An $O(n \log n)$-Time Algorithm for the Restriction Scaffold Assignment Problem

Justin Colannino
School of Comp. Sci., McGill Univ.
jcolan@cs.mcgill.edu

Mirela Damian*
Dept. of Comp. Sci., Villanova Univ.
mirela.damian@villanova.edu

Ferran Hurtado
Dept. de Matemàtica Aplicada I
Univ. Politècnica de Catalunya
Ferran.Hurtado@upc.edu

John Iacono
Dept. of Comp. and Inf. Sci.
Polytechnic Univ.
jiacono@poly.edu

Henk Meijer
School of Computing, Queen's Univ.
henk@cs.queensu.ca

Suneeta Ramaswami
Dept. of Comp. Sci., Rutgers Univ.
rsuneeta@camden.rutgers.edu

Godfried Toussaint
School of Comp. Sci., McGill Univ.
godfried@cs.mcgill.edu

*Corresponding author, Villanova University, Villanova, PA 19085. Phone: (610)-519-7414. Fax: 610-519-7889.

Abstract

The *restriction scaffold assignment* problem takes as input two finite point sets S and T (with S containing more points than T) and establishes a correspondence between points in S and points in T, such that each point in S maps to exactly one point in T, and each point in T maps to at least one point in S. In this paper we present an algorithm that finds a minimum-cost solution for this problem in $O(n \log n)$ time, provided that the points in S and T are restricted to lie on a line (linear time, if S and T are presorted). This improves the previously best-known $O(n^2)$-time algorithm for this problem.

1 Introduction

Consider two finite sets of points, a *source* set S and a *target* set T, with the cardinality of S greater than the cardinality of T, and total cardinality n. The objective of the *restriction scaffold assignment* problem is to establish a correspondence between the points in S and the points in T, such that each point in S corresponds to exactly one point in T, and each point in T corresponds to at least one point in S. This correspondence is measured by a cost function δ that assigns a cost $\delta(s, t)$ to each assigned pair (s, t). The cost of an assignment is the sum of the costs of all assigned pairs. The goal of this assignment problem is to find an assignment of minimum cost. This assignment problem is also known as the *many-to-one assignment* problem. The *one-to-one* version of this problem requires that each point in S be assigned to at most one point in T and each point in T be assigned exactly one point from S. Finally, the *many-to-many* version requires that each point in S and T be assigned at least one point in the other set. This latter version is also known as the *minimum-cost edge-covering* problem.

The simplest version of the assignment problem assumes that the points in S and T lie on a line and the cost function is the distance between pairs of points in the L_1 metric. In this setting, the one-to-one assignment problem has a simple $O(n \log n)$-time solution when $|S| = |T|$: first sort the

points in $O(n \log n)$ time, then assign the k^{th} point in S to the k^{th} point in T in $O(n)$ time [6], [16]. However, the situation $|S| > |T|$ arises in many practical applications, some of which we mention below. This situation was first addressed by Karp and Li [11], who provided an $O(n \log n)$-time algorithm for the one-to-one assignment problem (linear time, if S and T are given in sorted order). Simpler and equally efficient solutions have later been provided in [1, 4, 18].

The many-to-one assignment problem, or simply the assignment problem, appears in computational biology as the *restriction scaffold assignment* [3]. The goal here is to establish a correspondence between sparse experimental data (set S) and a restricted set of known structural building blocks (set T). Ben-Dor et. al. [3] modeled the restriction scaffold assignment as an assignment problem for points on a line, and suggested an $O(n \log n)$-time algorithm to solve this problem. However, Colannino and Toussaint [5] showed that this algorithm sometimes fails to yield a minimum-cost assignment. Thus, the best existing solution to this problem is the $O(n^2)$ algorithm given in [5]. In this paper we improve this result to $O(n \log n)$ for an arbitrary ordering of the input, and $O(n)$ if the input is in sorted order.

Eiter and Mannila [8] studied the assignment problem in the context of measuring the distance between two theories expressed in a logical language. They showed that for points in arbitrary dimensions, this problem has an $O(n^3)$-time solution that uses the Hungarian method [13]. When the points are restricted to a line, a minimum-cost assignment can be used in measuring the similarity between musical rhythms. In this context, Toussaint [17] proposed the use of the *swap distance* as a similarity measure when S and T have equal cardinalities. For the case of unequal cardinalities he generalized the swap distance to the *directed swap distance*, where the "direction" of the assignment (surjection) is from the larger set (S) to the smaller set (T). This similarity measure has since been successfully applied to a phylogenetic analysis of Flamenco metric patterns [6]. If the onsets of a rhythm are represented as points on a line separated by "silence" intervals, the directed swap distance between two rhythms represented by the sets S and T is precisely the cost

of an optimal assignment between S and T, with underlying cost function L_1.

Related to these problems, the *many-to-many* assignment problem is a restricted version of *bibranchings* first introduced by Schrijver [14]. Let $D = (V, E)$ be a directed graph, and let V be partitioned into two disjoint sets, the *source* vertices S and the *target* vertices T. A *bibranching* in D with respect to S is a set of edges B in E such that:

for each v in S, B contains a directed path from v to a vertex in T, and

for each v in T, B contains a directed path from a vertex in S to v.

For the special case when D is a bipartite graph with color classes S and T, and all the edges in D are directed from S to T, the bibranching is a *bipartite edge cover*. Keijsper and Pendavingh [12] describe an $O(|E|)$-time algorithm attributed to J. F. Geelen for reducing the minimum-cost bipartite edge cover problem to the maximum-cost matching problem. They also describe a solution for the latter problem that uses shortest path algorithms from [7] and [15] sped up with Fibonacci heaps [9]. Their algorithm runs in time $O(n'(|E| + n \log n))$, where $n' = min\{|S|, |T|\}$; this complexity is $O(n^3)$ in the worst case, matching the complexity of the approach of Eiter and Mannila. See [2, 10] for a survey on matching algorithms.

In this paper, we show that the many-to-one assignment problem with underlying cost function L_1 in one dimension can be solved in $O(n \log n)$ time ($O(n)$ time, if S and T are given in sorted order). Our algorithm is a simple extension of the $O(n \log n)$-time algorithm of Karp and Li [11] for finding a minimum-cost *one-to-one* assignment.

2 Preliminaries

Let $S = \{s_0, s_1, s_2, \ldots\}$ and $T = \{t_0, t_1, t_2, \ldots\}$ be two finite sets of points that lie on a horizontal line, with $|S| + |T| = n$ and $|S| > |T|$. For any $s \in S$ and $t \in T$, the cost $\delta(s, t)$ of an assigned pair (s, t) is the absolute value of the difference between the x-coordinates of s and t. To avoid

overloading the notation, we use the same symbol for a point and its x-coordinate. Thus, $\delta(s,t) = |s-t|$. We assume that $s_i < s_{i+1}, 0 \le i < |S|-1$ and $t_j < t_{j+1}, 0 \le j < |T|-1$.

An assignment $\mathcal{A}$ between S and T consists of pairs of points (s,t) (henceforth *edges*), with $s \in S$ and $t \in T$, such that each point in S belongs to exactly one edge in $\mathcal{A}$, and each point in T belongs to at least one edge in $\mathcal{A}$. The cost of $\mathcal{A}$ is

$$cost(\mathcal{A}) = \sum_{(s,t)\in\mathcal{A}} \delta(s,t)$$

Our goal is to find an assignment $\mathcal{A}$ of minimum cost. If two points in $S \cup T$ have the same x-coordinate, we can slightly shift one of them to the left or right. If the minimum-cost assignment is unique and the change is sufficiently small, this change will not affect the optimal assignment. If there are several assignments with the same optimal cost, at least one of them will be the optimal solution of the new point set. So we may assume without loss of generality that all points in $S \cup T$ are distinct.

For any $s \in S$ and $t \in T$, the value $|s-t|$ can be expressed in a different way as follows. Define a function $f_{s,t}$ to be 1 in the interval between s and t and 0 at any other point (see Figure 1). Then $|s-t| = \int_{-\infty}^{+\infty} f_{s,t}(x)dx$.

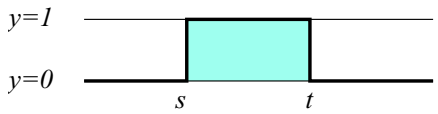

Figure 1: Function $f_{s,t}$. Shaded area represents the cost $|s-t|$.

The cost of an assignment $\mathcal{A}$ is therefore

$$cost(\mathcal{A}) = \sum_{(s,t)\in\mathcal{A}} \int_{-\infty}^{+\infty} f_{s,t}(x)dx = \int_{-\infty}^{+\infty} \sum_{(s,t)\in\mathcal{A}} f_{s,t}(x)dx$$

If we define

$$f_{\mathcal{A}}(x) = \sum_{(s,t)\in\mathcal{A}} f_{s,t}(x)$$

then the value $f_{\mathcal{A}}(a)$ is simply the number of edges in $\mathcal{A}$ pierced by the vertical line $x = a$, and the cost of $\mathcal{A}$ is

$$cost(\mathcal{A}) = \int_{-\infty}^{+\infty} f_{\mathcal{A}}(x)dx \tag{1}$$

Our definition of $f_{\mathcal{A}}$ is similar in nature to the *height* function $H : \mathbb{R} \to \mathbb{Z}$ introduced by Karp and Li [11]. Informally, they define $H(a)$ at each point a as the difference between the number of points in S and the number of points in T restricted to the interval $(-\infty, a]$ (or equivalently, to the left of the vertical line $x = a$). Thus H remains constant throughout each interval that does not contain a point in $S \cup T$. Figure 2 shows the stair-shaped curve of H for a small example. Note that *up* transitions in the curve correspond to points in S and *down* transitions correspond to points in T. We refer to the value $H(x)$ as the *height* of x. Note that $H(\infty) = |S| - |T|$.

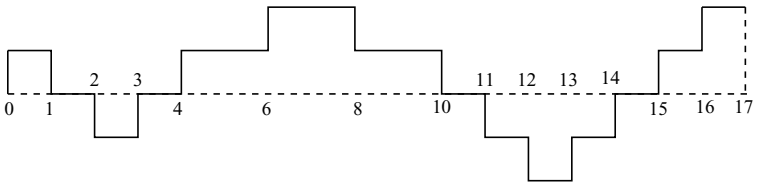

Figure 2: Height function for sets $S = \{0, 3, 4, 6, 13, 14, 15, 16\}$ and $T = \{1, 2, 8, 10, 11, 12\}$.

Lemma 1 *If $|S| = |T|$, then $\int_{-\infty}^{+\infty} |H(x)|\ dx$ is the cost of the assignment that assigns the k^{th} largest element of S to the k^{th} largest element of T.*

Proof: Follows immediately from (1) and the fact that, for this particular assignment, $f_{\mathcal{A}}(x) = |H(x)|$ at each point x. □

Figure 3a shows an assignment for two sets S and T, with $|S| = |T|$. The cost of this assignment is equal to the area shaded in Figure 3b, which is precisely the value of the integral $\int_{-\infty}^{+\infty} |H(x)|\ dx$.

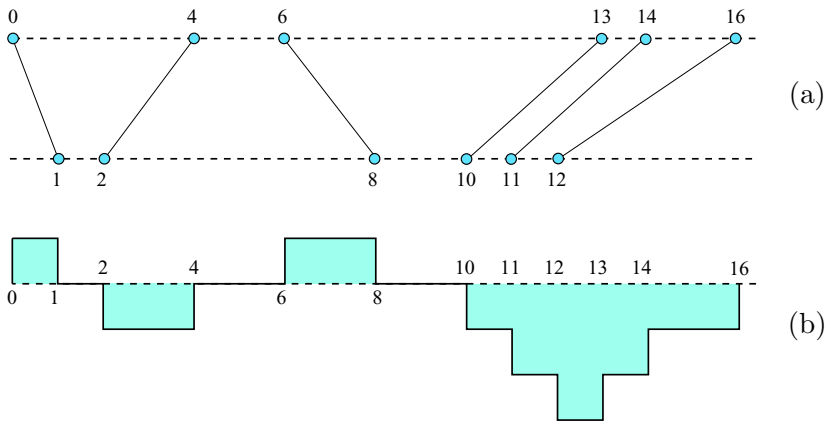

Figure 3: (a) One-to-one assignment for sets $S = \{0, 4, 6, 13, 14, 16\}$ and $T = \{1, 2, 8, 10, 11, 12\}$ (b) Shaded area represents the cost of the assignment.

3 Properties of a Minimum Cost Assignment

Our algorithm for computing a minimum-cost assignment $\mathcal{A}$ exploits several important properties of $\mathcal{A}$, which we discuss next. A *crossing* is defined by a pair of edges (a, d) and (b, c) such that $a < b$ in S and $c < d$ in T.

Lemma 2 *There exists a minimum-cost assignment with no crossings.*

Proof: Let $\mathcal{A}$ be a minimum-cost assignment between S and T with a minimum number of crossings. If $\mathcal{A}$ has zero crossings, the proof is finished. Otherwise, pick two crossing edges (a, d) and (b, c) in $\mathcal{A}$, with $a < b$ in S and $c < d$ in T. We show that $\mathcal{A}' = \mathcal{A} \backslash \{(a, d), (b, c)\} \cup \{(a, c), (b, d)\}$ is an assignment with $cost(\mathcal{A}') \leq cost(\mathcal{A})$, a contradiction. In particular, we show that $f_{\mathcal{A}'}(x) \leq f_{\mathcal{A}}(x)$ at each point x; then $cost(\mathcal{A}') \leq cost(\mathcal{A})$ follows immediately from (1).

First note that $f_{\mathcal{A}'}(x) \leq f_{\mathcal{A}}(x)$ is true for any x such that the vertical line L at x intersects neither of (a, d) and (b, c). Suppose now that L intersects (a, c). Then L must also intersect either (a, d) (see Figure 4a) or (b, c) (see Figure 4b) or both (see Figure 4c). Similarly, if L intersects (b, d), then L also intersects at least one of (a, d) and (b, c). Furthermore, if L intersects both (a, c) and (b, d), then L also intersects both (a, d) and (b, c) (see Figure 4c). It follows that $f_{\mathcal{A}'}(x) \leq f_{\mathcal{A}}(x)$.

□

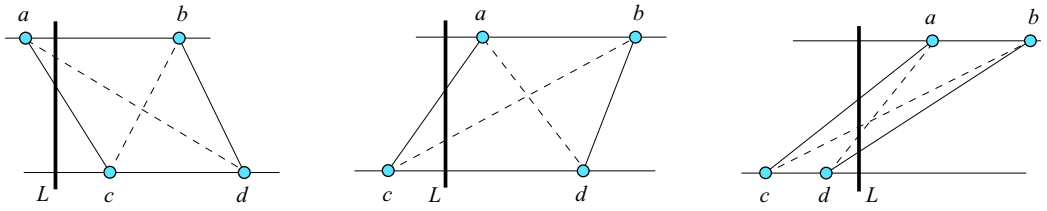

Figure 4: (a) Vertical line L intersects (a, c) and (a, d) (b) L intersects (a, c) and (b, c) (c) L intersects (a, c), (b, d), (a, d) and (b, c).

An assignment $\mathcal{A}$ can also be regarded as a function $\mathcal{A} : S \rightarrow T$ such that $\mathcal{A}(s) = t$ for each $(s, t) \in \mathcal{A}$. For any $t \in T$, let $\mathcal{A}^{-1}(t)$ denote the set of elements $s \in S$ such that $\mathcal{A}(s) = t$. For each point $s \in S$, define the *nearest neighbor* $N(s)$ to be point in T closest to s, i.e, $|N(s)-s| \leq |t-s|$ for any $t \in T$. In the case of a tie, $N(s)$ is arbitrarily picked from among the two candidate neighbors.

Lemma 3 *Let $\mathcal{A}$ be optimal and let $t \in T$ be such that $\mathcal{A}^{-1}(t)$ contains two or more elements. Then for each $s \in \mathcal{A}^{-1}(t)$, t is a nearest neighbor of s. Furthermore, T contains no points in between s and t.*

Proof: Assume to the contrary that there is $s \in S$ with $\mathcal{A}(s) = t, |\mathcal{A}^{-1}(t)| > 1$, and $N(s) \neq t$. Refer to Figure 5. Define a new assignment $\mathcal{A}'$ with $\mathcal{A}'(s) = N(s)$ and $\mathcal{A}'(x) = \mathcal{A}(x)$ for $x \neq s$. Note that $\mathcal{A}'$ is also an assignment: $\mathcal{A}^{-1}(t)$ contains at least one point. Also $cost(\mathcal{A}') = cost(\mathcal{A}) - |s - t| + |s - N(s)|$ (see Figures 5a and 5b). Since $|s - N(s)| < |s - t|$, it follows that $cost(\mathcal{A}') < cost(\mathcal{A})$,

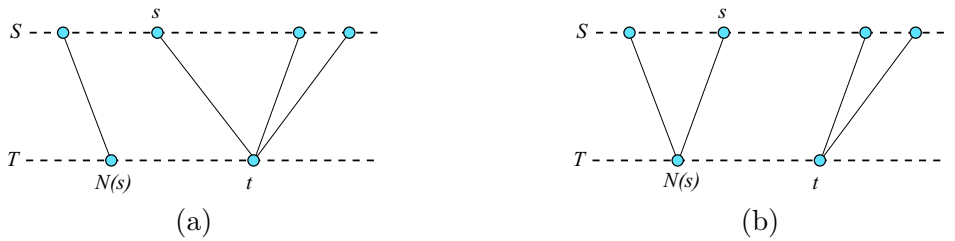

Figure 5: (a) Assignment $\mathcal{A}$ with $\mathcal{A}(s) \neq N(s)$ (b) Assignment $\mathcal{A}'$ with $\mathcal{A}'(s) = N(s)$

contradicting the fact that $\mathcal{A}$ is of minimum cost. Thus, t is a nearest neighbor of s.

The claim that T contains no points in between s and t is immediate: if such a point $t_1 \in T$ existed, then $|s - t_1| < |s - t|$, contradicting the fact that $N(s) = t$. □

Observe that for any subset $R \subset S$ of size $|R| = |S| - |T|$, there is a unique minimum-cost assignment (with no crossings) from $S \setminus R$ to T. Let $\mathcal{A}_{S \setminus R}$ denote the edges of such an assignment, and define a new assignment $\mathcal{A}_R : S \to T$ as follows:

$$\mathcal{A}_R(x) = \begin{cases} N(x) & \text{if } x \in R, \\ y & \text{if } x \in S \setminus R \text{ and } (x, y) \in \mathcal{A}_{S \setminus R} \end{cases} \tag{2}$$

Lemma 3 implies that there always exists a subset R such that $\mathcal{A}_R$ defines a minimum-cost assignment from S to T. Furthermore, R satisfy a special height condition, stated in the lemma below.

Lemma 4 *There exists a subset $R \subset S$ with $|R| = |S| - |T|$ such that $\mathcal{A}_R$ defines a minimum-cost assignment from S to T, and the k^{th} smallest element of R has height k.*

Proof: Let $\mathcal{A} : S \to T$ define a minimum-cost assignment. We prove the existence of $\mathcal{A}_R$ by constructing a set $R \subset S$ with the properties stated in this lemma. Initially R is empty. If $|\mathcal{A}^{-1}(t)| = 1$ for all $t \in T$, then R is empty and the proof is finished. Otherwise, we process points $t \in T$ for which $\mathcal{A}^{-1}(t)$ has two or more elements. For each such point we consider two cases, as depicted in Figure 6. If all points in $\mathcal{A}^{-1}(t)$ are less than t, then we add to R all but the largest (rightmost) point in $\mathcal{A}^{-1}(t)$ (see Figure 6a). Otherwise, we add to R all points in $\mathcal{A}^{-1}(t)$ except for the smallest (leftmost) point greater than t (see Figure 6b).

We now define A_R as in (2). Since $\mathcal{A}_R$ is identical to $\mathcal{A}$, $\mathcal{A}_R$ is a minimum cost many-to-one assignment from S to T.

It remains to show that the k^{th} smallest element of R has height k. To see this, first consider the smallest element of a nonempty set $\mathcal{A}^{-1}(t) \cap R$. Call this element r and suppose it is the k^{th} smallest element of R. It follows then that (i) R contains $k - 1$ points less than r, and (ii) T and $S \setminus R$ contain an equal number of elements less than r. This latter claim follows from Lemma 3,

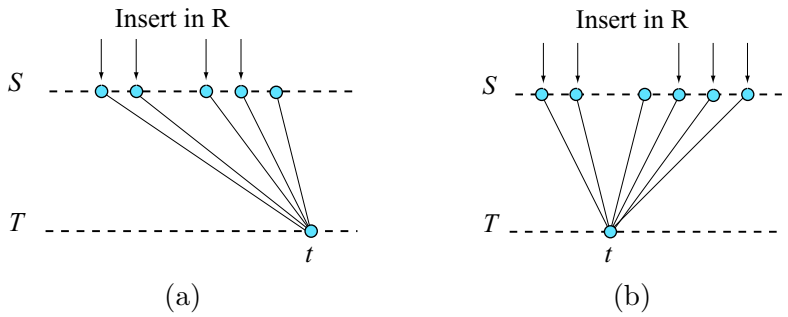

Figure 6: (a) All points in $\mathcal{A}^{-1}(t)$ are less than t. (b) Some points in $\mathcal{A}^{-1}(t)$ are greater than t.

which tells us that T contains no elements in between r and t, and the following observation: the way in which we have selected R ensures that if t lies to the left of r (i.e., $t < r$), the assigned item for t in S/R lies to the left of r, and if t lies to the right of r ($t > r$), the assigned item for t in S/R lies to the right of r. These together imply that $H(r) = k$.

We now show that the points in $\mathcal{A}^{-1}(t) \setminus \{r\}$ have height values $k+1, k+2, \ldots$, in order from smallest to largest. By Lemma 3, T contains no points in between s and t, for each $s \in \mathcal{A}^{-1}(t)$. Then the points in $R \cap \mathcal{A}^{-1}(t)$ have incrementally increasing height values. It follows that the height of the k^{th} smallest element of R is k. □

Let H_R represent the height function restricted to sets $S \setminus R$ and T. This means that for each x, $H_R(x)$ is the difference between the number of points in $S \setminus R$ and the number of points in T restricted to the interval $(-\infty, x]$.

Lemma 5 *The cost of assignment $\mathcal{A}_R$ is*

$$\sum_{r \in R} |r - N(r)| + \int_{-\infty}^{+\infty} |H_R(x)| dx \tag{3}$$

Proof: By Lemma 1 we have that the contribution of $S \setminus R$ to the cost of $\mathcal{A}_R$ is $\int_{-\infty}^{+\infty} |H_R(x)| dx$. Since each point in R maps to its nearest neighbor, the contribution of R to the cost of $\mathcal{A}_R$ is $\sum_{r \in R} |r - N(r)|$. These together conclude the lemma. □

Theorem 6 *Let $R \subset S$ be a subset of size $|R| = |S| - |T|$ with two properties:*

i. The k^{th} smallest element of R has height k.

ii. R minimizes the quantity from (3).

Then $\mathcal{A}_R$ defines a minimum-cost assignment from S to T.

Proof: By Lemma 4, we know that there exists a set R that satisfies *(i)*. By Lemma 5, R satisfies *(ii)*. It follows that $\mathcal{A}_R$ is a minimum-cost assignment from S to T. □

4 Computing a Minimum Cost Assignment

Theorem 6 gives an exact description of the set R that yields a minimum-cost assignment $\mathcal{A}_R$. We now turn to the problem of efficiently determining this set. With this goal in mind, we introduce the following notation. For any point x and any integer k, define the *relative height* of x with respect to k as

$$h^k(x) = \begin{cases} 1, & \text{if } H(x) \geq k \\ -1, & \text{if } H(x) < k \end{cases}$$

Observe that when a point s is removed from S, $H(x)$ decreases by 1 for all $x > s$. Suppose that $H(s) = k$, and let m be the largest point in $S \cup T$. The removal of s causes the area under the height function between s and m to decrease by the quantity $\int_s^m h^k(x)dx$. We use this observation to define the *profit* of removing s from S and placing it in R (recall that $\mathcal{A}_R$ assigns each item in R to its nearest neighbor), as follows:

$$P(s) = \int_s^m h^k(x)dx - |s - N(s)| \tag{4}$$

The profit function quantifies the benefit of placing s in R, the goal being to minimize the cost of the assignment defined by $\mathcal{A}_R$. The integral term in (4) represents the effect of excluding s from

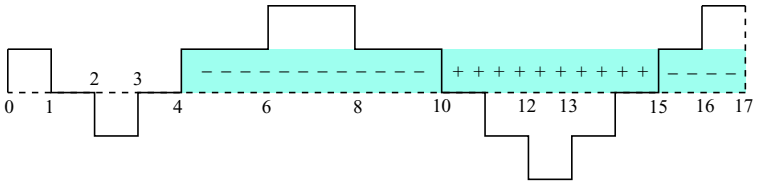

Figure 7: A depiction of the integral $\int_s^m h^k(x)dx$ for $s = 4$. The integral represents the effect of excluding 4 from the one-to-one assignment from S to T.

the one-to-one assignment from $S \setminus R$ to T, as depicted in Figure 7. The term $|s - N(s)|$ in (4) represents the cost of assigning s to its nearest neighbor. We minimize the cost of the assignment defined by $\mathcal{A}_R$ by choosing items s that maximize $P(s)$. This is formalized in the following lemma.

Lemma 7 *Let $R \subset S$ be a set with elements $r_1 < r_2 \ldots < r_{|S|-|T|}$ such that $H(r_k) = k$ and r_k maximizes $P(s)$ among all points $s \in S$ of height k. Then R minimizes*

$$\sum_{r \in R} |r - N(r)| + \int_{-\infty}^{+\infty} |H_R(x)|dx$$

Proof: Karp and Li [11] proved that any set R of size $|S| - |T|$ whose k^{th} smallest element has height k satisfies the equality

$$\int_{-\infty}^{+\infty} |H_R(x)|dx = \int_0^m |H(x)|dx - \sum_{r \in R} \int_r^m h^k(x)dx$$

Summing up the cost contribution of R to both sides of the equality yields

$$\sum_{r \in R} |r - N(r)| + \int_{-\infty}^{+\infty} |H_R(x)|dx = \sum_{r \in R} |r - N(r)| + \int_0^m |H(x)|dx - \sum_{r \in R} \int_r^m h^k(x)dx$$

This is equivalent to

$$\sum_{r \in R} |r - N(r)| + \int_{-\infty}^{+\infty} |H_R(x)|dx = \int_0^m |H(x)|dx - \sum_{r \in R} P(r)$$

Since $P(r_k)$ is maximized at each height k and there is only one element in R at each height, we have that R maximizes $\sum_{r \in R} P(r)$, which in turn minimizes

$$\sum_{r \in R} |r - N(r)| + \int_{-\infty}^{+\infty} |H_R(x)| dx$$

as required (refer to Lemma 5). □

The following algorithm uses the preceding lemma to determine the optimal set R, and then compute the minimum-cost assignment.

4.1 The Assignment Algorithm

Initially R is the empty set.

1. Sort S and T.
2. Calculate $H(x)$ for each $x \in S \cup T$. In between consecutive points, H is constant.
3. Calculate $P(s)$ for each $s \in S$.
4. For $k = 1, 2, \ldots |S| - |T|$
   - 4.1 Find the leftmost point r_k of height k that maximizes $P(r_k)$.
   - 4.2 Add r_k to R.
5. Return $\mathcal{A}_R$.

Lemma 8 *The assignment algorithm computes a minimum-cost assignment from S to T.*

Proof: Let r_k be the element of R of height k returned by the algorithm. If we show that $r_1 < r_2 < \ldots < r_{|S|-|T|}$, then it follows by Lemma 7 that $\mathcal{A}_R$ is a minimum-cost assignment. We prove below, by contradiction, that indeed $r_1 < r_2 < \ldots < r_{|S|-|T|}$.

Let m be the largest point in S. Assume that there exists some $k(1 \leq k \leq |S| - |T| - 1)$ for which the algorithm returns r_k and r_{k+1}, with $r_k > r_{k+1}$. Let s_k be the maximal element at height k in $S \setminus R$ which is less than r_{k+1}. By continuity, such an s_k must exist. Similarly, let s_{k+1} be the minimal element at height $k+1$ in $S \setminus R$ which is greater than r_k. Such an s_{k+1} must exist since the height at ∞ is $H(\infty) = |S| - |T|$. Refer to Figure 8.

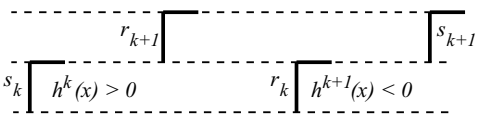

Figure 8: $s_k(s_{k+1})$ is the closest point at height $k(k+1)$ to the left (right) of $r_{k+1}(r_k)$.

Since $H(r_{k+1}) = H(s_{k+1})$ and $r_{k+1} < s_{k+1}$, we have that

$$\int_{r_{k+1}}^{m} h^{k+1}(x)dx = \int_{r_{k+1}}^{s_{k+1}} h^{k+1}(x)dx + \int_{s_{k+1}}^{m} h^{k+1}(x)dx$$

From this and equation (4), we can derive the following relation between the profit functions of r_{k+1} and s_{k+1}:

$$P(r_{k+1}) = P(s_{k+1}) + \int_{r_{k+1}}^{s_{k+1}} h^{k+1}(x)dx - |r_{k+1} - N(r_{k+1})| + |s_{k+1} - N(s_{k+1})| \tag{5}$$

Note that equality (5) is the result of breaking up the integral corresponding to $P(r_{k+1})$ into two parts, and taking into account the distance from each element to its nearest neighbor. Similarly, we can derive the following relation between $P(r_k)$ and $P(s_k)$:

$$P(s_k) = P(r_k) + \int_{s_k}^{r_k} h^k(x)dx - |s_k - N(s_k)| + |r_k - N(r_k)| \tag{6}$$

The nearest neighbor of s_k cannot be farther than $N(r_{k+1})$. This translates into:

$$|s_k - N(s_k)| \leq |r_{k+1} - N(r_{k+1})| + |s_k - r_{k+1}|$$

Also note that $h^k(x)$ is positive on the interval (s_k, r_{k+1}), which allows us to rewrite the previous equation as:

$$|s_k - N(s_k)| \leq |r_{k+1} - N(r_{k+1})| + \int_{s_k}^{r_{k+1}} h^k(x)dx \tag{7}$$

Similar arguments lead to the following relationship between nearest neighbors of r_k and s_{k+1}:

$$|r_k - N(r_k)| \geq |s_{k+1} - N(s_{k+1})| + \int_{r_k}^{s_{k+1}} h^{k+1}(x)dx \tag{8}$$

Finally, on the interval (r_{k+1}, r_k) note that

$$\int_{r_{k+1}}^{r_k} h^{k+1}(x)dx \leq \int_{r_{k+1}}^{r_k} h^k(x)dx \tag{9}$$

Let $M_k = |s_k - N(s_k)| - |r_k - N(r_k)|$. Simple arithmetic that involves inequalities (7), (8) and (9) yields

$$\int_{s_k}^{r_k} h^k(x)dx - M_k \geq \int_{r_{k+1}}^{s_{k+1}} h^{k+1}(x)dx + M_{k+1}$$

This along with (5) and (6) implies that

$$P(s_k) - P(r_k) \geq P(r_{k+1}) - P(s_{k+1})$$

Since r_{k+1} was picked by the assignment algorithm, we have that $P(r_{k+1}) \geq P(s_{k+1})$. This implies that $P(s_k) \geq P(r_k)$, but since s_k lies to the left of r_k, the assignment algorithm would have picked s_k instead of r_k, a contradiction. □

4.2 Complexity Analysis

Sorting in step 1 takes $O(n \log n)$ time. All other steps run in $O(n)$ time. The only steps where this is not obvious are steps 2 and 3 that involve computing $H(x)$ and $P(x)$ respectively. $H(x)$ can

be computed for all $s \in S$ by conducting a sweep of the sorted points in $S \cup T$, adding one when we encounter an element of S and subtracting one when we encounter an element of T.

Since all nearest neighbors of the elements of S can easily be computed in linear time, to show that we can compute the profit function for all elements of S in linear time we concern ourselves only with computing the integral of relative height function h^k. This integral can be computed in linear time for all points in S at height k in a sweep from right to left. For the rightmost element s_r of S at height k $\int_{s_r}^{m} h^k(x)dx = |s_r - m|$, where m is the largest point in S. Suppose that we know $\int_{s}^{m} h^k(x)dx$ for some item s at height k. Let $s' < s$ be the largest element in S also at height k, and let $t < s$ be the largest element in T at height k. Note that by continuity, t exists and must be greater than s'. Also note that $h^k(x)$ is positive for all $s' \leq x \leq t$, and $h^k(x)$ is negative for all $t < x < s$. Thus we can derive the following equation:

$$\int_{s'}^{m} h^k(x)dx = \int_{s}^{m} h^k(x)dx + |s' - t| - |t - s| \tag{10}$$

This value can be computed in constant time for each $s' \in S$. Thus we can compute $P(s)$ for all $s \in S$ in linear time.

It follows that the assignment algorithm runs in $O(n \log n)$ time. Furthermore, if S and T are given in sorted order, the assignment algorithm runs in $O(n)$ time.

5 Conclusion

We have shown that the one-to-one assignment algorithm in [11] can be extended to produce a minimum-cost many-to-one assignment. The algorithm runs in $O(n \log n)$ time, if the input points are given in arbitrary order, and in $O(n)$ time, if the input points are presorted. To our knowledge, this is the first solution to the assignment problem that achieves this time complexity.

References

[1] A. Aggarwal, A. Bar-Noy, S. Khuller, D. Kravets, and B. Schieber. Efficient minimum cost matching and transportation using the quadrangle inequality. *J. Algorithms*, 19(1):116–143, 1995.

[2] D. Avis. A survey of heuristics for the weighted matching problem. *Networks*, 13:475–493, 1983.

[3] A. Ben-Dor, R.M. Karp, B. Schwikowski, and R. Shamir. The restriction scaffold problem. *J. of Computational Biology*, 10(2):385–398, 2003.

[4] S. R. Buss and P. N. Yianilos. Linear and o(n log n) time minimum-cost matching algorithms for quasi-convex tours. *SIAM J. of Computing*, 27(1):170–201, 1998.

[5] J. Colannino and G. Toussaint. An algorithm for computing the restriction scaffold assignment problem in computational biology. *Information Processing Letters*, 95(Issue 4):466–471, 2005.

[6] Miguel Díaz-Bañez, Giovanna Farigu, Francisco Gómez, David Rappaport, and Godfried T. Toussaint. El compás flamenco: a phylogenetic analysis. In *Proc. of BRIDGES: Math. Connections in Art, Music and Science*, Southwestern College, Winfield, Kansas, 2004.

[7] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. of the Association for Computing Machinery*, 19:248–264, 1972.

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

[9] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. of the Association for Computing Machinery*, 34:596–615, 1987.

[10] Z. Galil. Efficient algorithms for finding maximum matchings in graphs. *ACM Computing Surveys*, 18:23–38, 1986.

[11] R.M. Karp and S.-Y.R. Li. Two special cases of the assignment problem. *Discrete Mathematics*, 13(46):129–142, 1975.

[12] J. Keijsper and R. Pendavingh. An efficient algorithm for minimum-weight bibranching. *J. of Combinatorial Theory*, 73(Series B):130–145, 1998.

[13] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics*, 2:83–97, 1955.

[14] A. Schrijver. Min-max relations for directed graphs. *Annals of Discrete Mathematics*, 16:261–280, 1982.

[15] N. Tomizawa. On some techniques useful for the solution of transportation network problems. *Networks*, 1:173–194, 1972.

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

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

[18] M. Werman, S. Peleg, R. Melter, and T. Kong. Bipartite graph matching for points on a line or a circle. *J. Algorithms*, 7:277–284, 1986.