



# Constrained many-to-many point matching in two dimensions

L. E. Caraballo<sup>1</sup> · R. A. Castro<sup>2</sup> · J. M. Díaz-Báñez<sup>1</sup> · M. A. Heredia<sup>2</sup> · J. Urrutia<sup>3</sup> · I. Ventura<sup>1</sup>  · F. J. Zaragoza<sup>2</sup>

Received: 10 May 2023 / Accepted: 15 December 2023  
© The Author(s) 2024

## Abstract

In the minimum-weight many-to-many point matching problem, we are given a set  $R$  of red points and a set  $B$  of blue points in the plane, of total size  $N$ , and we want to pair up each point in  $R$  to one or more points in  $B$  and vice versa so that the sum of distances between the paired points is minimized. This problem can be solved in  $O(N^3)$  time by using a reduction to the minimum-weight perfect matching problem, and thus, it is not fast enough to be used for on-line systems where a large number of tunes need to be compared. Motivated by similarity problems in music theory, in this paper we study several constrained minimum-weight many-to-many point matching problems in which the allowed pairings are given by geometric restrictions, i.e., a bichromatic pair can be matched if and only if the corresponding points satisfy a specific condition of closeness. We provide algorithms to solve these constrained versions in  $O(N)$  time when the sets  $R$  and  $B$  are given ordered by abscissa.

**Keywords** Many-to-many matching · Dynamic programming · Melodic similarity · Geometric measure

## 1 Introduction

Given two finite sets of red and blue points in the plane,  $R$  and  $B$ , respectively, with total size  $N$ , a *point matching* between  $R$  and  $B$  pairs up red and blue points. Although matching problems have many applications, e.g. shape matching or color-based image retrieval, there has been an increase in the activity for studying music score matching in the field of music information retrieval (MIR). MIR is actually a major research area whose main objective is to develop methods for finding musical information from a digitised collection of musical works, that is, encoded scores or audio recordings [1]. It is well known that melodies are rich in

---

L. E. Caraballo, R. A. Castro, J. M. Díaz-Báñez, M. A. Heredia, J. Urrutia, I. Ventura and F.J. Zaragoza contributed equally to this work.

Extended author information available on the last page of the article

features that allow distinguishing between them and people are able to recognize a large number of musical works from their principal melodies. Thus, comparing two melodies is one of the most critical research topics in the area. At this point, it is worth noting that even though MIR research has been carried out on mainstream Western music, there is a great current interest in oral tradition musics where, in addition to basic parameters as pitch, duration, loudness and timbre, higher level music features, as voice leading, phrasing, and ornamentation, are required to be incorporated in the study of perceptual similarity [2, 3].

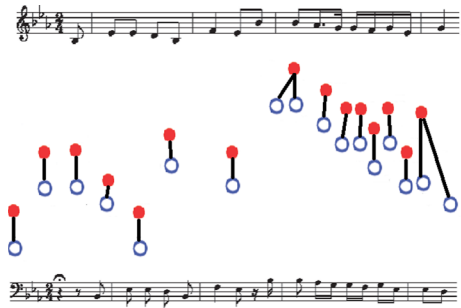
A common problem of a MIR system is to match two melodies and establish a similarity rank. In this way, a melody can be modeled as a set of points in the plane formed by the (on-set time, pitch) pairs [4] and the similarity between two melodies can be measured by the cost (or weight) of an optimal matching between the sets of points representing each melody. The cost of each pair in the matching is commonly measured using the Manhattan distance [5, 6], although other metrics have also been tested: the Euclidean distance, the cosine metric, the Pearson correlation, and others. Also, a graph-based representation of symbolic musical data has been considered [7].

In this framework, some papers studied versions of the minimum-weight matching problem [8–12]. For example, given two finite sets of red and blue points in the plane,  $R$  and  $B$ , respectively, in a *many-to-one matching* from  $R$  to  $B$  with  $|R| < |B|$ , each point of  $R$  is assigned to one or more points of  $B$ , and each point of  $B$  is assigned to only one point of  $R$  [9]. An  $O(|B|^2)$  time algorithm based on a reduction to the problem of computing the single source shortest path problem in a weighted directed acyclic graph has been proposed in [8].

Other type of matching used is that in which each point of  $R$  (respectively,  $B$ ) is assigned to at least one point in  $B$  (respectively,  $R$ ), the *many-to-many matching*. The minimum-weight many-to-many matching problem or minimum bipartite edge cover was studied in [13]. It can be solved in  $O(N^3)$  time using the Hungarian method proposed by Kuhn [14]. The many-to-many matching in the special case where  $R$  and  $B$  are points on the real line was addressed in [15]. They provided an  $O(N)$  time algorithm, assuming that the points are already sorted, and asked whether the minimum-weight many-to-many matching problem in the plane could be solved in  $o(N^3)$ . The question has been open until a recent work [16] gave an  $O(N^2 \text{poly}(\log N))$  time algorithm that reduces the problem to an equivalent perfect matching problem.

The existence of the ornamented parts of a melody justifies the use of a many-to-many matching to compute melodic similarity because this matching is more versatile and can match several notes of either melody with a note of the other. Figure 1 shows a many-to-many matching between two excerpts of melodies to be compared via melodic similarity. The upper melody is less ornamented, and then a note could be matched with several of the more ornate melody. Unfortunately, computing the minimum-weight many-to-many matching is time consuming and, as a consequence, it is not very useful for online music retrieval systems [17]. Thus, this paper focuses on some geometric constrained versions of the minimum-weight many-to-many matching in 2D that can be solved more efficiently.

**Fig. 1** Two pieces of music (extracted from [12]) and their representations as point sets in 2D, along with the optimum many-to-many matching used for computing the melodic similarity. In the ornamented parts (on the right side), one note is matched to several



The constraints are derived from the assignment rules that define the melodic matching. It might make sense to assign a note to one or more notes located in its vicinity when both melodies have the same tempo. For example, a point of a melody could be matched only to the first point to the left or to the right in the other melody. Different similarity measures can be defined in melody-based MIR research, see for example [18]. On the other hand, for computational modeling of melodic similarity, the measures must be tested against human perception. We propose in this paper three geometric measures with constraints based on vicinity. The evaluation on their suitability for modeling the perceived melodic similarity through human judgements is out of the scope of this paper. Two examples where this type of study is carried out for flamenco cante and Indian art music are [19, 20], respectively. At this point, we would like to point out that, although the problems introduced here are inspired by MIR and, the metrics could be used to compare (among other things) melodic shapes, they are interesting in their own right in the area of discrete optimization. In this paper, we show that for some specific graphs  $\mathcal{G} = (R \cup B, E)$ , where  $E$  is the set of edges connecting permitted pairings, the minimum-weight many-to-many matching can be solved in linear time.

The remainder of this paper is organized as follows. Section 2 provides a detailed description of the problems. Sections 4, 3 and 5 show the algorithms. Finally, in Sect. 6 we present our conclusions and provide suggestions for future work.

## 2 The constrained problems

Let  $B = \{b_1, \dots, b_n\}$  and  $R = \{r_1, \dots, r_m\}$  be sets of blue and red points, respectively, in the time-pitch plane. In the following, we use the notation  $n + m$  instead of  $N$  since, in general, the computation time will be a function of  $n$  and  $m$ . We assume that no two points from  $R$  (or  $B$ ) lie on a common vertical line since this happens in applications as melodic similarity for singing voice, where only a single note (point in the time-pitch plane) is emitted at a given time. Thus,  $t(b_1) < t(b_2) < \dots < t(b_n)$  and  $t(r_1) < t(r_2) < \dots < t(r_m)$ , where  $t(p)$  denotes the time coordinate (abscissa) of a point  $p \in R \cup B$ . We also assume that the matching cost is the sum of the Manhattan distances between paired points. However, the proposed algorithms can be applied to any metric.

The variants studied in this paper come from restricting the allowed pairings. Essentially, we limit the matching rule by considering, for example, some monotony properties with respect to time. Let  $r \in R$  and  $b \in B$ . In this constrained setting, we consider the following problems:

**Problem 2.1** *t*-NEAREST MANY-TO-MANY MATCHING. Given  $R$  and  $B$ , find a minimum-weight many-to-many matching  $\mathcal{M}$  such that for every pair  $(r, b)$  in  $\mathcal{M}$ ,  $r$  is the  $t$ -nearest red point of  $b$  or,  $b$  is the  $t$ -nearest blue point of  $r$ , where the  $t$ -nearest of a point  $p$  is the closest point to  $p$  of the other color with respect to the time.

**Problem 2.2** *t*-MONOTONE MANY-TO-MANY MATCHING. Given  $R$  and  $B$ , find a minimum-weight many-to-many matching  $\mathcal{M}$  such that for every pair  $(r, b) \in \mathcal{M}$ ,  $r$  is the first red point to the left or to the right of  $b$ ; or  $b$  is the first blue point to the left or to the right of  $r$ .

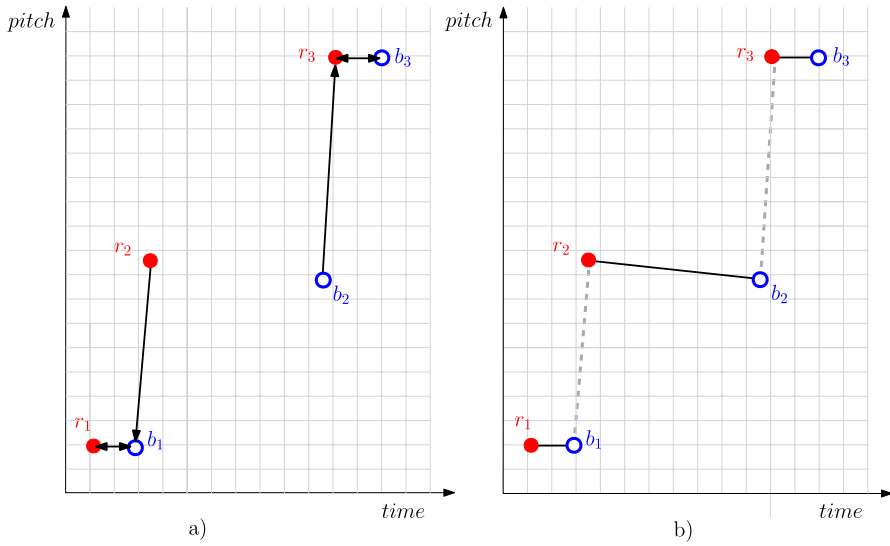
**Problem 2.3** MANY-TO-MANY MATCHING IN A TREE. Given a bichromatic weighted tree where every edge connects vertices of different color, compute a many-to-many matching of minimum cost in the tree.

From now on, in Problems 2.1 and 2.2 we call a segment connecting a pair  $(r, b)$  as a *feasible* edge for the matching. Note that a feasible edge for Problem 2.1 is also a feasible edge for Problem 2.2. The converse is not true. In Problem 2.3 a feasible edge is an edge of the tree. Note that the problems above can be stated in the graph theory framework as follows: Let  $\mathcal{G}$  be an undirected graph with node set  $R \cup B$  and edge set the feasible edges, find a minimum weight edge cover in  $\mathcal{G}$ . The weights or costs of the edges of  $\mathcal{G}$  correspond to the Manhattan distance between the ending points. In the following, let  $d(p, p')$  denote the Manhattan distance between the points  $p, p' \in R \cup B$ .

In Problems 2.1 and 2.2,  $\mathcal{G}$  is a special graph that we will describe in the next sections and, in Problem 2.3,  $\mathcal{G}$  is a generic tree.

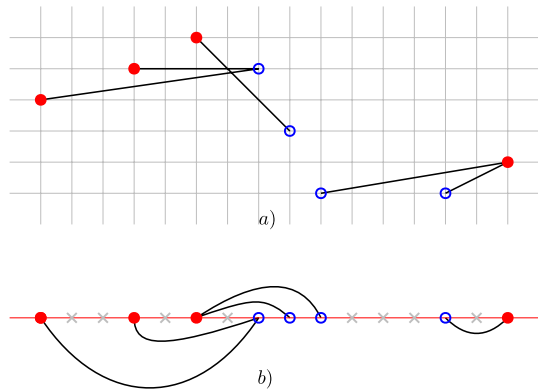
Figure 2 shows optimal solutions (black edges) for Problems 2.1 and 2.2 considering the same sets of points  $R$  and  $B$ . Since the constraints of Problem 2.2 are less stringent than those of Problem 2.1, the cost of an optimal solution for Problem 2.2 is guaranteed to be no larger for Problem 2.1.

It is worth mentioning that finding an optimal constrained many-to-many matching for sets of points on the plane cannot be solved by projecting the set of bicolored points to the *time*-axis, solving the optimal many-to-many matching on these points, and then lifting the solution thus obtained to the original point set. In the example illustrated in Fig. 3a, we show a bicolored point set such that the optimal solutions corresponding to Problems 2.1 and 2.2 match. However, the solution obtained by projecting the points onto the *time*-axis induces a many-to-many matching in the original point set that is not optimal; see Fig. 3b. The same assertion remains valid when we project onto the pitch-axis.



**Fig. 2** **a** Optimal matching for Problem 2.1, **b** Optimal matching for Problem 2.2. Dashed edges represent feasible edges that do not belong to the optimal solution

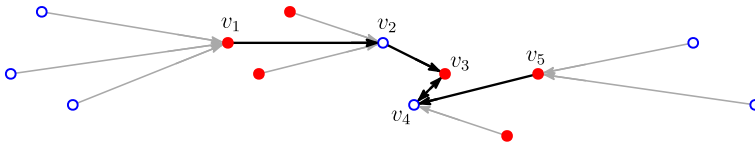
**Fig. 3** **a** Optimal solution for Problems 2.1 and 2.2, **b** Optimal solution for the 1D problem studied in [15]



### 3 t-nearest many-to-many matching

Let  $\mathcal{G} = (R \cup B, E)$  be the graph whose edges are the feasible edges of a  $t$ -nearest many-to-many matching in  $R \cup B$ . The main observation to efficiently solve Problem 2.1 is the following result that uses a special type of tree. A *caterpillar* is a tree in which all non-leaf vertices lie on a single path. The path, obtained by deleting the leaves of a caterpillar, is called the *spine*.

**Lemma 3.1** *Let  $\mathcal{G}$  be the graph composed by all the feasible edges of a  $t$ -nearest matching. Then  $\mathcal{G}$  is a forest. Furthermore, all the connected components of  $\mathcal{G}$  are caterpillars.*



**Fig. 4** An arrow from  $v_i$  to  $v_j$  indicates that  $v_j$  is the  $t$ -nearest point of  $v_i$ . The spine of the caterpillar is darker

**Proof** Suppose that  $\mathcal{G}$  has  $h$  connected components. Let  $G$  be a connected component of  $\mathcal{G}$  with  $s$  vertices. For each vertex in  $G$ , the  $t$ -nearest matching induces an edge, therefore  $G$  has at most  $s$  edges. On the other hand, the edge of the  $t$ -nearest bichromatic pair in  $G$ , that is, the edge connecting two points of different colors which are mutually the closest ones with respect to the time, is counted twice. Then  $G$  has at most  $(s - 1)$  edges. Since  $G$  is connected, it must be a tree.

Now, we will prove that  $G$  is a caterpillar, i.e. every vertex in  $G$  with degree 2 or more, lies on a single path (which is the spine of the caterpillar). The vertices of  $G$  can be separated, from left to right, into maximal monochromatic slabs  $G_0, G_1, G_2, \dots$  alternating colors (e.g  $G_0$  is blue,  $G_1$  is red, and so on). The  $t$ -nearest pair of vertices in  $G$  connects the rightmost vertex of a slab  $G_i$  with the leftmost vertex of the next slab  $G_{i+1}$ . Since  $G$  is a tree, it is easy to check that all the vertices in  $G_j$ ,  $j \leq i$ , are connected with the leftmost vertex in  $G_{j+1}$ . Analogously, all the vertices in  $G_j$ ,  $j > i$ , are connected with the rightmost vertex in  $G_{j-1}$ . Notice that the leftmost vertices of  $G_1, G_2, \dots, G_{i+1}$  and the rightmost vertices of  $G_i, G_{i+1}, \dots$  are the vertices with degree 2 or more, and they conform the spine of the caterpillar. See Fig. 4 for an illustration of the proof.  $\square$

Let  $\mathcal{G}$  be the caterpillar forest induced by the set of feasible edges in the  $t$ -nearest matching on  $R \cup B$ . A minimum  $t$ -nearest many-to-many matching in  $\mathcal{G}$  is the union of the optimal matching in each of its connected components. Therefore, we solve the problem for each tree of  $\mathcal{G}$ .

Given  $G$ , a caterpillar tree of  $\mathcal{G}$ , let  $S = (v_1, \dots, v_k)$  be the vertices along the spine of  $G$ , see Fig. 4. Notice that the edges incident on leaves are always in the minimum  $t$ -nearest matching. Thus, the key question is which edges of the spine  $(v_i, v_{i+1})$  should be taken to build the minimum  $t$ -nearest matching.

We say that a vertex  $v_i$  of  $S$  is *covered* if it is connected to leaves, and we denote by  $w(i)$  the sum of weights corresponding to edges connecting  $v_i$  with leaves. Note that, both endpoints of  $S$ ,  $v_1$  and  $v_k$ , are always covered. For non-covered vertices, we set  $w(i) = 0$ . With all the above, The following result is straightforward and establishes a recurrence that allows to compute the minimum weight  $t$ -nearest matching on a caterpillar  $G$  using dynamic programming.

**Lemma 3.2** For all,  $1 \leq i \leq k$ , let  $c(i)$  denote the cost of the minimum  $t$ -nearest many-to-many matching in the sub-caterpillar formed by  $v_1, v_2, \dots, v_i$  (and their leaves). Set  $c(0) = 0$ . Then,  $c(1) = w(1)$ , and for  $i = 2, \dots, k$  we have:

$$c(i) = \begin{cases} w(i) + \min \{c(i - 1), c(i - 2) + w(i - 1) + d(v_{i-1}, v_i)\} & \text{if } v_i \text{ is covered, and} \\ c(i - 1) + d(v_{i-1}, v_i) & \text{otherwise.} \end{cases}$$

Another alternative formula is:

$$c(i) = \begin{cases} w(i) + \min \{c(i - 1), c(i - 2) + d(v_{i-1}, v_i)\} & \text{if } v_i \text{ is covered and } v_{i-1} \text{ isn't,} \\ w(i) + c(i - 1) & \text{if } v_i \text{ and } v_{i-1} \text{ are covered, and} \\ c(i - 1) + d(v_{i-1}, v_i) & \text{otherwise.} \end{cases}$$

Note that, after progressively computing  $c(1), c(2), \dots, c(k)$ , it will result that the cost of the minimum  $t$ -nearest matching on caterpillar  $G$  is  $c(k)$ . Now we are ready to set up the final result of this section.

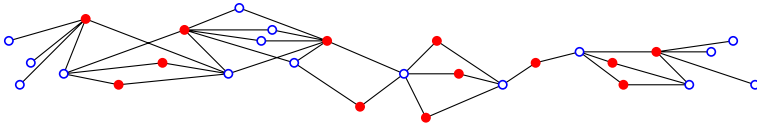
**Theorem 3.3** The  $T$ -NEAREST MANY-TO-MANY MATCHING PROBLEM can be solved in  $O(n + m)$  time.

**Proof** In a first sweep from left to right, we can determine the red/blue slabs of points. In a second sweep, we can determine the feasible edges and the caterpillars (connected components) of the graph  $\mathcal{G}$ . Additionally, in this second sweep, for every point  $v_i$  forming part of some spine, we can compute  $w(i)$ . This first phase of the algorithm takes  $O(n + m)$  time.

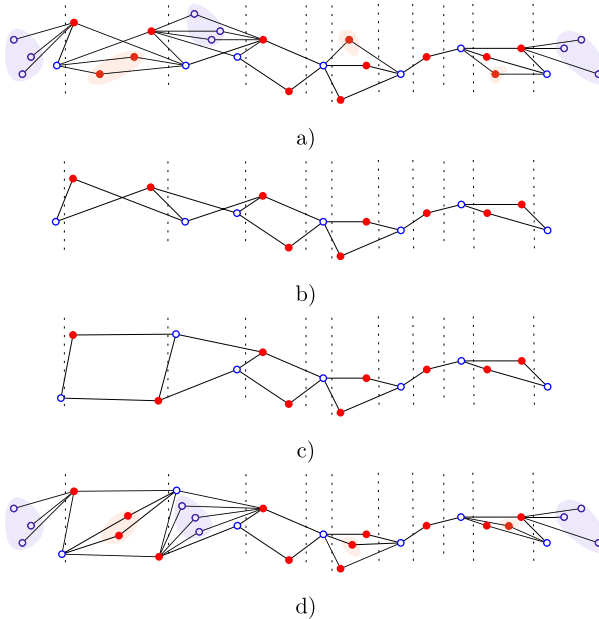
Now, focus on a single caterpillar  $G$  in  $\mathcal{G}$ . Suppose that  $G$  has a spine  $S = (v_1, \dots, v_k)$ . For all  $1 \leq i \leq k$ , let  $c(i)$  denote the cost of the minimum  $t$ -nearest many-to-many matching in the sub-caterpillar formed by  $v_1, v_2, \dots, v_i$  (and their leaves). Then  $c(1) = w(1)$  by Lemma 3.2. Now, we can compute  $c(i)$  progressively (in  $O(1)$  time) for all  $2 \leq i \leq k$  using the previously computed values  $c(i - 1)$  and  $c(i - 2)$ , and the recurrence established by Lemma 3.2. At the end,  $c(k)$  holds the cost of the minimum  $t$ -nearest many-to-many matching on  $G$  and compute it took  $O(k)$  time. Note that the set all the vertices forming some spine is a subset of  $R \cup B$ . Then, this phase takes  $O(n + m)$  time. The result follows.  $\square$

### 4 $t$ -Monotone many-to-many matching

Let  $\mathcal{G} = (R \cup B, E)$  be the graph, where the set  $E$  is formed by all the feasible edges for a  $t$ -monotone many-to-many matching in  $R \cup B$ . See Fig. 5 for an example. A remarkable property of  $\mathcal{G}$  is that it is a connected graph that can be drawn on the plane without intersections. Before proving this property, we introduce some terminology. First, separate the set of points  $R \cup B$  into maximal monochromatic slabs



**Fig. 5** The edges of  $\mathcal{G} = (R \cup B, E)$  are the feasible edges of a  $t$ -monotone many-to-many matching in  $R \cup B$ . Note that  $\mathcal{G}$  contains cycles



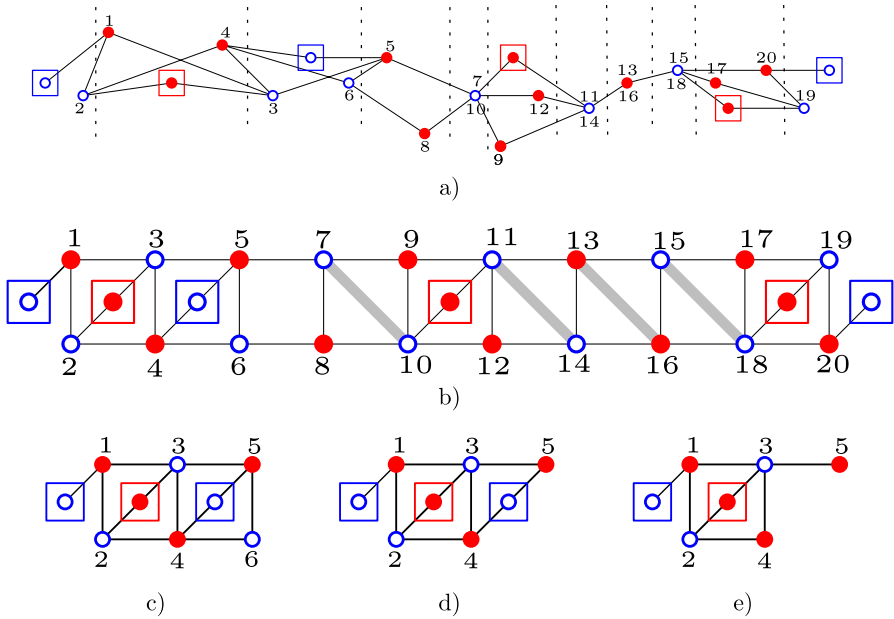
**Fig. 6** **a** Vertical dotted lines separate maximal monochromatic slabs. The shaded regions contain the *internal* points. **b**  $\mathcal{G}'$ , the subgraph of  $\mathcal{G}$  induced by the non-internal points of  $R \cup B$ . **c** Planar drawing of  $\mathcal{G}'$ ,  $\Gamma$ . **d** Planar drawing of  $\mathcal{G}$

using vertical lines, as in Fig. 6a. In every slab, we call *internal* points the vertices between the left- and right-most point of the slab. Note that for the special case of the first (resp. last) slab, we consider that all the points are internal except the right (resp. left) most one. The rest of the vertices are named *external*.

**Lemma 4.1** *Let  $\mathcal{G} = (R \cup B, E)$  be the graph where  $E$  is composed by all the feasible edges for a  $t$ -monotone many-to-many matching. Then  $\mathcal{G}$  is a planar graph.*

**Proof** The idea of our strategy is to make a planar drawing of a subgraph of  $\mathcal{G}$ , and then add the rest of  $\mathcal{G}$  keeping planarity. Let  $\mathcal{G}'$  be the subgraph of  $\mathcal{G}$  induced by the external points of  $R \cup B$ , see Fig. 6b. Note that a planar drawing  $\Gamma$  of graph  $\mathcal{G}'$  can be easily done by ‘untangling’ every crossing from left to right. This will result on a connected graph which is sequence of convex quadrilaterals and paths, see Fig. 6c.





**Fig. 7** **a** Internal points of the slabs are represented by *little squares*. First and last vertex of the slabs have been numerated. **b**  $\mathcal{G}$  as a concatenation of boxes, a diamondback graph. **c** Graph until the second box. **d** Graph until the second wedge including internal points. **e** Graph until the second wedge without internal points

We construct the planar drawing of  $\mathcal{G}$  from  $\Gamma$  by placing each internal point on its corresponding slab, inside of the convex quadrilateral in  $\Gamma$ , as shown in Fig. 6d. Note that each quadrilateral  $Q$  has exactly two vertices that differ in color with the internal points within it. To complete the drawing, we join these corners with all the internal points in  $Q$  avoiding crossings.  $\square$

We now describe the structure inherent to  $\mathcal{G}$  that will be crucial to design our algorithm. For this, we numerate the external vertices in the following way. For the  $i$ -th separator vertical line, let  $2i - 1$  be the vertex to the right of the vertical line and  $2i$  be the vertex to the left of the vertical line, as illustrated in Fig. 7a. Note that, if the slab contains only one point, it receives two numbers. For the sake of simplicity, we have represented the internal points of a slab using a little square with a point inside using the corresponding color.

In Fig. 7b, a simplified planar diagram of the numerated structure in Fig. 7a is shown. Note that vertices with two assigned numbers (single-point slabs) are represented by two points connected with a bold-light segment. This structure is a

concatenation of *boxes*<sup>1</sup> that we call a *diamondback-snake graph* or *diamondback-graph*, for short,<sup>2</sup> which allows to build the minimum-weight  $t$ -monotone matching with a dynamic programming approach.

The key idea is that we can progressively compute, for  $i = 1, 2, \dots$ , the minimum  $t$ -monotone matching until the  $i$ -th box, that is, the box with left vertical edge on the  $i$ -th slab separator. Figure 7c shows the structure until the second box ( $i = 2$ ).

Let  $c(i)$  denote the cost of the minimum  $t$ -monotone matching until the  $i$ -th box. Let  $c^\circ(i)$  denote the cost of the minimum  $t$ -monotone partial matching until the  $i$ -th box where all the points are matched except the top-right vertex of the  $i$ -th box, labeled as  $2i + 1$ . Let  $c_\circ(i)$  denote the cost of the minimum  $t$ -monotone partial matching until the  $i$ -th box where all the points are matched except the bottom-right vertex of the  $i$ -th box, labeled as  $2i + 2$ . And finally, let  $c^\circ_\circ(i)$  denote the cost of the minimum  $t$ -monotone partial matching until the  $i$ -th box where all the points are matched except the top- and bottom-right vertices of the  $i$ -th box ( $2i + 1$  and  $2i + 2$ ). For the sake of simplicity, we refer by  $c$ -costs until the  $i$ -th box to the values  $c(i)$ ,  $c^\circ(i)$ ,  $c_\circ(i)$  and  $c^\circ_\circ(i)$ .

For the base case, we introduce an artificial 0-th box ending at the edge  $(1, 2)$ . Note that all the internal points of the first slab (if there are any) must be matched with 1 and taking into account the combinatorics of using or not the vertical edge  $(1, 2)$ , the computation of the  $c$ -costs until the 0-th box can be done in  $O(p_0)$  time, where  $p_0$  denotes the number of internal points in the first slab. Denoting by  $I$  the sum of the weights of the edges connecting 1 with the internal points in the first slab, we have:

$$\begin{aligned} c(0) &= I + d(1, 2) \\ c^\circ(0) &= \infty \quad (\text{it is impossible matching 2 without matching 1}) \\ c_\circ(0) &= \begin{cases} I & \text{if there are internal points in the first slab,} \\ \infty & \text{otherwise} \end{cases} \\ c^\circ_\circ(0) &= \begin{cases} 0 & \text{if there is not internal points in the first slab,} \\ \infty & \text{otherwise} \end{cases} \end{aligned} \quad (1)$$

In the rest of this section we show how to calculate the  $c$ -costs until the  $i$ -th box from the  $c$ -costs until the  $(i - 1)$ -th box.

The easy case is when the  $i$ -th box corresponds to a single-point slab:

$$\begin{aligned} c(i) &= \min\{c^\circ(i - 1), c(i - 1)\} + d(2i + 1, 2i + 2) \\ c^\circ(i) &= c(i - 1) \\ c_\circ(i) &= \infty \quad (\text{it does not make sense}) \\ c^\circ_\circ(i) &= c^\circ(i - 1). \end{aligned} \quad (2)$$

<sup>1</sup> A box is composed by a quadrilateral with vertices in alternating colors and the corresponding internal points. Note that the vertical lines of the boxes match the slab separator vertical lines.

<sup>2</sup> <https://www.nationalgeographic.com/animals/reptiles/facts/eastern-diamondback-rattlesnake>.

If the  $i$ -th box is not single-point slab, then denote the  $i$ -th *wedge* as the triangle corresponding the top-left half of the  $i$ -th box. Additionally, to simplify the formulae, we separate the analysis of the wedges on: considering the internal points of the box and without the internal points of the box, as shown in Fig. 7d, e, respectively.

Considering the wedge without internal points (Fig. 7e), we introduce the following notation: Let  $w(i)$  denote the cost of the minimum  $t$ -monotone matching until the  $i$ -th wedge. Let  $w^\circ(i)$  denote the cost of the minimum  $t$ -monotone partial matching until the  $i$ -th wedge where all the points are matched except the top-right vertex of the  $i$ -th box ( $2i + 1$ ). Let  $w_\circ(i)$  denote the cost of the minimum  $t$ -monotone partial matching until the  $i$ -th wedge where all the points are matched except the bottom-left vertex of the  $i$ -th box ( $2i$ ). And finally, let  $w_\circ^\circ(i)$  denote the cost of the minimum  $t$ -monotone partial matching until the  $i$ -th wedge where all the points are matched except the top-right and bottom-left vertices of the  $i$ -th box ( $2i + 1$  and  $2i$ ). By simplicity, we refer by  $w$ -costs until the  $i$ -th wedge to the values  $w(i)$ ,  $w^\circ(i)$ ,  $w_\circ(i)$  and  $w_\circ^\circ(i)$ . Taking into account the combinatorics of using the top edge of the  $i$ -th box ( $2i + 1, 2i - 1$ ), we can compute the  $w$ -costs until the  $i$ -th wedge from the  $c$ -costs until the previous box as follows:

$$\begin{aligned}
 w(i) &= c(i - 1) + d(2i + 1, 2i - 1) \\
 w^\circ(i) &= c(i - 1) \\
 w_\circ(i) &= \min\{c_\circ(i - 1), c_\circ^\circ(i - 1)\} + d(2i + 1, 2i - 1) \\
 w_\circ^\circ(i) &= c_\circ(i - 1)
 \end{aligned}
 \tag{3}$$

Now, considering the wedge with its internal points (Fig. 7d), we denote the  $\hat{w}$ -costs until the  $i$ -th wedge as the values  $\hat{w}(i)$ ,  $\hat{w}^\circ(i)$ ,  $\hat{w}_\circ(i)$  and  $\hat{w}_\circ^\circ(i)$ , analogous to  $w(i)$ ,  $w^\circ(i)$ ,  $w_\circ(i)$  and  $w_\circ^\circ(i)$ , respectively, but considering that all the internal points in the  $i$ -th box (if there is any) are matched to the top-right or the left-bottom corner of the  $i$ -th box.

To compute the  $\hat{w}$ -costs until the  $i$ -th wedge, we need to introduce some additional notation. Let  $I(i)$  be the cost of assigning the internal points in the  $i$ -th box in a greedy manner, i.e., assigning each point to its nearest neighbor. Let  $I \downarrow$  (resp.  $I \uparrow$ ) denote the cost of assigning all the internal points to the left-bottom (resp. top-right) vertex of the  $i$ -th box. Let  $I \updownarrow(i)$  denote the cost of the assignment with minimum cost such that the top-right and the left-bottom vertices of the  $i$ -th box are both covered by the set of internal points. If the set of internal points contains a single point then it is matched to both corners (top-right and left-bottom); if it contains more than one point, then they are assigned in a greedy manner except maybe one point  $q$  which is forcefully switched from one corner to the other to fulfill the covering of both corners ( $q$  is the point with minimum cost of switching from one corner to the other). Note that  $I(i) \leq \min\{I \uparrow(i), I \downarrow(i), I \updownarrow(i)\}$  and, in case that making a greedy assignment, both corners result covered:  $I(i) = I \updownarrow(i)$ .

We are ready to establish the formulae for  $\hat{w}$ . If the  $i$ -th box does not contain internal points then the  $\hat{w}$ -costs coincide with the  $w$ -costs. If the box does contain internal points then the  $\hat{w}$ -costs can be computed as follows:

$$\begin{aligned}
\hat{w}(i) &= \min \begin{cases} w(i) + I(i) \\ w^\circ(i) + \min \{I \downarrow(i), I \uparrow(i)\} \\ w_\circ(i) + \min \{I \downarrow(i), I \uparrow(i)\} \\ w_\circ^\circ(i) + I \downarrow(i) \end{cases} \\
\hat{w}^\circ(i) &= \min \{w^\circ(i), w_\circ^\circ(i)\} + I \downarrow(i) \\
\hat{w}_\circ(i) &= \min \{w_\circ(i), w_\circ^\circ(i)\} + I \uparrow(i) \\
\hat{w}_\circ^\circ(i) &= \infty
\end{aligned} \tag{4}$$

Finally, having computed the  $\hat{w}$ -costs until the  $i$ -th wedge, we only need to consider the combinatorics of using the vertical right edge  $(2i + 1, 2i + 2)$  or the horizontal bottom edge  $(2i, 2i + 2)$  of the  $i$ -th box to compute the  $c$ -costs until here:

$$\begin{aligned}
c(i) &= \min \begin{cases} \hat{w}(i) + \min \{d(2i, 2i + 2), d(2i + 1, 2i + 2)\} \\ \hat{w}^\circ(i) + d(2i + 1, 2i + 2) \\ \hat{w}_\circ(i) + d(2i, 2i + 2) \\ \hat{w}_\circ^\circ(i) + d(2i, 2i + 2) + d(2i + 1, 2i + 2) \end{cases} \\
c^\circ(i) &= \min \{\hat{w}^\circ(i), \hat{w}_\circ^\circ(i)\} + d(2i, 2i + 2) \\
c_\circ(i) &= \hat{w}(i) \\
c_\circ^\circ(i) &= \hat{w}^\circ(i)
\end{aligned} \tag{5}$$

Equations on (1), (2), (3), (4) and (5) establish a recurrence to compute progressively via dynamic programming the  $c$ -costs until the  $i$ -th box. This approach leads us to the following result:

**Theorem 4.2** *The  $t$ -MONOTONE MANY-TO-MANY MATCHING PROBLEM can be solved in  $O(n + m)$  time. Furthermore, the cost of the minimum  $t$ -monotone many-to-many matching is  $c(k)$  if the last slab does not contain internal points and,  $\min \{c(k), c_\circ(k)\} + J$ , otherwise, where  $k$  is the number of boxes and  $J$  is the cost of assigning the internal points of the last slab (to the bottom-right vertex of the last box).*

**Proof** In a first sweep, from left to right, we can determine the red/blue slabs. In a second sweep we can establish the feasible edges, the internal and external points, and the boxes with their vertices. With one more sweep we can compute the values  $I(i)$ ,  $I \downarrow(i)$ ,  $I \uparrow(i)$  and  $I \downarrow(i)$  for every box with internal points.

All these operations can be performed in  $O(n + m)$  time.

In  $O(1)$  time, we compute the  $c$ -costs of the artificial 0-th box by following the formula (1). After that, we can progressively compute the  $c$ -costs until the  $i$ -th box for  $1 \leq i \leq k$ , where  $k$  is the number of boxes in the diagram. If the  $i$ -th box is a single-point box, the associated  $c$ -costs can be computed in  $O(1)$  time using (2). If it is not, then compute the associated  $w$ -costs following (3) in  $O(1)$  time. Then, using these values, compute the  $\hat{w}$ -costs associated to the  $i$ -th box in  $O(1)$  time using (4), and finally, use (5) to compute the  $c$ -costs until the  $i$ -th box using the already

computed  $\hat{w}$ -costs. It will take  $O(k)$  time, that is,  $O(n + m)$  time, to compute the values  $c(k)$  and  $c_o(k)$  (needed to apply the formula stated in the claim).

Finally, computing the cost of assigning the internal points of the last slab (to the bottom-right vertex of the last box) takes  $O(p_{k+1})$  time where  $p_{k+1}$  denotes the cardinality on the set of internal points in the last slab. Thus, the result follows.  $\square$

### 5 Many-to-many matching in a tree

In the above problems, we considered that the set of feasible edges for the matching lie on a specific type of graph (a caterpillar forest or a diamondback graph). In this section, we assume that such a graph is a generic tree and, for the sake of simplicity, with a matching we refer to a many-to-many matching on the tree.

In this section we show how to solve Problem 2.3 in linear time. Considering that the set of feasible edges in Problem 2.1 conforms a forest (Lemma 3.1), the approach of this section can be applied to solve Problem 2.1 in linear time as well. However, procedure shown in Sect. 3 use some nice intrinsic properties (e.g., the connected components of the feasible edges graph are caterpillar) which do not hold for generic trees, allowing to build simpler recurrence formulae for Problem 2.1.

Let  $T$  be a bichromatic tree of size  $n + m$  that we assume to be rooted at a node  $r$ . Let  $v$  be an arbitrary node. Let  $T_v$  be the subtree of  $T$  rooted at  $v$ . Given a matching  $M$ , we denote by  $M|T_v$  the set of edges in  $M$  connecting two vertices of  $T_v$ . Note that  $v$  could be matched by  $M|T_v$  (i.e.,  $v$  is matched in  $M$ , at least, by one of its children in  $T$ ) or not (i.e.,  $v$  is matched in  $M$  by its parent in  $T$ ).

Let  $M_{v,0}$  denote the minimum matching in  $T_v \setminus \{v\}$  and let  $M_{v,1}$  denote the minimum matching in  $T_v$ . In the following, we will show how to compute  $M_{v,0}$  and  $M_{v,1}$  for every  $v$  by using the matching sets  $M_{w,0}$  and  $M_{w,1}$ , where  $w$  is a children of  $v$  in  $T$ .

From now on, assume that the children of a node  $v$  are in some sequential order. Let  $T_v^i$  denote the subtree of  $T$  formed by  $v$  and the subtrees rooted at the first  $i$  children of  $v$ . Extending the previous notation, let  $M_{v,0}^i$  and  $M_{v,1}^i$  denote the minimum matching in  $T_v^i \setminus \{v\}$  and  $T_v^i$ , respectively. Following this notation,  $T_v = T_v^c$ , where  $c$  is the number of children of  $v$  in  $T$ .

It is easy to prove by contradiction the following result:

**Lemma 5.1** *Let  $M$  be a minimum matching. Let  $v$  be an arbitrary node with  $c$  children ( $c = 0$  if  $v$  is a leaf). If  $M|T_v^i$  (with  $1 \leq i \leq c$ ) matches  $v$  then there is no better matching than  $M|T_v^i$  in  $T_v^i$ . If  $M|T_v^i$  does not match  $v$ , then there is no better matching than  $M|T_v^i$  in the forest  $T_v^i \setminus \{v\}$ .*

Let  $C_{v,0}^i$  and  $C_{v,1}^i$  denote the cost of a minimum matching in  $T_v^i \setminus \{v\}$  and  $T_v^i$ , respectively. Set  $C_{v,0} = C_{v,0}^c$  and  $C_{v,1} = C_{v,1}^c$ , where  $c$  denotes the number of children of  $v$ . Note that  $C_{r,1}$  holds the cost of the minimum matching in  $T$ .

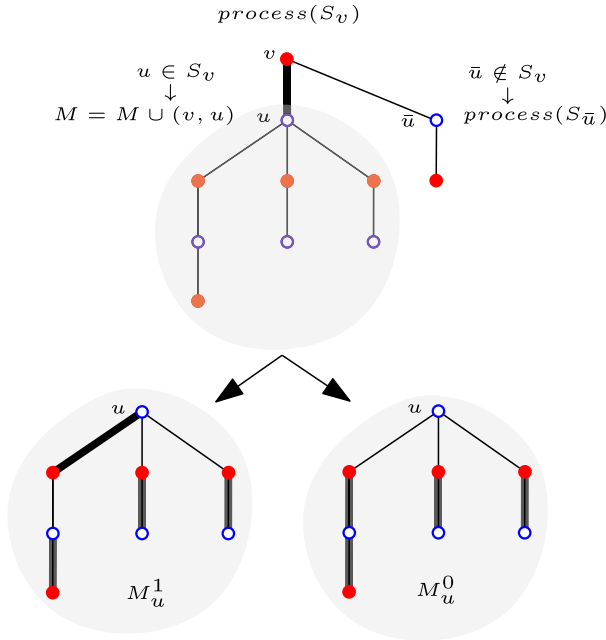


Fig. 8 Illustration of the subroutine  $process(S_v)$

**Lemma 5.2** *Let  $v$  be a node in  $T$  and let  $c$  denote the number of children of  $v$  in  $T$ . The values  $C_{v,0}$  and  $C_{v,1}$  can be computed in  $O(c)$  time in a bottom-up fashion using the values  $C_{u,0}$  and  $C_{u,1}$  of each node  $u$ , child of  $v$  in  $T$ .*

**Proof** Starting by the base case:  $v$  is a leaf and we have that  $C_{v,0} = 0$ , and  $C_{v,1} = \infty$  because  $T_v$  is formed by a single node and it does not allow a matching of  $v$ . In case  $v$  is an inner node, we compute  $C_{v,0}^i$  and  $C_{v,1}^i$  ( $1 \leq i \leq c$ ) in a left-right fashion as follows.

Let  $u$  be the  $i$ -th child of  $v$  in a left-right ordering and denote by  $w(v, u)$  the weight of the edge connecting  $v$  and  $u$ . The following formulae can be easily deduced:

- If  $i = 1$  then:

$$\begin{aligned} C_{v,0}^1 &= C_{u,1} \\ C_{v,1}^1 &= \min\{C_{u,1}, C_{u,0}\} + \omega(v, u). \end{aligned} \tag{6}$$

- If  $i > 1$  then:

$$\begin{aligned} C_{v,0}^i &= C_{v,0}^{(i-1)} + C_{u,1} \\ C_{v,1}^i &= \min \left\{ \begin{array}{l} \min\{C_{v,0}^{(i-1)}, C_{v,1}^{(i-1)}\} + \omega(v, u) + \min\{C_{u,0}, C_{u,1}\} \\ C_{v,1}^{(i-1)} + C_{u,1} \end{array} \right\} \end{aligned} \tag{7}$$

The formulae above allow us to use a linear-time dynamic programming algorithm to find the cost of a minimum matching in the subtree in  $O(c)$  time, where  $c$  is the number of children of  $v$ .  $\square$

To explicitly find that matching, for every node  $v$ , we maintain a set  $S_v$  containing all the children of  $v$  matched by  $v$  in the minimum matching of  $T_v$ . Note that if  $u$  is the  $i$ -th child of  $v$  then  $u \in S_v$  if and only if:

$$C_{v,1}^i = \min \left\{ C_{v,0}^{(i-1)}, C_{v,1}^{(i-1)} \right\} + \omega(v, u) + \min \{ C_{u,0}, C_{u,1} \} \quad (\text{see equation (7)}).$$

Thus, an optimal matching  $M$  could be constructed with the following procedure: We start by  $M = \emptyset$ . For a node  $v$  of  $T$ , we call  $\text{process}(S_v)$  the following subroutine (Fig. 8).

For all children  $u$  of  $v$ :

1. if  $u \notin S_v$ , then call  $\text{process}(S_u)$ ,
2. if  $u \in S_v$ , set  $M = M \cup \{(v, u)\}$  and
  - if  $C_{u,1} < C_{u,0}$ , then call  $\text{process}(S_u)$ ,
  - if  $C_{u,1} \geq C_{u,0}$ , then, for every  $u'$  child of  $u$ , call  $\text{process}(S_{u'})$ .

Applying  $\text{process}(S_r)$ , where  $r$  is the root of  $T$ , all the vertices are visited once and a set of edges corresponding to a minimum matching is obtained. Thus, we arrive to the last result of this work:

**Theorem 5.3** *The MANY-TO-MANY MATCHING IN A TREE PROBLEM can be solved in  $O(n + m)$  time.*

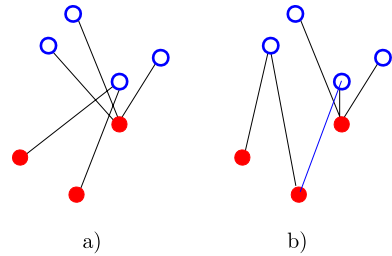
## 6 Conclusions and one open problem

This paper presents some constrained versions of an optimization problem that has been considered both in computational geometry and graph theory. For example, the problem has been studied in a geometric setting inspired by computational studies in musical theory. Given a bichromatic point set  $S$  in the plane, the many-to-many point matching on  $S$  looks for pairing up each point of one color to one or more points of the other color and vice versa so that the sum of distances between the paired points is minimized.

On the other hand, the problem can also be defined in a graph theoretical framework when a complete bipartite graph  $\mathcal{G}$  is given and the goal is to find a subset of edges of  $\mathcal{G}$  with minimum total weight such that each vertex of  $\mathcal{G}$  is an endpoint of some edge in the subset. This is named the minimum-weight bipartite edge cover problem.

Since the problem is time consuming in the plane, we showed that some constrained versions can be solved in linear time if the point set is given ordered from

**Fig. 9** **a** A many-to-many matching. **b** A merged many-to-many matching



left to right. The constrained versions appear, for example, in the area of melodic similarity, where the melodic contour is defined by a set of ordered notes in the pitch-time plane. Our restrictions limit the geometric matching rule by considering some monotony properties with respect to time, which leads to the set of feasible edges for the matching being in a specific type of graph. These restrictions make sense in musical similarity where the alignment of one note must not be to another far away in time.

We leave as an open problem whether the following variant of the many-to-many matching problem can also be solved in linear time.

**Problem 6.1** MERGED MANY-TO-MANY MATCHING PROBLEM. Given  $R$  and  $B$ , find a minimum-weight many-to-many matching such that if  $b_i$  is matched with  $r_j$  then  $b_{i+1}$  is matched with  $r_{j'}$  such that  $j' \geq j$ . Analogously, if  $r_j$  is matched with  $b_i$  then  $r_{j+1}$  is matched with  $b_{i'}$  such that  $i' \geq i$ .

Figure 9 shows the difference between a non-restricted many-to-many matching and a merged many-to-many matching. The question is to know whether this problem can be solved with a subquadratic algorithm.

We include here a quadratic-time approach. Let  $A[i, j]$  denote the best matching between the sequences  $b_1, \dots, b_i$  and  $r_1, \dots, r_j$ . It is easy to see (by definition of this constrained matching) that  $(b_1, r_1)$  and  $(b_i, r_j)$  are in the matching. We can also prove that:

$$A[1, 1] = \omega(b_1, r_1)$$

$$A[i, j] = \min\{A[i-1, j-1], A[i-1, j], A[i, j-1]\} + \omega(b_i, r_j).$$

where  $\omega(\cdot, \cdot)$  is the cost function. Note that using the formula above, having at hand the values  $A[i-1, j-1]$ ,  $A[i-1, j]$  and  $A[i, j-1]$ , the value of  $A[i, j]$  can be computed in  $O(1)$  time. Thus, a dynamic programming approach solves the problem in  $O(nm)$  time.

**Acknowledgements** Luis Evaristo Carballo, José Miguel Díaz-Báñez and Inmaculada Ventura are partially supported by European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement #734922 and Ministerio de Ciencia e Innovación CIN/AEI/10.13039/501100011033 (PID2020-114154RB-I00). Rodrigo A. Castro, Marco A. Heredia, Jorge Urrutia, and Francisco J. Zaragoza are partially supported by Sistema Nacional de Investigadores, Consejo Nacional de Ciencia y Tecnología, México. Jorge Urrutia is partially supported by PAPIIT IN105221 Programa de Apoyo a la Investigación e Innovación Tecnológica, UNAM, Mexico.



**Funding** Funding for open access publishing: Universidad de Sevilla/CBUA.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Byrd, D., Crawford, T.: Problems of music information retrieval in the real world. *Inf. Process. Manag.* **38**(2), 249–272 (2002)
2. Cornelis, O., Lesaffre, M., Moelants, D., Leman, M.: Access to ethnic music: advances and perspectives in content-based music information retrieval. *Signal Process.* **90**(4), 1008–1031 (2010)
3. Mora, J., Gómez, F., Gómez, E., Díaz-Báñez, J.M.: Melodic contour and mid-level global features applied to the analysis of flamenco cantes. *J. New Music Res.* **45**(2), 145–159 (2016)
4. Clifford, R., Christodoulakis, M., Crawford, T., Meredith, D., Wiggins, G.A.: A fast, randomised, maximal subset matching algorithm for document-level music retrieval. In: *ISMIR 2006, 7th International Conference on Music Information Retrieval*, Victoria, Canada, 8–12 October 2006, Proceedings, pp. 150–155 (2006)
5. Gudmundsson, J., Klein, O., Knauer, C., Smid, M.: Small Manhattan networks and algorithmic applications for the earth mover's distance. In: *Proceedings of the 23rd European Workshop on Computational Geometry*, pp. 174–177 (2007)
6. Wang, W., Zhang, G., Lu, J.: Collaborative filtering with entropy-driven user similarity in recommender systems. *Int. J. Intell. Syst.* **30**(8), 854–870 (2015). <https://doi.org/10.1002/int.21735>
7. Mokbel, B., Hasenfuss, A., Hammer, B.: Graph-based representation of symbolic musical data. In: *Graph-Based Representations in Pattern Recognition: 7th IAPR-TC-15 International Workshop, GbRPR 2009, Venice, Italy, May 26–28, 2009. Proceedings 7*, pp. 42–51. Springer, Berlin (2009)
8. Colannino, J., Toussaint, G.: An algorithm for computing the restriction scaffold assignment problem in computational biology. *Inf. Process. Lett.* **95**(4), 466–471 (2005). <https://doi.org/10.1016/j.ipl.2005.05.007>
9. Ben-Dor, A., Karp, R., Schwikowski, B., Shamir, R.: The restriction scaffold problem. *J. Comput. Biol. J. Comput. Mol. Cell Biol.* **10**(3–4), 385–398 (2003). <https://doi.org/10.1089/10665270360688084>
10. Toussaint, G.: The geometry of musical rhythm. In: Akiyama, J., Kano, M., Tan, X. (eds.) *Discrete and Computational Geometry*, pp. 198–212. Springer, Berlin (2005). [https://doi.org/10.1007/11589440\\_20](https://doi.org/10.1007/11589440_20)
11. Toussaint, G.T.: A comparison of rhythmic similarity measures. In: *Proceedings of Fifth International Conference on Music Information Retrieval*, Barcelona, Spain, 10–15 October, pp. 242–245 (2004)
12. Typke, R., Wiering, F., Veltkamp, R.C.: Transportation distances and human perception of melodic similarity. *Musicae Scientiae* **11**(1\_suppl), 153–181 (2007). <https://doi.org/10.1177/102986490701100107>
13. Eiter, T., Mannila, H.: Distance measures for point sets and their computation. *Acta Informatica* **34**(2), 109–133 (1997). <https://doi.org/10.1007/s002360050075>
14. Kuhn, H.W.: The Hungarian method for the assignment problem. *Naval Res. Logist. Q.* **2**(1–2), 83–97 (1955). <https://doi.org/10.1002/nav.3800020109>
15. Colannino, J., Damian, M., Hurtado, F., Langerman, S., Meijer, H., Ramaswami, S., Souvaine, D., Toussaint, G.: Efficient many-to-many point matching in one dimension. *Graphs Comb.* **23**(1), 169–178 (2007). <https://doi.org/10.1007/s00373-007-0714-3>

16. Bandyapadhyay, S., Maheshwari, A., Smid, M.: Exact and approximation algorithms for many-to-many point matching in the plane. In: Ahn, H.-K., Sadakane, K. (eds.) 32nd International Symposium on Algorithms and Computation (ISAAC 2021). Leibniz International Proceedings in Informatics (LIPIcs), vol. 212, pp. 44–14414. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2021). <https://doi.org/10.4230/LIPIcs.ISAAC.2021.44>
17. Lin, H.-J., Wu, H.-H., Kao, Y.-T.: Geometric measures of distance between two pitch contour sequences. *J. Comput.* **19**(2), 44–66 (2008). <https://doi.org/10.1016/j.jpl.2008.09.014>
18. Di Lorenzo, P., Di Maio, G.: The Hausdorff metric in the melody space: a new approach to melodic similarity. In: The 9th International Conference on Music Perception and Cognition, Alma Mater Studiorum University of Bologna, pp. 22–26 (2006)
19. Kroher, N., Gómez, E., Guastavino, C., Gómez, F., Bonada, J.: Computational models for perceived melodic similarity in a cappella flamenco singing. In: ISMIR, pp. 65–70 (2014). Citeseer

## Authors and Affiliations

L. E. Carballo<sup>1</sup> · R. A. Castro<sup>2</sup> · J. M. Díaz-Báñez<sup>1</sup> · M. A. Heredia<sup>2</sup> · J. Urrutia<sup>3</sup> · I. Ventura<sup>1</sup>  · F. J. Zaragoza<sup>2</sup>

✉ I. Ventura  
iventura@us.es

L. E. Carballo  
lcarballo@us.es

R. A. Castro  
racc@azc.uam.mx

J. M. Díaz-Báñez  
dbanez@us.es

M. A. Heredia  
hvma@azc.uam.mx

J. Urrutia  
urrutia@matem.unam.mx

F. J. Zaragoza  
franz@azc.uam.mx

<sup>1</sup> Departamento de Matemática Aplicada II, Universidad de Sevilla, Seville, Spain

<sup>2</sup> Departamento de Sistemas, Universidad Autónoma Metropolitana Azcapotzalco, Mexico City, Mexico

<sup>3</sup> Instituto de Matemáticas, Universidad Nacional Autónoma de México, Mexico City, Mexico

20. Ganguli, K.K., Rao, P.: On the perception of raga motifs by trained musicians. *J. Acoust. Soc. Am.* **145**(4), 2418–2434 (2019)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.