Tatra Mt. Math. Publ. **36** (2007), 39–59

TATRA MOUNTAINS
Mathematical Publications

A NEW METHOD OF MEASURING SIMILARITY FOR A SPECIAL CLASS OF DIRECTED GRAPHS

Matthias Dehmer — Alexander Mehler

ABSTRACT. The problem of graph similarity is challenging and important in many areas of science, e.g., mathematics [Sobik, F.: *Graphmetriken und Klassifikation strukturierter Objekte*, ZKI-Informationen, Akad. Wiss. DDR, **2**, (1982), 63–122]; [Zelinka, B.: *On a certain distance between isomorphism classes of graphs*, Čas. Pěst. Mat., **100**, (1975), 371–373], biology [Koch, I., Lengauer, T.; Wanke, E.: *An algorithm for finding maximal common subtopologies in a set of protein structures*, J. Comput. Biology, **3**, (1996), 289–306], and chemistry [Skvortsova, M. I., Baskin, I. I., Stankevich, I. V., Palyulin, V. A., Zerirow, N. S.: *Molecular similarity in structure-property relationship studies. Analytical description of the complete set of graph similarity measures*, International Symposium CACR'96, (1996) pp. 542–646]. In this paper, we design a new method, which uses sequence alignment techniques [Altschul, S. F., Gish, W., Miller, V., Myers, E. W., Lipman, D. J.: *Basic local alignment search tool*, J. Molecular Biology **125**, (1991), 403–410]; [Altschul, S. F., Madden, T. L., Miller, W., Schaffer, A. A., Zhang, J., Zhang, Z., Lipman, D. J.: *Gapped BLAST and PSI–BLAST: a new generation of protein database search programs*, Nucleic Acids Res. **25**, (1997), 3389–3402], [Kilian, J., Hoos, H. H.: *MusicBLAST–gapped sequence alignment for MIR*, in: Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR), (2004)], to measure the structural similarity of unlabeled, hierarchical, and directed graphs. More precisely, if h denotes the maximal length of a path from the root to a leaf of a given hierarchical and directed graph $\hat{\mathcal{H}}$, we align out-degree and in-degree sequences induced by the vertex sequences on a level i, $0 \leqslant i \leqslant h$. On the basis of the level alignments, we construct measured values and prove that they are similarity measures. In our algorithm, which uses the well-known technique of dynamic programming, the alignments of out-degree and in-degree sequences are decoupled. Therefore, we obtain a family $(d_i(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2))_{1 \leqslant i \leqslant 3}$ of graph similarity measures. As an application, we examine the measures on a graph corpus of 464 graphs, where the graphs represent web-based hypertext structures (websites).

2000 Mathematics Subject Classification: Primary 05C75, 05C20, 68R15; Secondary 90C39, 68R10, 91B82.
Keywords: digraphs, similarity measures, sequence alignments, degree sequences.

1. Introduction

In this paper, we introduce novel graph similarity measures for a special class of directed graphs. First, we have to define the term graph similarity. In general, the concept of similarity is not unique and thus cannot be completely formalized. If we consider two structured objects, we can refer to several similarity aspects, e.g., semantic similarity, structural similarity, and functional similarity. In the following, we introduce similarity measures on unlabeled, hierarchical and directed graphs, which are based on structural similarity. In order to construct such a measure, we have to take into account the structural properties and parameters of the graphs under consideration.

Measures of distances between graphs have been frequently investigated. In the literature, the problem of computing the distance between graphs is often called *inexact graph matching* [7]. Many similarity measures on graphs are based on isomorphic relations and subgraph isomorphism [25], [31], respectively. An example of such a similarity measure is the well-known Zelinka-distance [32]. The Zelinka-distance is based on the principle that the more similar two graphs are, the bigger the common induced subgraph is. In other words, graphs which have a large common induced subgraph, have a small distance and vice versa. Zelinka was the first to introduce this measure for unlabeled graphs. Sobik [28] [29] and Kaden [16], [17] generalized this measure for arbitrary (also labeled) graphs of different order and proved that it is a metric. But, for large graphs, the complexity is considered to be unacceptable for practical use and Kaden has obtained further similarity measures on graphs by transforming the graphs by injective mapping. For example, Kaden [16] considered *line graphs* [4] and used the Zelinka-distance to compute the similarity of the transformed graphs.

Shapiro [26] introduced a known similarity metric for graphs based on the corresponding adjacency matrices. Let the graphs be G_1, G_2 and the corresponding adjacency matrices A_1, A_2. Permute the rows and the columns in the matrix A_2 in such a way that the matrix elements conform with the matrix elements of A_1 as much as possible. Now, Shapiro defines the graph distance between G_1, G_2 by the minimal number of dissenting matrix elements and proves that it is a metric.

In addition, an important class of similarity measures based on the *edit distance* of graphs has been investigated by [33] [34]. The edit distance is based on basic weighted transformation steps like deletions, substitutions, and insertions of vertices and edges. Since there is an infinite number of different possibilities for transforming G_2 in G_1, the similarity of the graphs is defined as the minimum cost of transformations.

Other approaches from the inexact graph matching paradigm consider distances between graphs on the basis of *graph grammars* [10], [24]. These methods are primarily interesting for theoretical aspects but not for practical use since the specific grammar is difficult to obtain.

This paper is organized as follows: In the next Section (2) we state some fundamental definitions and explain topological properties of our graphs in terms of out-degree and in-degree sequences. In Section 3 we present our new approach for measuring the structural similarity of unlabeled, hierarchical, and directed graphs. In Section (4) we apply our new method to a graph corpus C_G and state the experimental results. The paper finishes in Section (5) with a summary and conclusion.

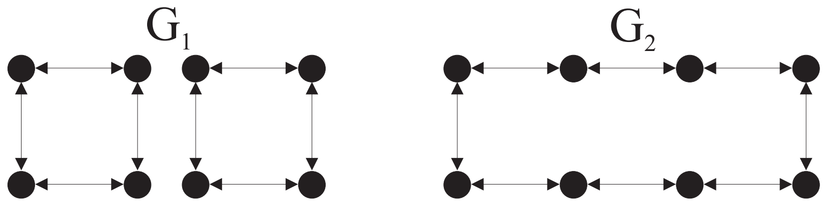

FIGURE 1.1. Two directed graphs with the same degree sequences but $G_1 \not\tilde{=} G_2$.

2. Fundamental definitions and topological aspects of graphs

DEFINITION 2.1. Let $\mathcal{H} = (V, E)$, $E \subseteq V \times V$, $|V| < \infty$ be a directed graph.

$\mathcal{N}^+(v) := \{\tilde{v} \in V \backslash \{v\} | (v, \tilde{v}) \in E\}$ denotes the set of out-neighbours of v,
$\mathcal{N}^-(v) := \{\tilde{u} \in V \backslash \{v\} | (\tilde{u}, v) \in E\}$ denotes the set of in-neighbours of v,
$\delta_{out}(v) := |\mathcal{N}^+(v)|$,
$\delta_{in}(v) := |\mathcal{N}^-(v)|$.

$s_j^{out}(\mathcal{H}) \in \mathbb{N}$, $0 \leqslant j \leqslant k_{out} := \max_{v \in V}\{\delta_{out}(v)\}$ (or $s_i^{in}(\mathcal{H}) \in \mathbb{N}$, $0 \leqslant i \leqslant k_{in} := \max_{v \in V}\{\delta_{in}(v)\}$) denotes the number of vertices of $\mathcal{H}$ with out-degree j (or in-degree i). The vector

$$s^{out}(\mathcal{H}) := \left(s_0^{out}(\mathcal{H}), s_1^{out}(\mathcal{H}), \ldots, s_{k_{out}}^{out}(\mathcal{H})\right),$$

or

$$s^{in}(\mathcal{H}) := \left(s_0^{in}(\mathcal{H}), s_1^{in}(\mathcal{H}), \ldots, s_{k_{in}}^{in}(\mathcal{H})\right)$$

is called the out-degree (or in-degree) sequence of $\mathcal{H}$.

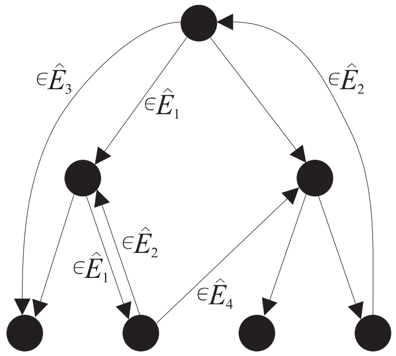

FIGURE 2.1. Edge types of Definition 2.2.

Degree sequences can be useful for describing the structure of a graph. As a preliminary, we note an important property of degree sequences. If G_1, G_2 are arbitrary directed graphs, then the following assertion holds: $G_1 \cong G_2 \Longrightarrow s^{out}(G_1) = s^{out}(G_2) \wedge s^{in}(G_1) = s^{in}(G_2)$. The reverse assertion is not always true. For example, the degree sequences of the graphs G_1, G_2 shown in Figure (1.1) are $s^{out}(G_1) = (0,0,8) = s^{out}(G_2) = (0,0,8)$, $s^{in}(G_1) = (0,0,8) = s^{in}(G_2) = (0,0,8)$, but $G_1 \ncong G_2$. Therefore, we have to examine how far degree sequences are applicable for measuring the structural similarity of our graphs.

Another important question is: Given the sequences of natural numbers $(s_0, s_1, \cdots, s_p)$ and $(\hat{s}_0, \hat{s}_1, \cdots, \hat{s}_{\hat{p}})$ under what conditions are the two sequences (out-degree and in-degree) sequences of a certain graph H? This problem has been solved by [12], [13], [14]. Now, we will construct the class of graphs that we want to examine in this paper.

DEFINITION 2.2. Let be $\mathcal{H} = (V, E)$, $E \subseteq V \times V$, $|V| < \infty$ and $\hat{V} := \{v_{0,1}, v_{1,1}, v_{1,2}, \ldots, v_{1,\sigma_1}, v_{2,1}, v_{2,2}, \ldots, v_{2,\sigma_2}, \ldots, v_{h,1}, v_{h,2}, \ldots, v_{h,\sigma_h}\}$. Define h as the maximal length of a path from the root $v_{0,1}$ to a leaf. $v_{i,j}$ denotes the jth vertex on the ith level, $0 \leqslant i \leqslant h$, $1 \leqslant j \leqslant \sigma_i$. σ_i is maximal in the sense that there is no other vertex sequence such that $v_{i,1}, v_{i,2}, \ldots, v_{i,\hat{\sigma}_i}$ with $\hat{\sigma}_i > \sigma_i$. $\mathcal{L}\colon \hat{V} \longrightarrow \mathbb{N}$, $\mathcal{L}(v_{i,j}) := i$ is a function which determines the level of a vertex $v_{i,j}$. The edge types are now defined by:

$$\begin{aligned}\hat{E}_1 := \big\{&(v_{i,\nu}, v_{i+1,\nu_j}) | v_{i,\nu}, v_{i+1,\nu_j} \in \hat{V},\ 1 \leqslant j \leqslant k,\ k := \delta_{out}(v_{i,\nu}),\\ &\mathcal{L}(v_{i+1,\nu_j}) = \mathcal{L}(v_{i,\nu}) + 1 \wedge \big((\nexists\,(v_{i,\bar{\nu}}, v_{i+1,\nu_k}),\ \bar{\nu} > \nu)\\ &\quad \vee (\nexists\,(v_{i,\hat{\nu}}, v_{i+1,\nu_1}), \hat{\nu} < \nu)\big), \qquad \nu_1 < \nu_2 < \cdots < \nu_k\big\}, \end{aligned} \tag{2.1}$$

$$\begin{aligned}\hat{E}_2 := \big\{&(v_{i+s,\nu}, v_{i,\bar{\nu}}) | v_{i+s,\nu}, v_{i,\bar{\nu}} \in \hat{V},\ \mathcal{L}(v_{i,\bar{\nu}}) = \mathcal{L}(v_{i+s,\nu}) - s, s \leqslant h \\ &\wedge \exists! \big(\underbrace{(v_{i,\bar{\nu}}, v_{i+1,\nu_1})}_{\in \hat{E}_1}, \dots, \underbrace{(v_{i+s-1,\nu_j}, v_{i+s,\nu})}_{\in \hat{E}_1}\big), 1 \leqslant \bar{\nu} \leqslant \sigma_i, 1 \leqslant \nu_1 \leqslant \sigma_{i+1}, \cdots, \\ &1 \leqslant \nu_j \leqslant \sigma_{i+s-1}, 1 \leqslant \nu \leqslant \sigma_{i+s}\big\},\end{aligned} \tag{2.2}$$

$$\begin{aligned}\hat{E}_3 := \big\{&(v_{i,\bar{\nu}}, v_{i+s,\nu}) | v_{i,\bar{\nu}}, v_{i+s,\nu} \in \hat{V},\ \mathcal{L}(v_{i+s,\nu}) = \mathcal{L}(v_{i,\bar{\nu}}) + s, 1 < s \leqslant h \\ &\wedge \exists! \big(\underbrace{(v_{i,\bar{\nu}}, v_{i+1,\nu_1})}_{\in \hat{E}_1}, \dots, \underbrace{(v_{i+s-1,\nu_j}, v_{i+s,\nu})}_{\in \hat{E}_1}\big), 1 \leqslant \bar{\nu} \leqslant \sigma_i, 1 \leqslant \nu_1 \leqslant \sigma_{i+1}, \cdots, \\ &1 \leqslant \nu_j \leqslant \sigma_{i+s-1}, 1 \leqslant \nu \leqslant \sigma_{i+s}\big\},\end{aligned} \tag{2.3}$$

$$\begin{aligned}\hat{E}_4 := &\big\{(v_{i,\nu}, v_{i,\bar{\nu}}) | v_{i,\nu}, v_{i,\bar{\nu}} \in \hat{V},\ \mathcal{L}(v_{i,\nu}) = \mathcal{L}(v_{i,\bar{\nu}}) \wedge (\nu < \bar{\nu} \vee \nu > \bar{\nu})\big\} \\ &\cup\ \big\{(v_{i+s,\nu}, v_{i,\bar{\nu}}) | v_{i+s,\nu}, v_{i,\bar{\nu}} \in \hat{V},\ (v_{i+s,\nu}, v_{i,\bar{\nu}}) \notin \hat{E}_2,\ \mathcal{L}(v_{i,\bar{\nu}}) \\ &= \mathcal{L}(v_{i+s,\nu}) - s, s \leqslant h\big\} \\ &\cup\ \big\{(v_{i,\nu}, v_{i+s,\bar{\nu}}) | v_{i,\nu}, v_{i+s,\bar{\nu}} \in \hat{V},\ (v_{i,\nu}, v_{i+s,\bar{\nu}}) \notin \hat{E}_1, \hat{E}_3,\ \mathcal{L}(v_{i+s,\bar{\nu}}) \\ &= \mathcal{L}(v_{i,\nu}) + s, s \leqslant h\big\}.\end{aligned} \tag{2.4}$$

Then $\hat{\mathcal{H}} := (\hat{V}, \hat{E})$, $\hat{E} := \hat{E}_1 \cup \hat{E}_2 \cup \hat{E}_3 \cup \hat{E}_4$ denotes the hierarchical and directed graph of $\mathcal{H}$.

We call the elements of the edge set of $\hat{\mathcal{H}}$ *Kernel-edges* (2.1), *Up-edges* (2.2), *Down-edges* (2.3) and *Across-edges* (2.4). The properties of each edge type are, briefly [23]:

- *Kernel-edges*: The *Kernel hierarchy* is induced by the Kernel-edges. Kernel-edges associate dominating nodes with their immediately dominated successor nodes.
- *Up-edges* associate analogously nodes of the Kernel hierarchy with one of their (dominating) predecessor nodes.
- *Down-edges* associate nodes of the Kernel hierarchy with one of their (dominated) successor nodes in terms of that Kernel hierarchy.
- *Across-edges* associate nodes of the Kernel hierarchy, none of which is an (im- mediate) predecessor of the other in terms of the Kernel hierarchy.

The Definition (2.2) provides a structural property of $\hat{\mathcal{H}}$ which is evident.

Proposition 2.1. *Let* $\hat{\mathcal{H}} := (\hat{V}, \hat{E})$. $\hat{\mathcal{H}}_{\mathrm{T}} := (\hat{V}, E_{\mathrm{T}})$, $E_{\mathrm{T}} := \hat{E} \backslash \{\hat{E}_2, \hat{E}_3, \hat{E}_4\}$ *is a directed rooted tree.*

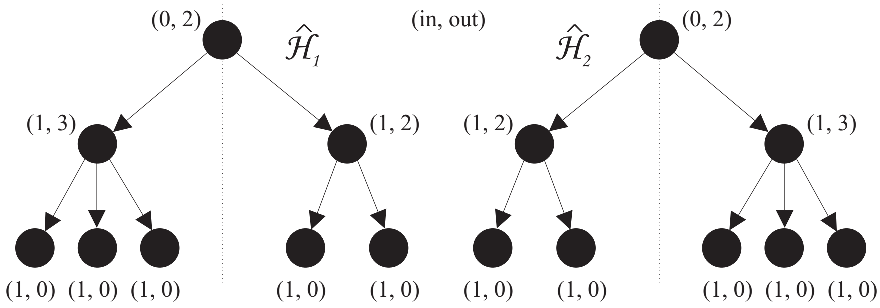

FIGURE 2.2. Two asymmetric graphs with the same degree sequences.

COROLLARY 2.2. *$\hat{\mathcal{H}}_{\mathrm{T}}^{\mathrm{rec}} := (w, \hat{\mathcal{H}}_{T_1}, \hat{\mathcal{H}}_{T_2}, \ldots, \hat{\mathcal{H}}_{T_{\delta_{out}(w)}})$ is the recursive description of $\hat{\mathcal{H}}_{\mathrm{T}}$.*

Now we will provide examples of graphs which fulfil Definition (2.2), but whose topology cannot be adequately described by degree sequences. With reference to the next chapter, we note that the vertices on each level i induce an out-degree and in-degree sequence in a left to right order. In particular, the out-degrees from each vertex $v_{i,j}$ on a level i, $0 \leqslant i \leqslant h$ are significant for the embeddings of the substructures associated with $v_{i,j}$.

By Definition (2.1), the graphs in Figure (2.2) have the same out-degree and in-degree sequences, $s^{out}(\hat{\mathcal{H}}_1) = (5, 0, 2, 1) = s^{out}(\hat{\mathcal{H}}_2) = (5, 0, 2, 1) \wedge s^{in}(\hat{\mathcal{H}}_1) = (1, 7) = s^{in}(\hat{\mathcal{H}}_2) = (1, 7)$. But they are not symmetrically located on the symmetry axis, indicated by the dashed lines. Therefore, we see that degree sequences of the given graphs have no influence on the embeddings of substructures in the graph.

Another example is Figure (2.3). It holds that $s^{out}(\hat{\mathcal{H}}_1) = (6, 1, 3, 0, 1) = s^{out}(\hat{\mathcal{H}}_2) = (6, 1, 3, 0, 1) \wedge s^{in}(\hat{\mathcal{H}}_1) = (0, 11) = s^{in}(\hat{\mathcal{H}}_2) = (0, 11)$. Nonetheless, $\hat{\mathcal{H}}_1$ and $\hat{\mathcal{H}}_2$ possess different topologies. Altogether, it follows that simple comparisons of out-degree and in-degree vectors are not suitable for determining the structural similarity of our graphs.

3. New approach for measuring the structural similarity of graphs

In Section (2) we saw that degree sequences cannot describe the topology of a graph completely. Since we are examining unlabeled, hierarchical and directed graphs, in the following, we will look at the out-degree and in-degree sequences (on a level i), induced by the vertex sequences $v_{i,1}, v_{i,2}, \ldots, v_{i,\sigma_i}$ and their edge relations in terms of Definition (2.2). Now, with respect to a *cost function* α the

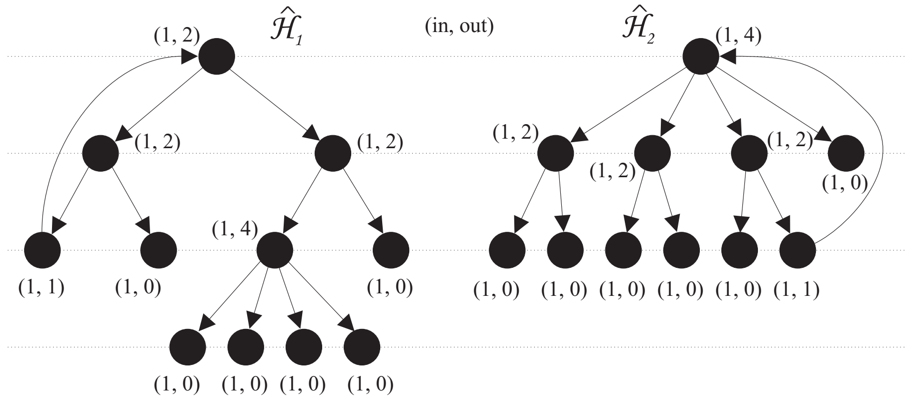

FIGURE 2.3. Two graphs with the same degree sequences but different topology.

more similar the out-degree and in-degree alignments on the levels i, $0 \leqslant i \leqslant h$ are, the more similar the common structure of the graphs is and vice versa. The cost function α, which we will define later, weighs alignments on the sequences $S_0^{\hat{\mathcal{H}}}$, $S_1^{\hat{\mathcal{H}}}, \ldots, S_h^{\hat{\mathcal{H}}}$ under certain conditions. With $w_1^{\hat{\mathcal{H}}} := v_{0,1}^{\hat{\mathcal{H}}}$, the structural embedding of a graph $\hat{\mathcal{H}}$ is essentially described by the sequences

$$
\begin{aligned}
S_0^{\hat{\mathcal{H}}} &:= w_1^{\hat{\mathcal{H}}}, \\
S_1^{\hat{\mathcal{H}}} &:= v_{1,1}^{\hat{\mathcal{H}}} \circ v_{1,2}^{\hat{\mathcal{H}}} \circ \cdots \circ v_{1,\delta_{out}(w_1^{\hat{\mathcal{H}}})}^{\hat{\mathcal{H}}}, \\
&\ \ \vdots \\
S_h^{\hat{\mathcal{H}}} &:= v_{h,1}^{\hat{\mathcal{H}}} \circ v_{h,2}^{\hat{\mathcal{H}}} \circ \cdots \circ v_{h,\sigma_h}^{\hat{\mathcal{H}}},
\end{aligned}
$$

together with their out-degree and in-degree sequences on a level i.

Now, the problem of determining the structural similarity between $\hat{\mathcal{H}}_1$ and $\hat{\mathcal{H}}_2$ is equivalent to determining the optimal alignment of

$$
\begin{aligned}
S_0^{\hat{\mathcal{H}}_1} &:= w_1^{\hat{\mathcal{H}}_1}, \\
S_1^{\hat{\mathcal{H}}_1} &:= v_{1,1}^{\hat{\mathcal{H}}_1} \circ v_{1,2}^{\hat{\mathcal{H}}_1} \circ \cdots \circ v_{i,\delta_{out}(w_1^{\hat{\mathcal{H}}_1})}^{\hat{\mathcal{H}}_1}, \\
&\ \ \vdots \\
S_{h_1}^{\hat{\mathcal{H}}_1} &:= v_{h_1,1}^{\hat{\mathcal{H}}_1} \circ v_{h_1,2}^{\hat{\mathcal{H}}_1} \circ \cdots \circ v_{h_1,\sigma_{h_1}}^{\hat{\mathcal{H}}_1},
\end{aligned}
$$

and

$$S_0^{\hat{\mathcal{H}}_2} := w_2^{\hat{\mathcal{H}}_2},$$
$$S_1^{\hat{\mathcal{H}}_2} := v_{1,1}^{\hat{\mathcal{H}}_2} \circ v_{1,2}^{\hat{\mathcal{H}}_2} \circ \cdots \circ v^{\hat{\mathcal{H}}_2}_{1,\delta_{out}(w_2^{\hat{\mathcal{H}}_2})},$$
$$\vdots$$
$$S_{h_2}^{\hat{\mathcal{H}}_2} := v_{h_2,1}^{\hat{\mathcal{H}}_2} \circ v_{h_2,2}^{\hat{\mathcal{H}}_2} \circ \cdots \circ v_{h_2,\sigma_{h_2}}^{\hat{\mathcal{H}}_2},$$

with respect to a cost function α. Based on these vertex sequences, we will later implement alignments of the corresponding out-degree and in-degree sequences. We regard this task as an optimization problem, in the sense of finding a minimum score path in an *alignment graph* (see Definition (3.1)). For this purpose, we use the method of *dynamic programming*, which is based on the *optimality principle* of Bellman [6].

Generally, the method of dynamic programming can be viewed as an optimization of a process $\mathcal{P}_{fin} = \mathcal{P}_1 \circ \mathcal{P}_2 \circ \cdots \circ \mathcal{P}_n$. Starting from the initial state $\mathcal{P}_0$ and a *control* c_1, the new state $\mathcal{P}_1$ will be calculated with a transition function $\mathcal{P}_1 = f_1(\mathcal{P}_0, c_1)$, analogous for $\mathcal{P}_2 = f_2(\mathcal{P}_1, c_2), \ldots, \mathcal{P}_n = f_n(\mathcal{P}_{n-1}, c_n)$. Now, we have to optimize a target function $F(\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_n, c_1, \ldots, c_n)$. F is additive such that there exists function add_i with $F = \sum_{i=1}^{n} \mathrm{add}_i(\mathcal{P}_{i-1}, c_i)$.

Now, let $\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2, v_{i,j}^{\hat{\mathcal{H}}_1}$, $0 \leqslant i \leqslant h_1$, $1 \leqslant j \leqslant \sigma_i$ denote the jth vertex on the ith level of $\hat{\mathcal{H}}_1$, analogous to $v_{i,j}^{\hat{\mathcal{H}}_2}$ for $\hat{\mathcal{H}}_2$. We define the sequences representing graphs

$$S_1 := w_1^{\hat{\mathcal{H}}_1} \circ v_{1,1}^{\hat{\mathcal{H}}_1} \circ v_{1,2}^{\hat{\mathcal{H}}_1} \circ \cdots \circ v_{h_1,\sigma_{h_1}}^{\hat{\mathcal{H}}_1}, \tag{3.1}$$

$$S_2 := w_2^{\hat{\mathcal{H}}_2} \circ v_{1,1}^{\hat{\mathcal{H}}_2} \circ v_{1,2}^{\hat{\mathcal{H}}_2} \circ \cdots \circ v_{h_2,\sigma_{h_2}}^{\hat{\mathcal{H}}_2}. \tag{3.2}$$

$S_k[i]$ denotes the ith position of the sequences S_k and it holds $S_1[n] = v_{h_1,\sigma_{h_1}}^{\hat{\mathcal{H}}_1}, S_2[m] = v_{h_2,\sigma_{h_2}}^{\hat{\mathcal{H}}_2}$, $\mathbb{N} \ni n, m \geqslant 1$, $S_k[1] = w_k^{\hat{\mathcal{H}}_k}$, $k \in \{1, 2\}$. In order to find the optimal alignment between S_1, S_2, we construct the alignment graph $G_{S_1,S_2} := (V_{S_1,S_2}, E_{S_1,S_2}, f_{E_{S_1,S_2}})$ with an edge labeling function $f_{E_{S_1,S_2}} : E_{S_1,S_2} \longrightarrow \mathbb{R}_+$.

Definition 3.1. Let $V_{S_1,S_2} := \{(i,j) | 0 \leqslant i \leqslant n,\ 0 \leqslant j \leqslant m\}$, $e_{Del} := (i-1, j) \to (i, j), e_{Ins} := (i, j-1) \to (i, j), e_{Subst} := (i-1, j-1) \to (i, j)$. The edge set E_{S_1,S_2} is now defined by

$$\begin{aligned} E_{S_1,S_2} := & \{e_{Del} | f_{E_{S_1,S_2}}(e_{Del}) = [S_1[i], -],\ i \in [1, n]\} \\ & \cup \{e_{Ins} | f_{E_{S_1,S_2}}(e_{Ins}) = [-, S_2[j]],\ j \in [1, m]\} \\ & \cup \{e_{Subst} | f_{E_{S_1,S_2}}(e_{Subst}) = [S_1[i], S_2[j]],\ i \in [1, n],\ j \in [1, m]\}. \end{aligned}$$

$G_{S_1,S_2} := (V_{S_1,S_2}, E_{S_1,S_2}, f_{E_{S_1,S_2}})$ denotes the alignment graph of the sequences S_1 and S_2.

$(i-1,j) \to (i,j)$ equals the deletion of $S_1[i]$ in S_1, $(i,j-1) \to (i,j)$ equals the insertion of $S_2[j]$ in S_1 at the ith position, and $(i-1,j-1) \to (i,j)$ equals the substitution $S_1[i]$ to $S_2[j]$.

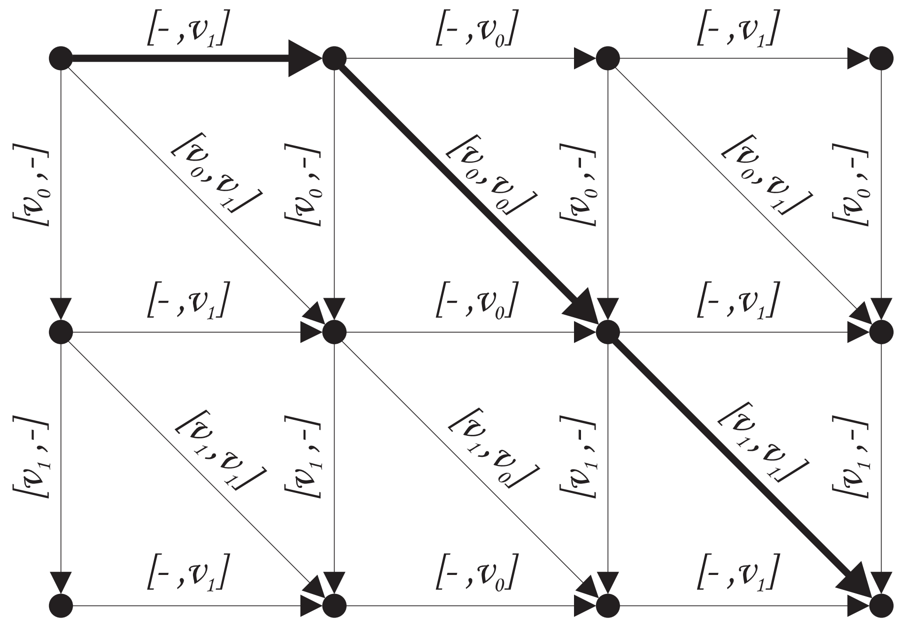

FIGURE 3.1. Alignment graph G_{S_1,S_2} of the sequences S_1, S_2.

As an application, we consider the two graphs $\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2$ represented as the sequences $S_1 := v_0 \circ v_1$ and $S_2 := v_1 \circ v_0 \circ v_1$ (simplified notation). The corresponding alignment graph is shown in Figure (3.1). From this figure we can read off (bold edges) the alignment:

$$\begin{array}{ccc} - & v_0 & v_1 \\ v_1 & v_0 & v_1 \end{array}$$

According to the edge labeling function $f_{E_{S_1,S_2}} : E_{S_1,S_2} \longrightarrow \mathbb{R}_+$ and for each possible aligned pair $[a,b]$ a cost function $\alpha([a,b]) \in \mathbb{R}_+$ is assigned, where a, b are sequence entries of S_1 and S_2 or the gap symbol $-$. Our algorithm with the complexity $O\big(|\hat{V}_1| \cdot |\hat{V}_2|\big)$ for finding the optimal alignment of the Sequences (3.1), (3.2) generates a matrix $\big(\mathcal{M}(i,j)\big)_{ij}$, $0 \leqslant i \leqslant n$, $0 \leqslant j \leqslant m$, where $\mathcal{M}(i,j)$ is equivalent to the minimal edit distance between the sequences $\tilde{S}_1, \tilde{S}_2$. Thereby $\tilde{S}_1$, $\tilde{S}_2$ consist of the first ith, jth characters of S_1 and S_2.

Hence, we are searching for $\mathcal{M}(n,m)$ because $\mathcal{M}(n,m)$ is the minimal edit distance

$$d(S_1, S_2) := \min_{S_1 \to S_2} \sum \alpha\big([a,b]\big)\,. \tag{3.3}$$

We find the optimal alignment by tracing back along the minimal values from $\mathcal{M}(n,m)$ to $\mathcal{M}(0,0)$, since we notionally assign pointers to the edit operations. It is well known that the Distance (3.3) is a metric developed by Levenstein [21]. The algorithm is recursive and may be stated as

$$\mathcal{M}(0,0) := 0\,, \tag{3.4}$$

$$\mathcal{M}(i,0) := \mathcal{M}(i-1,0) + \alpha(S_1[i], -)\colon 1 \leqslant i \leqslant n\,, \tag{3.5}$$

$$\mathcal{M}(0,j) := \mathcal{M}(0,j-1) + \alpha(-, S_2[j])\colon 1 \leqslant j \leqslant m\,, \tag{3.6}$$

$$\mathcal{M}(i,j) := \min \begin{cases} \mathcal{M}(i-1,j) + \alpha(S_1[i], -) \\ \mathcal{M}(i,j-1) + \alpha(-, S_2[j])\colon i \in [1,n],\ j \in [1,m] \\ \mathcal{M}(i-1,j-1) + \alpha\big(-, S_1[i], S_2[j]\big)\,. \end{cases} \tag{3.7}$$

The Equations (3.4), (3.5), (3.6) indicate the initial conditions. They also define the penalty of unpaired elements at the outset of the sequences S_1, S_2. Definition (3.7) states that all possible transformations (edit operations), that is to say deletions, insertions, and substitutions, have influence on the algorithm. We have now described how the recursive algorithm, by means of the Equations (3.4), (3.5), (3.6) and (3.7), finds the optimal alignment of the sequences S_1, S_2. In order to construct our graph similarity measures, in the following we define the functions $\alpha^{out}, \alpha^{in}$.

$$\alpha^{out}\left(v^{\hat{\mathcal{H}}_1}_{i_1,j_1}, v^{\hat{\mathcal{H}}_2}_{i_2,j_2}\right) := \begin{cases} \beta^{out}\left(\delta_{out}(v^{\hat{\mathcal{H}}_1}_{i_1,j_1}), \delta_{out}(v^{\hat{\mathcal{H}}_2}_{i_2,j_2}), \sigma^1_{out}\right) & : \ i_1 = i_2\,, \\ +\infty & : \ else\,, \end{cases} \tag{3.8}$$

$0 \leqslant i_k \leqslant h_k$, $1 \leqslant j_k \leqslant \sigma_{i_k}$, whereas $\beta^{out}(x, y, \sigma^k_{out}) := 1 - e^{-\frac{1}{2}\frac{(x-y)^2}{\left(\sigma^k_{out}\right)^2}}$, $x, y, \sigma^k_{out} \in \mathbb{R}$, $k \in \{1,2\}$, and

$$\alpha^{out}\left(v^{\hat{\mathcal{H}}_1}_{i,j_1}, -\right) := \beta^{out}\left(\delta_{out}(v^{\hat{\mathcal{H}}_1}_{i,j_1}), \xi, \sigma^2_{out}\right)\,, \tag{3.9}$$

$$\alpha^{out}\left(-, v^{\hat{\mathcal{H}}_2}_{i,j_2}\right) := \beta^{out}\left(\xi, \delta_{out}(v^{\hat{\mathcal{H}}_2}_{i,j_2}), \sigma^2_{out}\right)\,. \tag{3.10}$$

With $\beta^{in}(x, y, \sigma^k_{in}) := 1 - e^{-\frac{1}{2}\frac{(x-y)^2}{\left(\sigma^k_{in}\right)^2}}$ we define straightforward

$$\alpha^{in}\left(v^{\hat{\mathcal{H}}_1}_{i_1,j_1}, v^{\hat{\mathcal{H}}_2}_{i_2,j_2}\right) := \begin{cases} \beta^{in}\left(\delta_{in}(v^{\hat{\mathcal{H}}_1}_{i_1,j_1}), \delta_{in}(v^{\hat{\mathcal{H}}_2}_{i_2,j_2}), \sigma^1_{in}\right) & : \ i_1 = i_2\,, \\ +\infty & : \ else\,, \end{cases} \tag{3.11}$$

$$\alpha^{in}\left(v_{i,j_1}^{\hat{\mathcal{H}}_1}, -\right) := \beta^{in}\left(\delta_{in}(v_{i,j_1}^{\hat{\mathcal{H}}_1}), \xi, \sigma_{in}^2\right), \tag{3.12}$$

$$\alpha^{in}\left(-, v_{i,j_2}^{\hat{\mathcal{H}}_2}\right) := \beta^{in}\left(\xi, \delta_{in}(v_{i,j_2}^{\hat{\mathcal{H}}_2}), \sigma_{in}^2\right), \tag{3.13}$$

$\xi > 0$. The choice $\xi > 0$ in the Equations (3.9), (3.10) (3.12), (3.13) prevents an alignment between two leaves being better evaluated as an alignment between a leaf and a gap. The Definitions (3.8), (3.11) of the functions $\alpha^{out}, \alpha^{in}$ state that we do not align vertices on different levels. To prevent this, we set the gap penalty to $+\infty$, whereby our dynamic programming algorithm will never choose this cost-intensive path. In order to valuate the alignments on each level of the given graphs $\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2$, we define the functions

$$\gamma^{out}(\hat{\mathcal{H}}_k, i) := \frac{\sum_{j=1}^{\sigma_i^k} \hat{\alpha}_{out}\left(v_{i,j}^{\hat{\mathcal{H}}_k}, \text{ align }\left(v_{i,j}^{\hat{\mathcal{H}}_k}\right)\right)}{\sigma_i^k}, \tag{3.14}$$

$$\gamma^{in}(\hat{\mathcal{H}}_k, i) := \frac{\sum_{j=1}^{\sigma_i^k} \hat{\alpha}_{in}\left(v_{i,j}^{\hat{\mathcal{H}}_k}, \text{ align }\left(v_{i,j}^{\hat{\mathcal{H}}_k}\right)\right)}{\sigma_i^k}, \tag{3.15}$$

$k \in \{1, 2\}$. Once again, σ_i^k denotes the upper index of a vertex on level i related to $\hat{\mathcal{H}}_k$. To complete the Definitions (3.14), (3.15), we define the mapping align and $\hat{\alpha}_{out}, \hat{\alpha}_{in}$ as follows:

$$\text{align }\left(v_{i,j_1}^{\hat{\mathcal{H}}_1}\right) := \begin{cases} v_{i,j_2}^{\hat{\mathcal{H}}_2} & : \text{ align}^{-1}\left(v_{i,j_2}^{\hat{\mathcal{H}}_2}\right) = v_{i,j_1}^{\hat{\mathcal{H}}_1}, \\ - & : else, \end{cases} \tag{3.16}$$

$$\hat{\alpha}^{out}\left(v_{i,j_1}^{\hat{\mathcal{H}}_1}, -\right) := \beta^{out}\left(\delta_{out}(v_{i,j_1}^{\hat{\mathcal{H}}_1}), \xi, \hat{\sigma}_{out}^1\right), \tag{3.17}$$

$$\hat{\alpha}^{out}\left(-, v_{i,j_2}^{\hat{\mathcal{H}}_2}\right) := \beta^{out}\left(\xi, \delta_{out}(v_{i,j_2}^{\hat{\mathcal{H}}_2}), \hat{\sigma}_{out}^1\right), \tag{3.18}$$

$$\hat{\alpha}^{out}\left(v_{i,j_1}^{\hat{\mathcal{H}}_1}, v_{i,j_2}^{\hat{\mathcal{H}}_2}\right) := \beta^{out}\left(\delta_{out}(v_{i,j_1}^{\hat{\mathcal{H}}_1}), \delta_{out}(v_{i,j_2}^{\hat{\mathcal{H}}_2}), \hat{\sigma}_{out}^2\right), \tag{3.19}$$

$$\hat{\alpha}^{in}\left(v_{i,j_1}^{\hat{\mathcal{H}}_1}, -\right) := \beta^{in}\left(\delta_{in}(v_{i,j_1}^{\hat{\mathcal{H}}_1}), \xi, \hat{\sigma}_{in}^1\right), \tag{3.20}$$

$$\hat{\alpha}^{in}\left(-, v_{i,j_2}^{\hat{\mathcal{H}}_2}\right) := \beta^{in}\left(\xi, \delta_{in}(v_{i,j_2}^{\hat{\mathcal{H}}_2}), \hat{\sigma}_{in}^1\right), \tag{3.21}$$

$$\hat{\alpha}^{in}\left(v_{i,j_1}^{\hat{\mathcal{H}}_1}, v_{i,j_2}^{\hat{\mathcal{H}}_2}\right) := \beta^{in}\left(\delta_{in}(v_{i,j_1}^{\hat{\mathcal{H}}_1}), \delta_{out}(v_{i,j_2}^{\hat{\mathcal{H}}_2}), \hat{\sigma}_{in}^2\right), \tag{3.22}$$

Finally, if we set

$$\begin{aligned}\gamma^{out}(i) &:= 1- \\ &\frac{1}{\sigma_i^1+\sigma_i^2}\cdot\left\{\sum_{j=1}^{\sigma_i^1}\hat{\alpha}^{out}\left(v_{i,j}^{\hat{\mathcal{H}}_1},\ \text{align}\left(v_{i,j}^{\hat{\mathcal{H}}_1}\right)\right)\right\} \\ &+\frac{1}{\sigma_i^1+\sigma_i^2}\cdot\left\{\sum_{j=1}^{\sigma_i^2}\hat{\alpha}^{out}\left(v_{i,j}^{\hat{\mathcal{H}}_2},\ \text{align}\left(v_{i,j}^{\hat{\mathcal{H}}_2}\right)\right)\right\}\,,\end{aligned} \tag{3.23}$$

and

$$\begin{aligned}\gamma^{in}(i) &:= 1- \\ &\frac{1}{\sigma_i^1+\sigma_i^2}\cdot\left\{\sum_{j=1}^{\sigma_i^1}\hat{\alpha}^{in}\left(v_{i,j}^{\hat{\mathcal{H}}_1},\ \text{align}\left(v_{i,j}^{\hat{\mathcal{H}}_1}\right)\right)\right\} \\ &+\frac{1}{\sigma_i^1+\sigma_i^2}\cdot\left\{\sum_{j=1}^{\sigma_i^2}\hat{\alpha}^{in}\left(v_{i,j}^{\hat{\mathcal{H}}_2},\ \text{align}\left(v_{i,j}^{\hat{\mathcal{H}}_2}\right)\right)\right\}\,,\end{aligned} \tag{3.24}$$

we obtain measures which indicate how similar the out-degree and in-degree alignments of two sequences on a level i are. It holds by construction $\gamma^{out}(i)$, $\gamma^{in}(i) \in [0,1]$. In the following we construct a family of graph similarity measures.

Generally, similarity measures have wide applications in several areas of science, for example in cluster analysis [3], [9], sociology [22] and psychology [11], [30]. Now, our similarity measures will be based on the structural description of our objects. We define similarity measures which are similar to the definition of Batagelj [5].

Definition 3.2. Let U be a set of units and a mapping $\phi\colon U \times U \longrightarrow [0,1]$. ϕ is called a similarity measure if

$$\phi(u,v) = \phi(v,u), \forall\, u,v \in U \quad (\text{Symmetry})\,, \tag{3.25}$$

and either

$$\phi(u,u) \leqslant \phi(u,v), \forall\, u,v \in U \quad (\text{Forward})\,, \tag{3.26}$$

or

$$\phi(u,u) \geqslant \phi(u,v), \forall\, u,v \in U \quad (\text{Backward})\,. \tag{3.27}$$

Finally, we prove the key result for measuring the structural similarity of unlabeled, hierarchical, and directed graphs.

Theorem 3.1. *Let* $\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2$, $0 \leqslant i \leqslant \rho$, $\rho := \max(h_1, h_2)$.

$$d_1(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2) := \frac{\sum_{i=0}^{\rho} \lambda_i \cdot \gamma^{fin}(i)}{\sum_{i=0}^{\rho} \lambda_i}, \tag{3.28}$$

$$d_2(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2) := \frac{\sum_{i=0}^{\rho} \gamma^{fin}(i)}{\rho + 1}, \tag{3.29}$$

$$d_3(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2) := \frac{\prod_{i=0}^{\rho} \gamma^{fin}(i)}{d_2(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2)}, \tag{3.30}$$

are a family $\left(d_i(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2)\right)_{1 \leqslant i \leqslant 3}$ *of Backward similarity measures, where* $\gamma^{fin}(i)$ *is defined as*

$$\gamma^{fin}(i) := \zeta \cdot \gamma^{out}(i) + (1 - \zeta) \cdot \gamma^{in}(i). \tag{3.31}$$

It holds $\left(d_i(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2)\right)_{1 \leqslant i \leqslant 3} \in [0, 1]$.

Proof. First, we consider the function

$$\gamma^{out}(i) := 1 - \frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^1} \hat{\alpha}^{out}\left(v_{i,j}^{\mathcal{H}_1},\ \text{align}\left(v_{i,j}^{\mathcal{H}_1}\right)\right) \right\} + \frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^2} \hat{\alpha}^{out}\left(v_{i,j}^{\hat{\mathcal{H}}_2},\ \text{align}\left(v_{i,j}^{\mathcal{H}_2}\right)\right) \right\}.$$

With Equation (3.16), it follows that we have to distinguish the three cases for the function $\hat{\alpha}^{out}$: $\hat{\alpha}^{out}\left(v_{i,j_1}^{\hat{\mathcal{H}}_1}, -\right)$, $\hat{\alpha}^{out}\left(-, v_{i,j_2}^{\hat{\mathcal{H}}_2}\right)$, $\hat{\alpha}^{out}\left(v_{i,j_1}^{\hat{\mathcal{H}}_1}, v_{i,j_2}^{\hat{\mathcal{H}}_2}\right)$. We infer by the Equations (3.17), (3.18), (3.19) that

$$\hat{\alpha}^{out}\left(v_{i,j_1}^{\hat{\mathcal{H}}_1},\ \text{align}\left(v_{i,j_1}^{\hat{\mathcal{H}}_1}\right)\right) \leqslant 1 \quad \text{and} \quad \hat{\alpha}^{out}\left(v_{i,j_2}^{\hat{\mathcal{H}}_2},\ \text{align}\left(v_{i,j_2}^{\hat{\mathcal{H}}_2}\right)\right) \leqslant 1.$$

Hence, by Definition (3.23) we obtain $\gamma^{out}(i) \leqslant 1$. The Proof $\gamma^{in}(i) \leqslant 1$ is identical. Since

$$\gamma^{fin}(i) \leqslant \zeta + (1 - \zeta) = 1,$$

we obtain

$$d_1(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2) \leqslant \frac{\sum_{i=0}^{\rho} \lambda_i}{\sum_{i=0}^{\rho} \lambda_i} = 1. \tag{3.32}$$

To prove that $d_1(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2)$ is symmetric, we see that

$$\gamma^{out}(i) := 1-$$
$$\frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^1} \hat{\alpha}^{out}\left(v_{i,j}^{\hat{\mathcal{H}}_1},\ \text{align}\ \left(v_{i,j}^{\hat{\mathcal{H}}_1} \right) \right) \right\}$$
$$+ \frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^2} \hat{\alpha}^{out}\left(v_{i,j}^{\hat{\mathcal{H}}_2},\ \text{align}\ \left(v_{i,j}^{\hat{\mathcal{H}}_2} \right) \right) \right\}$$
$$= 1-$$
$$\frac{1}{\sigma_i^2 + \sigma_i^1} \cdot \left\{ \sum_{j=1}^{\sigma_i^1} \hat{\alpha}^{out}\left(v_{i,j}^{\hat{\mathcal{H}}_2},\ \text{align}\ \left(v_{i,j}^{\hat{\mathcal{H}}_2} \right) \right) \right\}$$
$$+ \frac{1}{\sigma_i^2 + \sigma_i^1} \cdot \left\{ \sum_{j=1}^{\sigma_i^2} \hat{\alpha}^{out}\left(v_{i,j}^{\hat{\mathcal{H}}_1},\ \text{align}\ \left(v_{i,j}^{\hat{\mathcal{H}}_1} \right) \right) \right\} ,$$

and

$$\gamma^{in}(i) := 1-$$
$$\frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^1} \hat{\alpha}^{in}\left(v_{i,j}^{\hat{\mathcal{H}}_1},\ \text{align}\ \left(v_{i,j}^{\hat{\mathcal{H}}_1} \right) \right) \right\}$$
$$+ \frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^2} \hat{\alpha}^{in}\left(v_{i,j}^{\hat{\mathcal{H}}_2},\ \text{align}\ \left(v_{i,j}^{\hat{\mathcal{H}}_2} \right) \right) \right\}$$
$$= 1-$$
$$\frac{1}{\sigma_i^2 + \sigma_i^1} \cdot \left\{ \sum_{j=1}^{\sigma_i^1} \hat{\alpha}^{in}\left(v_{i,j}^{\hat{\mathcal{H}}_2},\ \text{align}\ \left(v_{i,j}^{\hat{\mathcal{H}}_2} \right) \right) \right\}$$
$$+ \frac{1}{\sigma_i^2 + \sigma_i^1} \cdot \left\{ \sum_{j=1}^{\sigma_i^2} \hat{\alpha}^{in}\left(v_{i,j}^{\hat{\mathcal{H}}_1},\ \text{align}\ \left(v_{i,j}^{\hat{\mathcal{H}}_1} \right) \right) \right\} .$$

Therefore, we conclude with Equation (3.28) that

$$d_1(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2) = d_1(\hat{\mathcal{H}}_2, \hat{\mathcal{H}}_1) .$$

To finalize the proof for the Similarity Measure (3.28), we have to show that

$$d_1(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_1) \geqslant d_1(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2) .$$

If $\hat{\mathcal{H}}_1 = \hat{\mathcal{H}}_2$, then $\gamma^{out}(i) = 1$, $\gamma^{in}(i) = 1$ and $\gamma^{fin}(i) = 1$. Therefore we infer from Equation (3.28) that $d_1(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_1) = 1$ and see

$$d_1(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_1) = 1 \geqslant \frac{\sum_{i=0}^{\rho} \lambda_i \cdot \gamma^{fin}(i)}{\sum_{i=0}^{\rho} \lambda_i} = d_1(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2),$$

the Backward property. In the case of Similarity Measure (3.29) we have nothing to prove, because if we set $1 = \lambda_0 = \lambda_1 = \cdots = \lambda_\rho$ in Equation (3.28), we obtain Equation (3.29). To prove the assertion of the theorem for Equation (3.30), we consider the well-known inequality [15]

$$(p_1 \cdot p_2 \cdots p_n)^{\frac{1}{n}} \leqslant \frac{p_1 + p_2 + \cdots + p_n}{n}, \; p_i > 0, \; 1 \leqslant i \leqslant n. \tag{3.33}$$

Since $\gamma^{fin}(i) \leqslant 1$, we can apply Inequality (3.33). We obtain

$$\begin{aligned}\gamma^{fin}(0) \cdot \gamma^{fin}(1) \cdots \gamma^{fin}(\rho) &\leqslant \left[\gamma^{fin}(0) \cdot \gamma^{fin}(1) \cdots \gamma^{fin}(\rho)\right]^{\frac{1}{\rho+1}} \\ &\leqslant \frac{\gamma^{fin}(0) + \gamma^{fin}(1) + \cdots + \gamma^{fin}(\rho)}{\rho + 1}\end{aligned}$$

and especially

$$1 \geqslant \frac{\gamma^{fin}(0) \cdot \gamma^{fin}(1) \cdots \gamma^{fin}(\rho)}{\frac{\gamma^{fin}(0) + \gamma^{fin}(1) + \cdots + \gamma^{fin}(\rho)}{\rho+1}}. \tag{3.34}$$

The symmetry condition is clear, because the expression in the denominator of Equation (3.30) is a special case of d_1. The Backward condition follows immediately from Inequality (3.34),

$$1 = d_3(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_1) \geqslant \frac{\prod_{i=0}^{\rho} \gamma^{fin}(i)}{d_2(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2)}.$$

$\square$

4. Experimental results

In this chapter, we report on the results obtained by testing the algorithm on a graph corpus C_G representing 464 conference/workshop websites from computer science and mathematics. The graph corpus has already been used for a study [8], [23] in *hypertext categorization*.

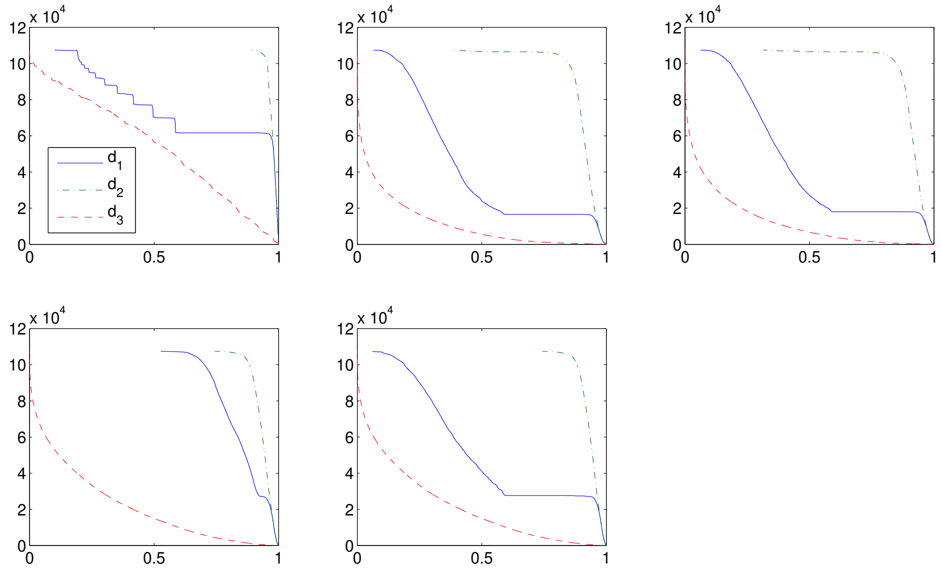

FIGURE 4.1. Distributions of the ranked similarity values for d_1, d_2 and d_3 regarding all data classes of Definition 4.1. The X-axis denotes the similarity values of $\left(d_i(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2)\right)_{0 \leqslant i \leqslant 3} \in [0, 1]$. The Y-axis denotes the number of graph pairs.

In Figure (4.1) we depict the distributions of the ranked similarity values of Theorem (3.1) by using the introduced different parameter spectra. Therefore, we express

DEFINITION 4.1. In terms of C_G we define data classes $D_1 - D_5$ which are manifested by the following parameter spectra:

(1) $D_1: \zeta = 1.0$ (Solely alignments of Kernel-edges); If $\gamma^{fin}(i) < 0.5$, set $\lambda_i = 100$, else $\lambda_i = 1$; parameter settings:

$$\sigma^1_{out} = 1.0, \sigma^2_{out} = 2.0, \sigma^1_{in} = 1.0, \sigma^2_{in} = 2.0, \hat{\sigma}^1_{out} = 3.0, \hat{\sigma}^2_{out} = 5.0,$$
$$\hat{\sigma}^1_{in} = 3.0, \hat{\sigma}^2_{in} = 5.0.$$

(2) $D_2: \zeta = 0.3$; If $\gamma^{fin}(i) < 0.5$, set $\lambda_i = 100$, else $\lambda_i = 1$; parameter settings:

$$\sigma^1_{out} = 1.0, \sigma^2_{out} = 1.0, \sigma^1_{in} = 1.0, \sigma^2_{in} = 1.0, \hat{\sigma}^1_{out} = 1.0, \hat{\sigma}^2_{out} = 5.0,$$
$$\hat{\sigma}^1_{in} = 1.0, \hat{\sigma}^2_{in} = 5.0.$$

(3) $D_3: \zeta = 0.5$; If $\gamma^{fin}(i) < 0.5$, set $\lambda_i = 100$, else $\lambda_i = 1$; parameter settings:

$$\sigma^1_{out} = 1.0, \sigma^2_{out} = 1.0, \sigma^1_{in} = 1.0, \sigma^2_{in} = 1.0, \hat{\sigma}^1_{out} = 1.0, \hat{\sigma}^2_{out} = 5.0,$$
$$\hat{\sigma}^1_{in} = 1.0, \hat{\sigma}^2_{in} = 5.0.$$

(4) D_4: $\zeta = 0.5$; If $\gamma^{fin}(i) < 0.5$, set $\lambda_i = 64$, else $\lambda_i = 16$; parameter settings:

$\sigma^1_{out} = 1.0, \sigma^2_{out} = 2.0, \sigma^1_{in} = 1.0, \sigma^2_{in} = 2.0, \hat{\sigma}^1_{out} = 3.0, \hat{\sigma}^2_{out} = 5.0,$
$\hat{\sigma}^1_{in} = 3.0, \hat{\sigma}^2_{in} = 5.0$.

(5) D_5: $\zeta = 0.5$; If $\gamma^{fin}(i) < 0.5$, set $\lambda_i = 100$, else $\lambda_i = 1$; parameter settings:

$\sigma^1_{out} = 1.0, \sigma^2_{out} = 1.0, \sigma^1_{in} = 1.0, \sigma^2_{in} = 1.0, \hat{\sigma}^1_{out} = 3.0, \hat{\sigma}^2_{out} = 5.0,$
$\hat{\sigma}^1_{in} = 3.0, \hat{\sigma}^2_{in} = 5.0$.

We regard the graph corpus C_G (see Table (4.3)) as homogeneous in the sense that it covers, structurally, the whole interval. Now, Figure (4.1) shows the ranked similarity values regarding the data classes $D_1 - D_5$. Then, we observe from the plots of Figure[1] (4.1) that the similarity measures d_1 and d_2 are not suitable. They do not cover the whole interval of the possible similarity values. If we set, in d_1, different values for λ_i, $0 \leqslant i \leqslant \rho := \max(h_1, h_2)$, the measure d_1 is very sensitive to structural variances of the graphs. The "visible steps" and especially the horizontal segments in the plots of d_1 induce clusters, which contain the same number of graph pairs with a certain similarity value on the X-axis. In contrast to this, d_3 covers the interval of similarity values very well. It exploits the whole interval. Therefore, in the following we use for further experiments only d_3.

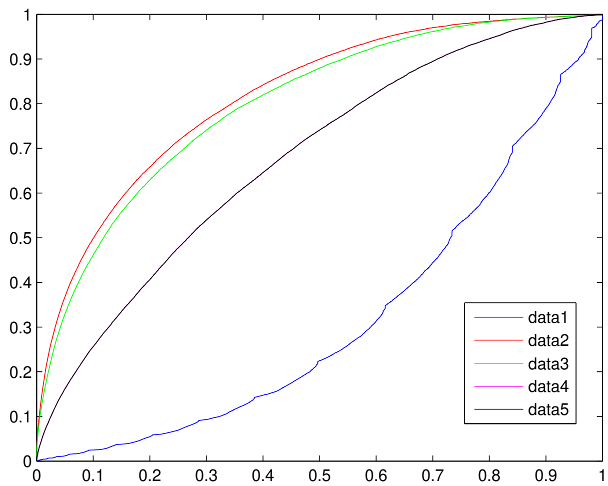

FIGURE 4.2. The X-axis corresponds to the values of $d_3 \in [0, 1]$ and the Y-axis represents the cumulative similarity distributions for D_1 - D_5.

[1]The first upper plot belongs to Item (1) of Definition (4.1), the second upper plot belongs to Item (2) of Definition (4.1) etc.

Furthermore, we compute the cumulative similarity distribution (see Figure (4.2)), of C_G based on Definition (4.1). In general, the computation of the cumulative similarity distribution of a graph corpus opens new perspectives: As a preprocessing step for the structural analysis of graph corpora we can decide how structurally different the graphs are.

In order to discuss the cumulative similarity distribution of C_G (see Figure (4.2)) we note that the data classes D_1 - D_5 are manifested by the same corpus C_G. We obtain a certain data class only by varying the parameters mentioned in Definition (4.1). Now, by varying the parameters we find the parameter set

$$(\zeta, \sigma^1_{out}, \sigma^2_{out}, \sigma^1_{in}, \sigma^2_{in}, \hat{\sigma}^1_{out}, \hat{\sigma}^2_{out}, \hat{\sigma}^1_{in}, \hat{\sigma}^2_{in}),$$

which captures enough structural information during the similarity measuring process. In the following we note that the plot of class D_1 differs in principle from the plots of data classes D_2 - D_5. We recognize that, e.g., 20% of the graph based conference websites have already the similarity value $d \leqslant 0.5$. Unlike 90% of the conference websites in D_2 have the similarity value $d \leqslant 0.5$. In summary, we conclude from Figure (4.2) that the similarity values of the conference websites in D_1 were significantly higher compared to the conference websites of data classes D_2 - D_5. This is plausible, because the conference websites in D_1 are treated solely as rooted trees without Across-edges, Up-edges and Down-edges. Hence, the main part of the conference websites of D_1 is significantly less structurally different than the websites of the remaining data classes. In terms of D_2 - D_5 the situation is inverted: In consideration of all types of conference websites the main part of the graph-based hypertext structures is structurally dissimilar on the basis of d_3. The plot of D_4 equals the plot of D_5.

5. Summary and conclusion

In this paper, we introduced a new method to measure the structural similarity of a special class of directed graphs. Thereby, the graphs are unlabeled, hierarchical, and directed and we applied our algorithm on a graph corpus C_G [23] of graph-based hypertext structures. The main contributions of the paper are:

(1) Starting from observations on degree sequences of directed graphs, we developed a new method for measuring the structural similarity of graphs. The main idea of our new similarity measures are based on the derivation of property strings (out-degree and in-degree sequences on each graph level) for each hierarchical and directed graph and then to align the property strings representing our graphs by a dynamic programming technique. From the resulting alignment we obtain a value of the scoring

function, which is minimized during the alignment process. The similarity of two hierarchical and directed graphs will be expressed by a cumulation of local similarity functions $\gamma^{out}(i), \gamma^{in}(i)$ which weighs two types of alignments: out-degree and in-degree alignments on a graph level. These alignments have both global and local significance. On the one hand, the sequence alignments will be implemented in a global sense, to compute the optimal alignment between the sequences S_1 and S_2. On the other hand, the alignments will be evaluated on the levels of the graphs by the function $\gamma^{fin}(i)$. Since the functions $\gamma^{out}(i), \gamma^{in}(i)$ and $\gamma^{fin}(i)$ are basically decoupled from the similarity measures of Theorem (3.1), we now can define new measured values d_i, where they are adapted to a new graph similarity problem. Therefore, we obtain a family of graph similarity measures. From the overall results mentioned above we see that our family of similarity measures $\left(d_i(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2)\right)_{1 \leqslant i \leqslant 3}$ is also different from graph similarity measures, which are based on isomorphic relations, e.g., [20], [28], [29], [32].

(2) In Section (4) we evaluated the measures $\left(d_i(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2)\right)_{1 \leqslant i \leqslant 3}$ on a graph corpus C_G, consisting of 464 graphs representing web-based hypertext structures. The results of the evaluation are reflected in Figure (4.1), (4.2). Especially, we showed that the cumulative similarity distribution provides useful information about our graph corpus C_G. With Definition (4.1) we answered the important and interesting question how structurally different the graphs of C_G are. Because our measures are parametric similarity measures depending on

$$(\zeta, \sigma^1_{out}, \sigma^2_{out}, \sigma^1_{in}, \sigma^2_{in}, \hat{\sigma}^1_{out}, \hat{\sigma}^2_{out}, \hat{\sigma}^1_{in}, \hat{\sigma}^2_{in}),$$

we were able to emphasize different structure types of our graph trees during the alignment process. For example, by setting $\zeta = 1$ we consider an unlabeled hypertext structure as a directed rooted tree. That means that we align only the out-degree property strings induced by edges from the underlying directed rooted tree. If we set $\zeta = 0$, we align the property strings induced by in-degree sequences only. In most of the cases we used $\zeta = \frac{1}{2}$, which weighs in- and out-degree sequences equally, $\gamma^{fin} = \frac{\gamma^{out}}{2} + \frac{\gamma^{in}}{2}$.

In the future we concentrate on a generalization of our new method for measuring the structural similarity of arbitrary graphs.

Acknowledgement. We would like to thank Frank Emmert–Streib and Jürgen Kilian for fruitful discussions.

REFERENCES

[1] ALTSCHUL, S. F.—GISH, W.—MILLER, W.—MYERS, E. W.—LIPMAN, D. J.: *Basic local alignment search tool*, J. Molecular Biology **125** (1991), 403–410.

[2] ALTSCHUL, S. F.—MADDEN, T. L.—MILLER, W.—SCHAFFER, A. A.—ZHANG, J.—ZHANG, Z.—LIPMAN, D. J.: *Gapped BLAST and PSI–BLAST: a new generation of protein database search programs*, Nucleic Acids Res. **25** (1997), 3389–3402.

[3] ANDERBERG, R.: *Cluster Analysis for Applications*, in: Probab. Mat. Statist., Academic Press, New York, Vol. 19, 1973.

[4] BANG-JENSEN, J.—GUTIN, G.: *Digraphs. Theory, Algorithms and Applications*, Springer-Verlag, Berlin, 2000.

[5] BATAGELJ, V.: *Similarity measures between structured objects, MATH/CHEM/COMP*, in: Proceedings of an International Course and Conference on the Interfaces between Mathematics, Chemistry and Computer Sciences, Dubrovnik, Yugoslavia, 1988.

[6] BELLMAN, R.: *Dynamic Programming*, Princeton University Press, Princeton, New Yersey, 1957.

[7] BUNKE, H.: *Recent developments in graph matching*, in: Proceedings of the 15th Int. Conf. on Pattern Recognition, Barcelona, Spain, Vol. 2, 2000, pp. 117–124.

[8] DEHMER, M.—MEHLER, A.—GLEIM, R.: *Aspekte der Kategorisierung von Webseiten*, in: Jahrestagung der Gesellschaft für Informatik 2004, GI-Edition, pp. 39–43 (Peter Dadam, Manfred Reigerts, eds.), Springer Verlag, Heidelberg, http//www.springeronline.com, 2004.

[9] EVERITT, B. S.—LANDAU, S.—LEESE, M.: *Cluster Analysis*, 4th edition, Arnold Publishers, London, 2001.

[10] GENERT, D.: *Measuring the similarity of complex structures by means of graph grammars*, Bull. EATCS **7** (1979), 3–9.

[11] GREGSON, A. M. R.: *Psychometrics of Similarity*, Academic Press, New York, 1975.

[12] HAKIMI, S. L.: *On the realizability of a set of integers as degrees of a graph*, J. SIAM Appl. Math. **10** (1962), 496–506.

[13] HAKIMI, S. L.: *On the degrees of the vertices of a directed graph*, J. Franklin Inst. **279** (1965), 290–308.

[14] HAVEL, V.: *A remark on the existence of finite graphs*, Čas. Pěst. Mat. **80** (1955), 477–480. (In Czech)

[15] HEUSER, L.: *Lehrbuch der Analysis. Teil 1*, Teubner Verlag, Stuttgart, 1991.

[16] KADEN, F.: *Graphmetriken und Distanzgraphen*, ZKI-Informationen, Akad. Wiss. DDR **2** (1982), 1–63.

[17] KADEN, F.: *Graph metrics and distance-graphs*, in: Graphs and other Combinatorial Topics (M. Fiedler, ed.), Teubner Texte zur Math., Leipzig, Vol. 59, 1983, pp. 145–158.

[18] KADEN, F.: *Halbgeordnete Graphmengen und Graphmetriken*, in: Graphs, Hypergraphs and Applications (H. Sachs, ed.), Teubner Texte zur Math., Leipzig, Vol. 73, 1985, pp. 92–95.

[19] KILIAN, J.—HOOS, H. H.: *MusicBLAST–gapped sequence alignment for MIR*, in: Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR), 2004 (to appear).

[20] KOCH, I.—LENGAUER, T.—WANKE, E.: *An algorithm for finding maximal common subtopologies in a set of protein structures*, J. Comput. Biology **3** (1996), 289–306.

[21] LEVENSTEIN, V. I.: *Binary codes capable of correcting deletions, insertions and reversals*, Soviet Phys. Dokl. **10** (1966), 707–710.

[22] LIEBETRAU, A. M.: *Measures of Association*, Sage University Paper Series on Quantitative Applications in the Social Sciences–Sage Publications, Beverley Hills, 1983.

[23] MEHLER, A.—DEHMER, M.—GLEIM, R.: *Towards logical hypertext structure. A graph-theoretic perspective*, in: Proc. of I2CS '04, Lecture Notes in Comput. Sci. Vol. 3473, Springer-Verlag, Berlin, 2006, 136–150.

[24] NAGL, M.: *Graph-Grammatiken*, Vieweg, Wiesbaden, 1979.

[25] READ, R. C.—CORNEIL, D. G.: *The graph isomorphism disease*, J. Graph Theory **1** (1977), 339–363.

[26] SHAPIRO, L. G.: *Organization of relational models*, in: Proc. Intern. Conf. on Pattern Recognition, München, 1982, pp. 360–365.

[27] SKVORTSOVA, M. I.—BASKIN, I. I.—STANKEVICH, I. V.—PALYULIN, V. A.—ZERIROW, N. S.: *Molecular similarity in structure-property relationship studies. Analytical description of the complete set of graph similarity measures*, International Symposium CACR '96, 1996, pp. 642–646.

[28] SOBIK, F.: *Graphmetriken und Klassifikation strukturierter Objekte*, ZKI-Inf., Akad. Wiss. DDR **2** (1982), 63–122.

[29] SOBIK, F.: *Modellierung von Vergleichsprozessen auf der Grundlage von Ähnlichkeitsmaßen für Graphen*, ZKI-Inf., Akad. Wiss. DDR **4** (1986), 104–144.

[30] TVERSKY, A.: *Features of similarity*, Psychological Review **84** (1977), 327–352.

[31] ULLMAN, J. R.: *An algorithm for subgraph isomorphism*, J. ACM **23** (1976), 31–42.

[32] ZELINKA, B.: *On a certain distance between isomorphism classes of graphs*, Čas. Pěst. Mat. **100** (1975), 371–373.

[33] ZHANG, K.—STATMAN, R.—SHASHA, D.: *On the editing distance between unordered labeled trees*, Inf. Process. Lett. **42** (1992), 133–139.

[34] ZHANG, K.—WANG, J.—JASON, T. L.—SHASHA, D.: *On the editing distance between undirected acyclic graphs*, Int. J. Found. Comput. Sci. **7** (1996), 43–57.

Received August 26, 2004

Matthias Dehmer
Technische Universität Darmstadt
G–64289 Darmstadt
GERMANY
E-mail: dehmer@tk.informatik.tu-darmstadt.de

Institute of Discrete Mathematics and Geometry
Vienna University of Technology
Wiedner Hauptstrasse 8–10
A–1040 Vienna
AUSTRIA
E-mail: mattias.dehmer@univie.ac.at

Alexander Mehler
Universität Bielefeld
G–33501 Bielefeld
GERMANY
E-mail: Alexander.Mehler@uni-bielefeld.de