tag{24}$$

And analogously $F_1(v) \geq F_1(v_o)$ (or $F_1(v) \geq F_1(v_u)$) holds for $s > s_o$ (or $s < s_u$). That a minimum for $F_1(v)$ (or $F_2(s)$) exists and is located within the interval $[v_u, v_o]$ (or $[s_u, s_o]$) is then shown by the extreme value theorem. □

4.2 How to Deal With Different Cardinality

If the two disjoint parts of the bipartite graph do not have the same cardinality, the smaller of the two sub-sets is filled with additional nodes, dubbed "dummy nodes". Here, the edge weight of all nodes of the larger subset with the dummy node is set to the maximum occurring edge weight. With the help of this construction, we are able to use the Hungarian algorithm to find the perfect matching within our data sets. In this matching, all intervals which are

[1] A continuous function on an interval $[a, b]$ is bounded on that interval

connected to a dummy node, are not included in the perfect matching. In order to be able to use this result completely in the similarity analysis of time interval data sets, the distance measure must be adapted since the maximum distance measure of all interval pairs has been incorporated into the similarity measure by each dummy node. For convenience, these additional summaries are initially removed from the similarity measure.

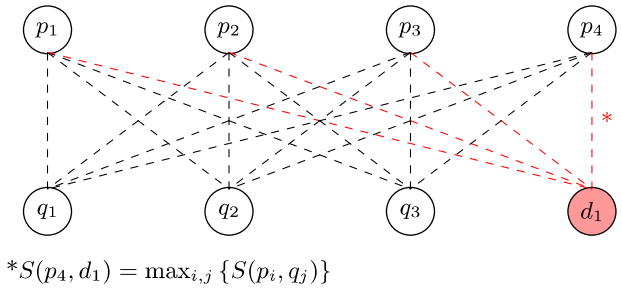

Fig. 7. Bigraph example with one dummy node d_1

In figure 7 we show an example with one dummy node. The adjusted calculation with the STIDes approach then is

$$S(P, \{Q \cup \{d_1\}\}) := \min_{\pi} \left\{ \sum_{i=1}^{n} S(p_i, q_{\pi(i)}) \right\}_{\pi \in \Pi} - |\{d_1\}| \cdot \max_{i,j} \{S(p_i, q_j)\} \quad (25)$$

The extent to which unmatched intervals influence the similarity measure must be considered according to the individual case and must be adapted accordingly. Another possibility to use data sets with different cardinality and therefore work with rectangular matrices within the Hungarian algorithm, is the algorithm presented by F. Bourgeois and J-C. Lassalle [14].

5 Discussion and Outlook

The STIDes approach is capable of processing different kinds of similarity views because of the capability to set different weight parameters λ_i according to each specific use case. However, this results in an additional effort in the basic setting of the method since the parameters must be set separately for each application. In addition, the creation of dummy nodes allows a determination of the similarity of two unequal data sets, but the remaining intervals do not yet influence the computation of similarity. Our approach incorporates the possibility to apply global changes (e.g. scaling or time shifts) to one of the data sets and we showed that for both scaling and shifting a optimal factor exists, such that the similarity between the two data sets is then optimal.

In the future, we will focus our research on these global changes like time shifts and scaling. We need to research, if the computing time regarding both defined functions 12 and 13 is still polynomial and how the solution can be computed efficiently. We will also investigate the combined effect of both time shifts and scaling. This combined influence can be represented by the structure of the method as a multidimensional function. However, to what extent this affects the complexity of the calculations must also be examined. The possibility to apply different shift and/or scaling factors to different groups of intervals within one data set is also an interesting case, which will be studied in future research. Within the future research, differences in the cardinality of the data sets will again be looked upon to be able to set influence parameters for the similarity measure.

6 Conclusion

At the beginning of this work, it was determined that in today's optimized production processes deviations can lead to unwanted time delays and additional costs. It turned out that a re-recognition of similar deviations from the past leads to a faster and more effective reaction possibility. In order to allow a quantification of similar situations, a similarity criterion on the basis of time interval data sets has been derived in this work, which compares the intervals themselves. For this purpose, a new similarity measure between two intervals was defined, which was transferred to a similarity measure of two data sets in a further step.

In this paper, a similarity measure depending on the relation of the intervals to each other was introduced. For this purpose, the properties of the intervals, such as the size of the overlap, start and end point distances were defined. From these properties distance values were derived, which in a weighted sum form the similarity measure of two intervals. This allows to individually weight each interval characteristic. On the basis of the weighted sum, the STIDes approach was defined, which compares two time interval data sets with one another. For this purpose, the minimum sum of the individual similarities is calculated over all possible interval pairs, which results in the defined similarity measure. An interval pair consists of an interval of each of the two considered time interval data sets. The possibility to weight each interval property is retained by this approach in the extended similarity measure of two data sets. In order to compensate for a possible cardinality difference between the data sets, dummy nodes were introduced, so that each interval can be assigned one partner from the other set and therefore the STIDes approach can be applied. The desired similarity measure of the possibly modified data sets is determined by the Hungarian algorithm in polynomial time ($O(n^3)$). The introduced methodology for identifying similarities also made it possible to incorporate global changes in the intervals of one data set into the analysis. In this context, it has been shown that the defined functions have a global minimum in order to be able to apply the

above described approach, but the complexity changes with the implementation of global changes is not yet researched.

Overall, the approach considered provides a versatile method for describing similarities, in which all properties of the intervals are included in the similarity analysis and, moreover, various types of dynamic changes within the data sets can be mapped. Due to the general representation of this methodology, the similarity analysis can be applied to a variety of problems and thus meets the goal of a general description of similarities between time interval data sets.

References

1. A. D. Jayal, F. Badurdeen, O.W. Dillon Jr. and I.S. Jawahir: *Sustainable manufacturing: Modeling and optimization challenges at the product, process and system levels.* CIRP Journal of Manufacturing Science and Technology 2.3 144-152 (2010)
2. W. B. Lee and H. C. W. Lau: *Factory on demand: the shaping of an agile production network.* International Journal of Agile Management Systems 1.2 83-87 (1999)
3. D. Metzler, S. Dumais and C. Meek: *Similarity measures for short segments of text.* European Conference on Information Retrieval, Springer Berlin Heidelberg (2007)
4. G. Yu, F. Li, Y. Qin, X. Bo, Y. Wu and S. Wang: *GoSemSim: an R package for measuring semantic similarity amoing GO terms and gene products* Bioinformatics vol. 26, no. 7, 976 - 978 (2010)
5. H. Ogata et al: *A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters* Nucleic acids research 28.20, 4021 - 4028 (2000)
6. A. Wang: *An Industrial Strength Audio Search Algorithm.* ISMIR (2003)
7. O. Kostakis, P. Papepetrou and J. Hollmèn: *ARTEMIS: assessing the similarity of event-interval sequences.* Machine Learning and Knowledge Discovery in Databases, 229 - 244 (2011)
8. P. Meisen, D. Keng, T. Meisen, M. Recchioni and S. Jeschke: *Similarity Search of Bounded TIDASETS within Large Time Interval Databases.* International Conference on Computational Science and Computational Intelligence (2015)
9. J. Kruscall and M. Liberman: The symmetric time warping algorithm: From continuous to discrete. Time Warps, String Edits, and Macromolocules: The theory and Practice of String Comparison.
10. Y. Chen, M. Chiang and M. Ko: *Discovering time-interval sequential patterns in sequence databases* Expert Systems with Applications 25.3, 343 - 354 (2003)
11. R. Sadasivam and K. Duraiswamy: *Efficient approach to discover interval-based sequential patterns* Journal of Computer Science 9.2, 225 (2013)
12. C. Koncilia, T. Morzy, R. Wrembel and J. Eder: *Interval OLAP: Analyting Interval Data* International Conference on Data Warehousing and Knowledge Discovery (2014).
13. J. Munkres: *Algorithms for the assignment and transportation problems* Journal of the Society for Industrial and Applied Mathematics 5.1, 32 - 38 (1957)
14. F. Bourgeois and J-C. Lassalle: An extension of the Munkres algorithm for the assignment problem to rectangular matrices. Communications of the ACM 14.12, 802-804 (1971)