Faster Algorithms for Computing Distances between One-Dimensional Point Sets

Justin Colannino Godfried Toussaint

School of Computer Science
McGill University
Montréal, Québec, Canada

July 8, 2005

Abstract

Let S and T be two finite sets of points on the real line with $|S| + |T| = n$ and $|S| > |T|$. We consider two distance measures between S and T that have applications in music information retrieval and computational biology: the *surjection* distance and the *link* distance. The former is called the *restriction scaffold assignment* problem in computational biology, and assigns each point of S to a point of T such that the sum of all the assignment costs is minimized, with the constraint that every element of T must be assigned at least one element of S. The cost of assigning an element s_i of S to an element t_j of T is $|s_i - t_j|$, i.e., the distance between s_i and t_j. In 2003 Ben-Dor, Karp, Schwikowski and Shamir [2] published an $O(n \log n)$ time algorithm for this problem. Here we provide a counter-example to their algorithm and present a new algorithm that runs in $O(n^2)$ time, improving the best previous complexity of $O(n^3)$. The link distance allows multiple assignments from both S to T and T to S. We give an $O(n^2)$ algorithm for the link distance which also improves the best previous complexity of $O(n^3)$.

1 Introduction

In the context of measuring the similarity of musical rhythms with the goal of performing a phylogenetic analysis of rhythms, Toussaint [9] proposed the use of the swap-distance. A comparison of this distance measure with other rhythm dissimilarity measures shows it to be superior in several respects [8]. In [9] and [8] the rhythms are represented as binary sequences, where a '1' denotes the onset of a note and a '0' denotes a silence. Furthermore, all the rhythms compared have the same length, n bits, with the same number k of 1's (or onsets). A swap operation on a string consists of interchanging two adjacent elements in that string. The swap distance between two binary strings is defined as the minimum number of swap operations needed to transform one string into the other. In this restricted version of the problem, computing the swap-distance is very simple. For all i, the ith '1' of one string must move to the position of the ith '1' of the second string. Therefore the number of swaps needed for one such operation is the difference between their indices (which may be viewed as integer x-coordinates. The swap distance is the sum of all the k

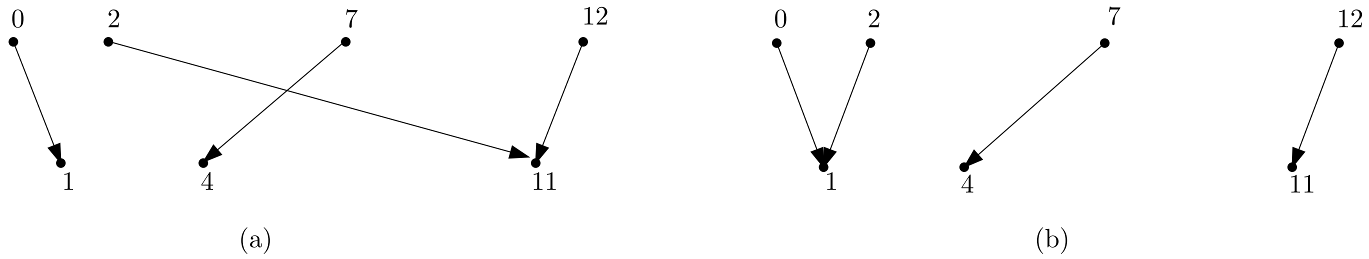

Figure 1: (a) A surjection between the two one dimensional sets $S = \{0, 2, 7, 12\}$ and $T = \{1, 4, 11\}$. (b) A minimal surjection between S and T.

differences, which may be trivially computed in $O(n)$ time given the two binary sequences as input. Note that actually performing the swaps may require $\Omega(n^2)$ swaps.

In a more general setting, the two rhythms have different values of k, and the algorithm described in the preceding will not work. In order to capture the attributes of the swap distance measure on two binary strings S and T where S has more elements than T, Toussaint proposed the *directed* swap distance, which was first applied to the phylogenetic analysis of Flamenco rhythms by Dìaz-Báñez et al. [4]. The directed swap distance is defined as the minimum number of swaps required to move every element of S to the index of an element of T, with the restriction that every element of T must have at least one element of S moved to its index.

The directed swap distance may be viewed as a version of the linear *assignment* problem [6], where the cost of an assignment between an element i of S and an element j of T is the distance between i and j. Furthermore, we may consider the more general input consisting of two sets of numbers on the real line rather than binary sequences. Here the real numbers play the role of the indices of the 1's in the binary sequence. In this setting, if both sets have equal cardinalities, the simple algorithm described in the preceding for binary sequences may still be used after sorting the sets, thus yielding an $O(n \log n)$ time algorithm.

An alternate way of viewing the directed swap distance is as a surjection, ψ, between two sets of elements S (the source) and T (the target) on the interval $(0, X)$ where $|S| \geq |T|$. This mapping is bound by the constraint that each element of T must have at least one element of S mapped to it. More formally the directed swap distance may be expressed as a surjection as follows:

$$\min_{\psi} \sum_{s \in S} |s - \psi(s)|. \tag{1}$$

Any surjection that satisfies the preceding equation we call a minimal surjection. Figure 1 depicts two different surjections between two sets of points on the line, one of which is minimal. Note that all the points actually have zero y-coordinates; they are shown in this way merely for the purpose of clarity.

In 1979 the philosopher Graham Oddie proposed surjections to measure the distance between two theories expressed in a logical language [7]. In 1997 Eiter and Mannila extended this idea by expressing theories as models, and thus as points in a metric space [5]. This gave them a new distance measure in a metric space which they called the surjection distance.

The surjection distance between two sets S and T is defined as follows:

$$\min_{\psi} \sum_{s \in S} \delta(s, \psi(s)), \tag{2}$$

where δ is a distance metric, and ψ is a surjection between S and T. They also proposed an algorithm for computing the surjection distance in $O(n^3)$ time, where $n = |S|$, by reducing the problem to finding a minimum-weight perfect matching in an appropriate graph.

In 2003, Ben-Dor et al. [2], in the context of the shotgun sequencing problem in computational biology, introduced a modified assignment problem similar to the directed swap problem where the points are real numbers on the line rather than bits in a binary string: the *restriction scaffold assignment* problem. They also presented an $O(n \log n)$ algorithm to compute this assignment problem. Their result relies heavily on a result of Karp and Li [6] which provides a linear time algorithm (after sorting) for computing the *one-to-one* assignment problem in the special case where all the points lie on a line. In the one-to-one assignment problem between S and T some elements of S remain unassigned.

In this note we give a counter-example to the algorithm of Ben-Dor et al., [2] for computing the restriction scaffold assignment problem, and show that the problem may be solved in $O(n^2)$ time.

2 The Algorithm of Eiter and Mannila

Since our $O(n^2)$ algorithm is inspired by, and based on, some results of Eiter and Mannila [5], we describe in this section their main results for the sake of clarity and completion. The proofs of these results may be found in [5]. Their algorithm is based on a reduction to finding a minimum weight perfect matching in a suitable bipartite graph. The reduction makes use of the following observation.

Lemma 1 *Let ψ be a minimal surjection from S to T. Then for any $s_i \neq s_j$ if $\psi(s_i) = \psi(s_j)$ the distance from s_i to $\psi(s_i)$ is not more than the distance from s_i to any other element of T.*

Taking advantage of this result their algorithm creates $k = |S| - |T|$ auxiliary vertices. These vertices serve as dummy nodes, for which the distance to any $s \in S$ is made equal to that of the shortest distance from s to any element of T. This allows the graph to effectively simulate the situation where two or more vertices are mapped to the same element of T.

The construction is carried out as follows. For two sets, S and T in a metric space, create a complete bipartite weighted graph $G = (X \cup Y, E, w)$. For each $s_i \in S$ construct a vertex $x_i \in X$. For each $t_j \in T$ create a vertex $u_j \in U$. Finally, for $k = |S| - |T|$, create a set of nodes $V = \{v_1, ..., v_k\}$, and let $Y = (U \cup V)$. The weight function w is defined as follows:

$$w(e) = \begin{cases} e = (x_i, u_j), & \delta(s_i, t_j) \\ e = (x_i, v_j), & \min_{t \in T} \delta(s_i, t) \end{cases} \tag{3}$$

where s_i, t_j are the set elements corresponding to the vertices x_i, u_j, respectively.

Figure 2 illustrates this construction with the sets $S = \{0, 2, 7, 12\}$ and $T = \{1, 4, 11\}$. The bold lines represent the minimum perfect bipartite matching, and correspond to the

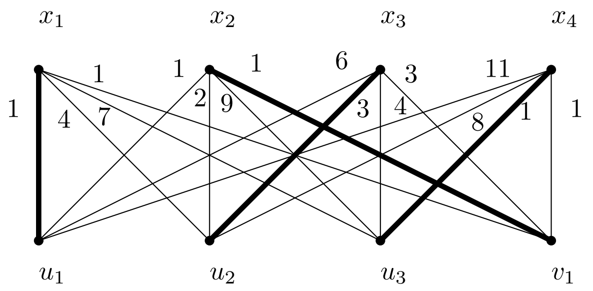

Figure 2: The bipartite graph created from $S = \{0,\ 2,\ 7,\ 12\}$, $T = \{1,\ 4,\ 11\}$. The bold lines represent the minimum perfect bipartite matching, which has a cost of 6.

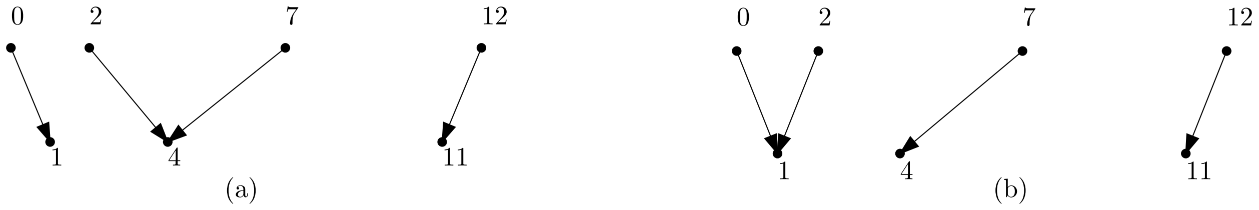

Figure 3: (a) The assignment returned by the algorithm in [2]. (b) A minimal assignment.

minimal surjection depicted in Figure 1 (b). Let $w(M)$ denote the minimum-weight perfect matching of the resulting graph, and let $c(\psi)$ denote the cost of the minimum surjection distance between S and T. Eiter and Mannila also prove the following theorem.

Theorem 2 $c(\psi) = w(M)$.

Letting n equal to $|S|$, Eiter and Mannila note that this reduction yields an $O(n^3)$ time algorithm since the construction takes $O(n^2)$ time and their matching algorithm of choice takes $O(n^3)$ time for a complete graph [3].

3 A Counter-Example to the Algorithm of Ben-Dor et al.

Ben-Dor et al. [2] presented an $O(n \log n)$ time algorithm for computing the restriction scaffold assignment problem. In this section we describe their algorithm and exhibit an example on which the algorithm does not yield the claimed optimal solution.

The key idea in their algorithm is to split the problem into two parts. First, the minimal one-to-one assignment, F, between S and T is found, leaving out a set $E \subset S$. Then each element of E is mapped to its nearest neighbor in S. The attractive feature of this procedure is that it runs in linear time provided that it uses the Karp-Li algorithm to solve the first part of the problem (the one-to-one assignment) on points that are already sorted.

A simple counter-example to this algorithm follows. Consider the two sets $S = \{0, 2, 7, 12\}$ and $T = \{1, 4, 11\}$ shown in Figure 3. The minimum one-to-one assignment function, F, is

$$\begin{aligned} F(0) &= 1 \\ F(2) &= 4 \\ F(12) &= 11 \end{aligned} \tag{4}$$

which has a total cost of 4. Adding in the distance between 7 and its nearest neighbor in T, 4, gives a total cost of $4 + 3 = 7$. However, this is not a minimum-cost surjective assignment. We have already seen a solution for these two sets in Section 2, where, using the algorithm of Eiter and Mannila, we were able to find a surjection ψ between S and T with values,

$$\begin{aligned} \psi(0) &= 1 \\ \psi(2) &= 1 \\ \psi(7) &= 4 \\ \psi(12) &= 11 \end{aligned} \tag{5}$$

which has a total cost of 6, thus providing a counter-example to the algorithm. Figure 3 illustrates this counter-example.

4 The New $O(n^2)$ Algorithm

In this section we propose a new $O(n^2)$ time algorithm for computing the the restriction scaffold assignment problem (as well as the directed swap distance). The algorithm is based on a reduction to the problem of computing the single source shortest path problem in a weighted directed acyclic graph. The construction is inspired by the graph-theoretic approach of Eiter and Mannila, and relies heavily on the following lemma, sometimes called the quadrangle inequality [1].

Lemma 3 *Let S and T be sets of points on the line. Also, let a distance function $\delta(s,t) = |s-t|$. Then for $a, b \in S$ where $a < b$, and $c, d \in T$ where $c < d$,*

$$\delta(a,c) + \delta(b,d) \leq \delta(a,d) + \delta(b,c) \tag{6}$$

Proof:

Three cases arise as pictured in Figure 4.

Case 1 : *a and b are both less than c or, symmetrically, greater than d*

Let $x = |a-b|, y = |b-c|$ and $z = |c-d|$. Then we have $\delta(a,c) + \delta(b,d) = x + 2y + z = \delta(a,d) + \delta(b,c)$.

Case 2 : *Either a or b, but not both, are in between c and d*

Let $x = |a-c|, y = |c-b|$ and $z = |b-d|$. Then, $\delta(a,c) + \delta(b,d) = x + z \leq x + 2y + z = \delta(a,d) + \delta(b,c)$.

Case 3 : *Both a and b are in between c and d*

Let $x = |c-a|, y = |a-b|$ and $z = |b-d|$. Then, $\delta(a,c) + \delta(b,d) = x + z \leq x + 2y + z = \delta(a,d) + \delta(b,c)$

□ We relate this lemma to the directed swap distance with the following corollary.

Corollary 4 *Let S and T be sets of points on the line. Then there exists a minimal surjection ψ^* from S to T such that for all $s_i < s_j$, $\psi^*(s_i) \leq \psi^*(s_j)$.*

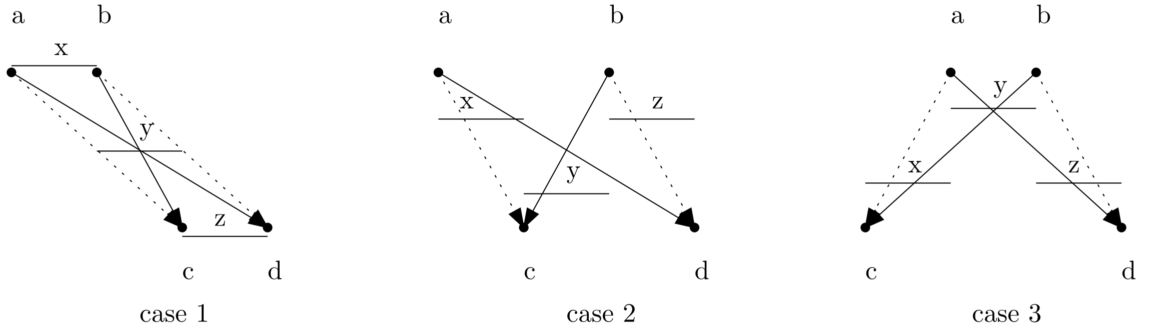

Figure 4: Illustration of Lemma 3. The dashed lines represent the smaller distance.

Proof: Let ψ be a minimal surjection function that does not satisfy the lemma. Then there must exist some $s_i < s_j$ where $\psi(s_j) < \psi(s_i)$. Let $\psi(s_j) = t_k$ and $\psi(s_i) = t_l$. Consider a new function, ψ^*, which differs from ψ only by having $\psi^*(s_i) = t_k$ and $\psi^*(s_j) = t_l$. By the quadrangle inequality lemma we have that $\sum_{s \in S} |s - \psi^*(s)| \leq \sum_{s \in S} |s - \psi(s)|$, thus proving our hypothesis. □

The first thing to note is that since all elements of S and T lie on a line, Corollary 4 implies that no crossings can occur in an optimal swap. Therefore, we know that the first element of S must be mapped to the first element of T. Furthermore, also by Corollary 4, given that s_i is mapped to t_j, we can predict that s_{i+1} is optimally mapped to either t_j or t_{j+1}. The algorithm builds a directed acyclic graph whose structure takes advantage of these two observations.

Let S, T be sets of integers on the interval $(0, X)$, where $|S| \geq |T|$, and let s_i be the ith element of S. We construct a weighted directed acyclic graph $G = (V, E)$ in the following way. For each pair i, j where $1 \leq i \leq |T|$ and $i \leq j \leq (i + |S| - |T|)$, we create a vertex, $v_{i,j}$. From each $v_{i,j}$ we add an edge to $v_{i,j+1}$ with weight $|t_i - s_{j+1}|$, and an edge to $v_{i+1,j+1}$ with weight $|t_{i+1} - s_{j+1}|$, provided $v_{i,j+1}$ and $v_{i+1,j+1}$ exist. Finally, we create a node labeled 'start' and put one edge from 'start' to $v_{1,1}$ with weight $|s_1 - t_1|$. An example of this construction is shown in Figure 5. The bold lines represent the minimum-length path through the graph, and correspond to the minimum surjection shown in Figure 1 (b).

It remains to show that the weight of the minimum-length path $w(P)$ returned from 'start' to $v_{|T|,|S|}$ is equal to the cost of the minimum surjection between S and T, $c(\psi)$. We do this by showing first that $c(\psi) \leq w(P)$, and then that $c(\psi) \geq w(P)$.

Lemma 5 $c(\psi) \leq w(P)$.

Proof: From a minimum weight path P^* we create a surjection ψ in the following way. For each vertex, $v_{i,j}$ that P^* passes through, let $\psi(s_j) = t_i$. We can verify that ψ is indeed a surjection by noting two things. First, each s_j is mapped to only one t_i, since edges from $v_{i,j}$ to $v_{i+1,j}$ do not exist in the construction. Second, every t_i receives at least one s_j, since no path exists from $v_{i,x}$ to $v_{i+2,y}$ without first passing through some $v_{i+1,z}$. This surjection ψ has a cost equal to that of P^*, since for $j = 1$ to $|S|$, $|s_j - \psi(s_j)| = w(v_{x,j-1}, v_{y,j})$. Hence, $c(\psi) \leq w(P)$. □

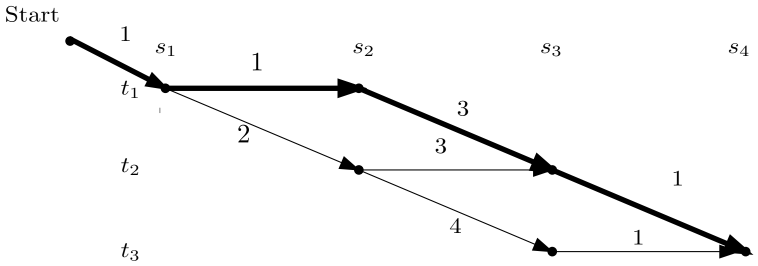

Figure 5: A directed acyclic graph created from $S = \{0, 2, 7, 12\}$, $T = \{1, 4, 11\}$. The bold lines represent the shortest path through the graph, which has weight 6.

Lemma 6 $c(\psi') \geq w(P)$:

Proof: From an optimal surjection ψ' we can create a path P using the following method. By Corollary 4, for $j = 1$ to $|S| - 1$, there are only two possibilities. When $\psi(s_j) = t_i$ and $\psi(s_j + 1) = t_i$ add the edge, $(v_{i,j}, v_{i,j+1})$ to P. Otherwise, $\psi(s_j) = t_i$ and $\psi(s_j + 1) = t_i + 1$, so we add the edge $(v_{i,j}, v_{i+1,j+1})$ to P. Finally we add the edge ('start',$v_{1,1}$).

It follows that P is a path through the constructed graph from 'start' to $v_{|S|+|T|}$. A consequence of Lemma 4 is that $\psi'(s_j) = t_i$ implies that $\psi'(s_{j+1}) = t_i$ or $\psi'(s_{j+1}) = t_{i+1}$, which means that the corresponding path P is connected in G. Also, notice that there are exactly $k = |S| - |T|$ integers, j, where $\psi'(s_j) = t_i$ and $\psi'(s_{j+1}) = t_i$. Thus, after $|S|$ steps along P we must be at node $v_{|S|,|S|-k} = v_{|S|,|T|}$. From the definition of the weight function it follows that P has a weight equal to that of ψ'. Hence, P is a valid path and $c(\psi') \geq w(P)$. □

Theorem 7 $c(\psi) = w(P)$.

Proof: The result is immediate from Lemmas 5 and 6. □

As for the complexity of this method, first let $|S| = n$ and $|T| = m$. Since in our construction each vertex has at most two edges pointing to it, we have that $|E| = O(|V|)$. Also note that $|V| = n * (n - m)$. Therefore the construction takes $O(|V| + |E|) = O(|V|)$ time, and the single source shortest path algorithm for directed acyclic graphs also takes $O(|V| + |E|) = O(|V|)$ time [3]. Therefore the entire algorithm runs in $O(|V|)$ time, which is $O(n^2)$ in the worst case, thus improving on the previous best complexity of $O(n^3)$ for this problem due to Eiter and Mannila [5].

5 The Link Distance

Eiter and Mannila also studied what they call the *link* distance.

Eiter and Mannila define a linking between two sets S and T as a relation $L \subsetneqq S \times T$ where the following two statements hold.

1. For every $s_i \in S$, there exists a $t_j \in T$ such that $(s_i, t_j) \in L$.
2. For every $t_i \in T$, there exists a $s_j \in S$ such that $(s_j, t_i) \in L$.

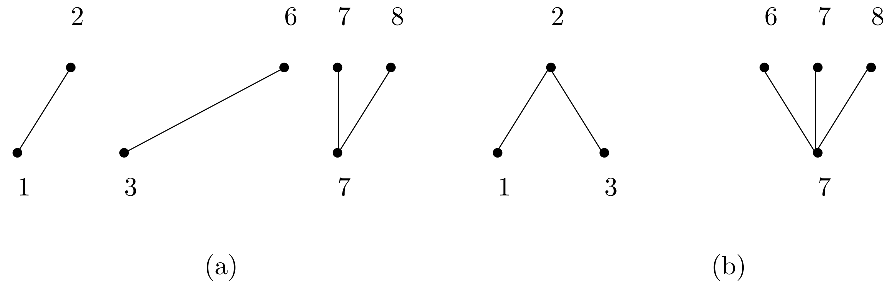

Figure 6: An example of (a) a minimal surjection and (b) a minimal linking on the same set.

The link distance is the cost of a minimal linking subject to some distance measure, δ. More formally, the link distance between two sets S and T is equal to:

$$\min_{L \in R} \sum_{(s,t) \in L} \delta(s,t)$$

Figure 6 demonstrates a case where the link distance seems to capture the relationship between two sets of points in a more natural way than the surjection distance.

5.1 The Link Distance Algorithm of Eiter and Mannila

Similar to their solution to finding the minimum surjection between two sets, Eiter and Mannila's solution to finding a minimal linking reduces the problem to finding a minimum-weight matching in a bipartite graph. The reduction depends on the following two lemmas, originally proved by Eiter and Mannila in [5].

Lemma 8 *Let S and T be sets and L a minimal linking between them. For $s \neq s'$ if $(s,t) \in L$, and $(s',t) \in L$ then the distance from s to t is less than the distance from s to any other element of T.*

Lemma 9 *Let S and T be sets and L be a minimal linking between them, then, for any relation $(s,t) \in L$, either s or t has degree* 1.

Eiter and Mannila proceed by constructing a bipartite graph on which the matching is performed. The key concept is again the creation of "dummy" nodes to handle the case when an element, x, of either set is linked to an element of the other set which has degree more than one, and hence, by Lemma 8 is x's nearest neighbor in that set. From Lemma 9 we know that any such x must have degree one. Therefore we need only create one copy for each element of $S \cup T$ with the weight of an edge between an element and its dummy equal to the distance between that element and its nearest neighbor.

To this end, the bipartite graph consists of two complete graphs. One graph represts the elements of S and T with weights according to the distances between the elements. The other is a complete zero weight "dummy" graph. An element in the dummy graph is connected to its corresponding element in the first graph with a weight equal to the distance to that element's nearest neighbor.

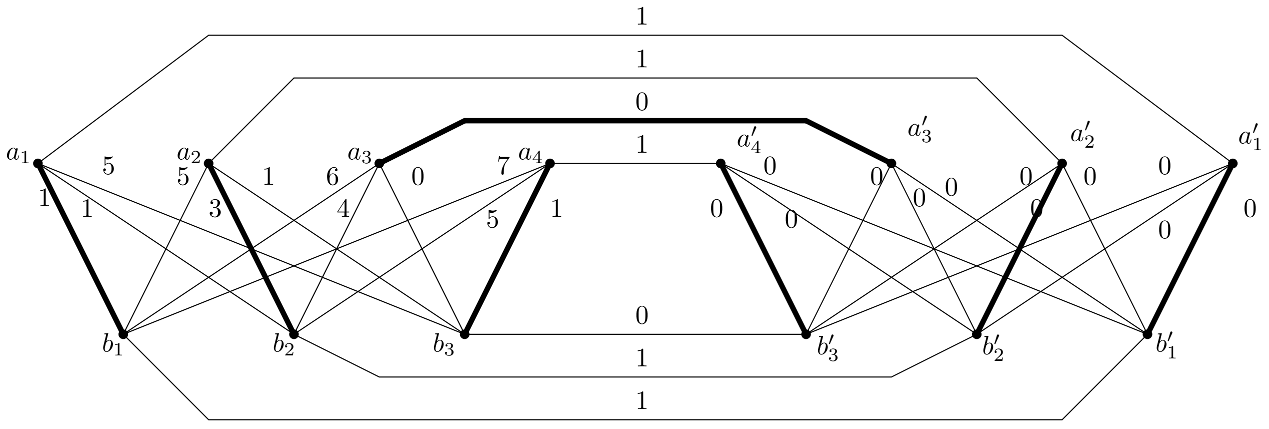

Figure 7: A graph G'' corresponding to S = {2,6,7,8}, T = {1,3,7}. The minimum weight perfect matching is given in bold.

More formally, we construct such a bipartite graph G'' as follows. For sets S and T create a graph $G = (A \cup B, E, w)$ in the following manner. For each $s \in S$ create an $a \in A$, and for each $t \in T$ create a $b \in B$. For each edge, $e = (a_i, b_j)$, set $w(e) = \delta(s_i, t_j)$. Next, create a zero weight copy of the graph $G' = (A' \cup B', E', w')$. From these two graphs we construct a graph G'' which is the union of G and G' and contains additional edges from a_i to a'_i and b_j to b'_j with weight equal to the minimum distance from s_i to an element of T and from t_j to an element of S, respectively. See Figure 7 for an example.

5.2 An Improved Algorithm for One-Dimensional Sets

While Eiter and Mannila's algorithm works for point sets in general space, our main concern is point sets in one dimension. In one dimension Lemma 3 also applies for the link distance. This leads us to the following remark, whose proof is analogous to that of Corolary 4.

Corollary 10 *Let S and T be sets of points on the line. Then there exists a minimal linking L^* from S to T such that for all $s_i < s_j$, $L^*(s_i) \leq L^*(s_j)$.*

This observation implies that in an minimal linking if we know that $(s_i, t_j) \in L$ then one of $(s_i, t_{j+1}), (s_{i+1}, t_j), (s_{i+1}, t_{j+1})$ must also be in L. Using this information it is not hard to reduce the problem, as before, to finding the shortest path through a weighted directed acyclic graph. The construction of this graph differs from the previous directed acyclic graph in that it is expanded to allow multiple linkings not only from elements of S to elements of T, but also from elements of T to elements of S.

The correctness proofs are similar to those for the surjection distance. As for the complexity of this method, let $|S| = n$ and $|T| = m$. In the construction $|V| = O(n * m)$, and since there are at most three edges per node, $|E| = O(|V|)$. Thus the construction takes $O(|V|) = O(n*m)$ time. The algorithm for finding the shortest path between two elements in a directed acyclic graph takes time $O(|V|+|E|)$ which in this case takes $O(|V|) = O(n*m)$ time. Thus the total complexity of the method takes $O(n*m) = O(n^2)$ time.

6 Acknowledgement

We thank Marc Berndl and Paco Gomez for helpful discussions concerning this research.

References

[1] Alok Aggarwal, Amotz Bar-Noy, Samir Khuller, Dina Kravets, and Baruch Schieber. Efficient minimum cost matching and transportation using the quadrangle inequality. *Journal of Algorithms*, 19(1):116–143, 1995.

[2] Amir Ben-Dor, Richard M. Karp, Benno Schwikowski, and Ron Shamir. The restriction scaffold problem. *Journal of Computational Biology*, 10(2):385–398, 2003.

[3] Cormen, Leiserson, and Rivest. *Introduction to Algorithms.* MIT Press, Cambridge Mass., 1990.

[4] J. Miguel Dìaz-Báñez, Giovanna Farigu, Francisco Gomez, David Rappaport, and Godfried Toussaint. El compás flamenco: A phylogenetic analysis. In *Proceedings of BRIDGES: Mathematical Connections in Art, Music, and Science*, pages 61–70, July 30 to August 1 2004.

[5] Thomas Eiter and Heikki Mannila. Distance measures for point sets and their computation. *Acta Informatica*, 34(2):109–133, 1997.

[6] Richard M. Karp and Shuo-Yen R. Li. Two special cases of the assignment problem. *Discrete Mathematics*, 13(2):129 – 142, 1975.

[7] G. Oddie. *Likeness to Truth.* D. Reidel Publishing Company, Dordrecht, Holland, 1986.

[8] Godfried Toussaint. A comparison of rhythmic similarity measures. In *Proc. 5th International Conference on Music Information Retrieval*, pages 242–245, 2004.

[9] G.T. Toussaint. Classification and phylogenetic analysis of african ternary rhythm timelines. In *Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science*, pages 25–36, 2003.