Acta Informatica 34, 109–133 (1997)

© Springer-Verlag 1997

Distance measures for point sets and their computation

Thomas Eiter[1], Heikki Mannila[2]

[1] Information Systems Department and Christian Doppler Lab for Expert Systems, Technical University of Vienna, Paniglgasse 16, A-1040 Vienna, Austria
[2] Department of Computer Science, University of Helsinki, P.O. Box 26, FIN-00014 Helsinki, Finland
(e-mail: eiter@dbai.tuwien.ac.at, mannila@cs.helsinki.fi)

Received: 1 July 1994 / 9 November 1995

Abstract. We consider the problem of measuring the similarity or distance between two finite sets of points in a metric space, and computing the measure. This problem has applications in, e.g., computational geometry, philosophy of science, updating or changing theories, and machine learning. We review some of the distance functions proposed in the literature, among them the minimum distance link measure, the surjection measure, and the fair surjection measure, and supply polynomial time algorithms for the computation of these measures. Furthermore, we introduce the minimum link measure, a new distance function which is more appealing than the other distance functions mentioned. We also present a polynomial time algorithm for computing this new measure.

We further address the issue of defining a metric on point sets. We present the metric infimum method that constructs a metric from any distance functions on point sets. In particular, the metric infimum of the minimum link measure is a quite intuitive. The computation of this measure is shown to be in NP for a broad class of instances; it is NP-hard for a natural problem class.

1 Introduction

A function $d : B \times B \rightarrow \Re^+$ is a *metric* on a nonempty set B, if for all $x, y, z \in B$ we have

1 $d(x, y) = 0$ if and only if $x = y$;
2 $d(x, y) = d(y, x)$
3 $d(x, z) \leq d(x, y) + d(y, z)$.

The function d is a *distance function* on B, if it satisfies (1) and (2).

In this paper we consider the following questions. (1) How can a distance function (or even metric) Δ on B be extended to a distance function or a metric d on the collection of all nonempty (finite) subsets of B? The extension condition we require is that for singletons, d agrees with Δ, that is, $d(\{x\}, \{y\}) = \Delta(x, y)$ for all x and y from B. (2) Algorithms for computing such extensions, which run in polynomial time if possible.

Questions of this type arise in several areas, such as cluster analysis [2, 20], computational geometry [1], philosophy of science [17, 18, 13, 15], updating and revising theories [6, 25], arbitration between theories [19], and machine learning [12, 11, 26].

The best-known metric between subsets of a metric space is the Hausdorff metric, defined as

$$d_h(S_1, S_2) = \max\{\max_{e \in S_1} \min_{f \in S_2} \Delta(e, f), \max_{e \in S_2} \min_{f \in S_1} \Delta(e, f)\}.$$

This metric is trivially computable in polynomial time, and it has some quite attractive properties. Unfortunately, it is not very well suited for some applications. The reason is that the Hausdorff distance does not take into account the overall structure of the point sets (see Fig. 1).

Several alternative distance functions between subsets have been proposed in philosophy of science [13], for the objective of measuring the distance between theories in a logical language $\mathscr{L}$. Here, the points of the metric space are the interpretations of $\mathscr{L}$, whose distance is measured by some metric Δ; for propositional $\mathscr{L}$, the Hamming distance between interpretations (i.e., the number of atoms on which they are different) is a natural choice. Each theory in $\mathscr{L}$ is identified with the set of its models, which thus is a set of points in the metric space.[1] This way, measuring the distance between two theories is abstracted to measuring the distance between two sets of points in a metric space. In the particular case where one theory describes categorically TRUTH (i.e., the true state of affairs), the distance value is taken for a quantitative extent of the truthlikeness of the other theory. Notice that measuring theory distance in this setting has suggestive applications for theory revision (change a theory into the closest possible one that meets the revision) and arbitration between theories.

An example can be drawn from the court domain (cf. [19]). Suppose that in a trial two testimonies (which can be represented as theories) are very similar while a third is much different. Given that the witnesses are independent of each other, the belief of the jury in the reliability of the third testimony will naturally decrease with growing distance to the other testimonies.

Another use of such distance functions between subsets is in the new area of knowledge discovery in databases [5, 16], where one often has to choose between or form clusters from different rules discovered from the data. The rules can be identified with the data points to which they apply, and then rule distance can be computed by using a distance function for subsets [21].

Particularly appealing among the distance functions between point sets that have been proposed in philosophy of science are the minimum distance measure, d_{md}, the surjection measure, d_s, and the fair surjection measure, d_{fs}. Intuitively, these measures compute the amount of distance covered when each point of one set is mapped to a point of the other set under some restrictions. Under d_{md}, each point of the one set is mapped to the point of the other set that is closest to it; for the surjection distance d_s, one considers all surjections from one set to the other and computes the length of the elements in the surjection; for the fair surjection distance d_{fs}, one additionally requires that the surjection has to be fair in the sense that it distributes the points as evenly as possible.

[1] Theories are thus not necessarily categorical, and reflect incomplete information about certain facts; they may leave open several possibilities for the true state of affairs, which is considered to be among the models.

While the properties and relationships of these measures have been considered, their computation was left open. In particular, it was unclear for some of these measures whether they can be computed efficiently if this is possible for the underlying distance function on points.

Another interesting issue is an appealing distance function on sets of points which satisfies the triangle inequality, i.e., a metric on the sets of points. Note that the distance measures d_{md}, d_s, and d_{fs} do not satisfy the triangle inequality.

The main results of this paper can be shortly summarized as follows.

- We propose a new measure, the *link measure* d_l, which is motivated by some intuitive problems with the other measures.
- We show that the measures d_{md}, d_s, d_{fs}, and d_l can be computed in polynomial time under very general assumptions. While this is trivial for d_{md}, the algorithms for the other measures use sophisticated matching and network flow techniques.
- We introduce the metric infimum method, a general construction that produces for a given distance function d on sets of points a refinement d^ω of d which is a metric and hence satisfies the triangle inequality. The intuition in the construction is to consider in computing the distance between two sets S_1 and S_2 all possible sequences of intermediate sets and take the minimum distance obtainable by using such a sequence. We show that applied to d_{md}, d_s, d_{fs} and d_l, this construction produces only two different metrics.
- We present an algorithm for computing the refined link measure d_l^ω which makes use of standard graph algorithms. This algorithm is not polynomial; we show, however, that a polynomial algorithm is not likely to exist by proving that computing d_l^ω is NP-hard. On the other hand, our results imply that for a broad class of instances, computing d_l^ω is NP-easy, i.e., possible in polynomial time with an oracle for some problem in NP (cf. [8]), and hence not much harder than the NP-complete problems.

The rest of the paper is organized as follows. After introducing some notation at the end of the introduction, we review in Sect. 2 measures for the distance of point sets derived from measures of theory distance from [13, 22, 14]. Furthermore, in this section we define the new minimum link measure d_l and compare it to the distance measures d_{md}, d_s, and d_{fs}. In Sect. 3 we analyze the structural properties of d_s, d_{fs}, and d_l, which are employed by our algorithms. Section 4 is devoted to the computation of the distance measures d_{md}, d_s, d_{fs}, and d_l, for which we describe polynomial time algorithms. In Sect. 5 we provide the metric infimum method and study properties of the metric produced from a distance function. In Sect. 6 we consider computing the d_l^ω measure and show NP-completeness of the associated decision problem. The concluding Sect. 7 states some open problems and issues for further work.

We start by defining some notation. We assume that B is finite; however, the constructions can be carried out also for infinite B under some conditions. We shall discuss this issue in Sect. 7. Elements of B are referred to as points. The set of all nonempty subsets of B is denoted by $\mathscr{P}_\emptyset(B)$. We use a special notation for the minimum distance of an element x from a set S:

$$\Delta_m(x, S) = min_{y \in S}\, \Delta(x, y).$$

We call each point $y \in S$ such that $\Delta(x, y) = \Delta_m(x, S)$ a *closest point of S for x*. We further denote by μ a function $\mu : B \times \mathscr{P}_\emptyset(B) \to B$ that for all x and nonempty S yields a closest point of S for x, i.e.

$$\mu(x, S) = y, \text{ such that } y \in S \text{ and } \Delta(x, y) = \Delta_m(x, S).$$

We assume that the reader knows about basic concepts of graph theory (cf. [3, 24]) and NP-completeness (cf. [8]).

Let the *union* of a family of graphs $\{G_i = (V_i, E_i) : 1 \leq i \leq n\}$ be the graph $(\bigcup_i V_i, \bigcup_i E_i)$. The union is *disjoint* if $V_i \cap V_j = \emptyset$, $1 \leq i < j \leq n$. In case that the G_i are weighted, i.e. $G_i = (V_i, E_i, w_i)$, the union $(\bigcup_i V_i, \bigcup_i E_i, \bigcup_i w_i)$ is only defined if the weight function $\bigcup_i w_i$ is well-defined.

We denote by $deg(v, G)$ the degree of vertex v in the graph G. A graph G is called a *line* iff it is a tree (i.e. a connected acyclic graph) on 2 vertices, and G is called a *star* iff it is a tree on > 2 vertices and exactly one vertex v has degree > 1, which is called the *center* of the star.

In our examples, we use as the base set B a finite fragment of the integral plane with, unless stated otherwise, the Manhattan metric, which is defined by

$$\Delta_M((x_1, y_1), (x_2, y_2)) = |x_1 - x_2| + |y_1 - y_2|.$$

2 Distance measures for point sets

In this section, we review some of the distance functions which have been proposed in the literature. A comparative analysis of these measures, from the viewpoint of philosophy of science, can be found in [13]. We further propose a new distance function which overcomes weaknesses of previously suggested distance functions in some situations.

Hausdorff distance (d_h) The *Hausdorff distance* d_h can be defined using the Δ_m notation as

$$d_h(S_1, S_2) = \max\{\max_{e \in S_1} \Delta_m(e, S_2), \max_{e \in S_2} \Delta_m(e, S_1)\}.$$

This function defines a distance function on $\mathscr{P}_\emptyset(B)$, which is a metric if Δ is a metric ([4]). The problem of computing the Hausdorff distance between geometric entities has been considered in the area of computational geometry (see, e.g., [9]).

The problem with d_h is that it is very sensitive to extreme points in the sets S_1 and S_2 (see Fig. 1.)

In fact, for every pair of sets S_1 and S_2 there are points $x_1 \in S_1$ and $x_2 \in S_2$ such that $d_h(S_1, S_2) = d_h(\{x_1\}, \{x_2\}) = \Delta(x_1, x_2)$. We are looking for a distance measure that attempts to combine information about the distances between the elements of S_1 and S_2. Therefore we do not consider d_h in the sequel.

Sum of minimum distances (d_{md}) The *sum of minimum distances* function d_{md} [13] is defined by

$$d_{md}(S_1, S_2) = \frac{1}{2}\left(\sum_{e \in S_1} \Delta_m(e, S_2) + \sum_{e \in S_2} \Delta_m(e, S_1)\right).$$

This function takes into account the distances between each element and the other set.

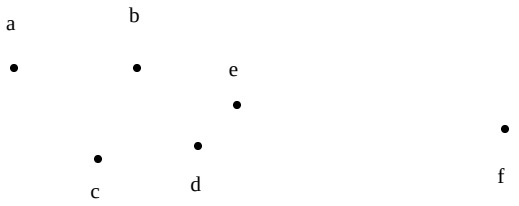


Fig. 1. Sets $\{a, b, c, d, e\}$ and $\{a\}$ are equally distant from $\{f\}$ according to the Hausdorff metric

Surjection distance (d_s) Oddie [14] suggested defining the distance between two sets by considering surjections that map the larger set to the smaller set. This leads to the following distance function:

$$d_s(S_1, S_2) = \min_{\eta} \sum_{(e_1,e_2)\in\eta} \Delta(e_1, e_2),$$

where the minimum is taken over all surjections η from the larger of the sets S_1 and S_2 to the smaller.

Fair surjection distance (d_{fs}) In order to overcome certain unsatisfactory behavior of the surjection measure d_s, Oddie introduced a variant of the surjection measure, in which admissible surjections must be fair. A surjection η between S_1 to S_2 is *fair*, if η maps the elements of S_1 as evenly as possible onto the elements of S_2, i.e., $| \, |\eta^{-1}(x)| - |\eta^{-1}(y)| \, | \leq 1$ for all $x, y \in S_2$, where $\eta^{-1}(z) = \{w : \eta(w) = z\}$. Thus,

$$d_{fs}(S_1, S_2) = \min_{\eta'} \sum_{(e_1,e_2)\in\eta'} \Delta(e_1, e_2),$$

where the minimum is taken over all surjections.[2]

Link distance (d_l) We propose an alternative distance measure between two sets. Given two sets $S_1, S_2 \subseteq B$, a *linking* between S_1 and S_2 is a relation $R \subseteq S_1 \times S_2$ satisfying

(1) for all $e_1 \in S_1$ there exists $e_2 \in S_2$ such that $(e_1, e_2) \in R$, and
(2) for all $e_2 \in S_2$ there exists $e_1 \in S_1$ such that $(e_1, e_2) \in R$.

The *minimum link distance* d_l between subsets S_1 and S_2 is defined by

$$d_l(S_1, S_2) = \min_{R} \sum_{(e_1,e_2)\in R} \Delta(e_1, e_2),$$

where the minimum is taken over all relations R such that R is a linking between S_1 and S_2.

[2] Oddie [14] actually considered only the version normalized by the size of the larger set.

For convenience, we adopt the following convention. For any binary relation $R \subseteq B \times B$, the *cost of* R, $c(R)$, is $c(R) = \sum_{(e_1,e_2)\in R} \Delta(e_1, e_2)$. Notice that the cost of a surjection, a fair surjection, and a linking between sets S_1 and S_2 is implicit by this definition.

Example 2.1 Consider the sets $S_1 = \{a, b, c, e\}$ and $S_2 = \{c, d\}$ in Fig. 2.

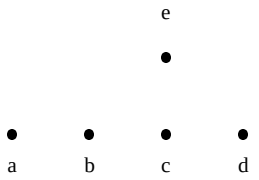


Fig. 2. Examples of the distance measures; the distance from a to b is the unit distance

The distance functions d_{md}, d_s, d_{fs} and d_l evaluate as follows.

$$
\begin{aligned}
d_{md}(S_1, S_2) &= ((0 + 1) + (0 + 1 + 1 + 2))/2 = 5/2.\\
d_s(S_1, S_2) &= 5; \quad \text{the surjection } \eta = \{ (a, c), (b, c), (e, c), (c, d) \}\\
&\qquad \text{is optimal and } c(\eta) = 2 + 1 + 1 + 1 = 5.\\
d_{fs}(S_1, S_2) &= 6; \quad \text{the fair surjection } \eta' = \{ (a, c), (b, d), (e, c), (c, d) \}\\
&\qquad \text{is optimal and } c(\eta') = 2 + 2 + 1 + 1 = 6.\\
d_l(S_1, S_2) &= 5; \quad \text{the linking } L = \{(a, c), (b, c), (e, c), (c, d)\}\\
&\qquad \text{is optimal and } c(L) = 1 + 1 + 1 + 2 = 5. \quad \square
\end{aligned}
$$

The following proposition is easily derived from the definitions.

Proposition 2.1 *Assume that Δ is a distance function on B. Then d_{md}, d_s, d_{fs}, and d_l are distance functions on $\mathscr{P}_\emptyset(B)$.* $\square$

For a discussion of the relationships and properties of d_{md}, d_s, and d_{fs}, the reader is referred to [13]. As mentioned above, these functions have been considered for measuring the distance between logical theories. However, they have also applications in other contexts.

The minimum distance measure is appealing in many cases. For example, suppose the distance between two countries with respect to travelling should be determined. For this purpose, the (possibly normalized) result of the d_{md} function, evaluated on sets S_1, S_2 of representative cities of the countries seems well apt; it amounts to the least cost of getting from a city in one country to a closest city in the other country, and takes account of each possible starting point. The intercity distance Δ may be measured in different ways, e.g. in terms of the geographical distance, or by the cheapest ticket (in this case, Δ may not be a metric). An example for an application of the surjection distance measures in the plane is given below.

We argue that the minimum link measure is more appealing than the other measures in some scenarios. A situation in which d_l is intuitively more appealing than d_s

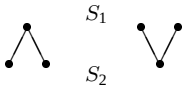
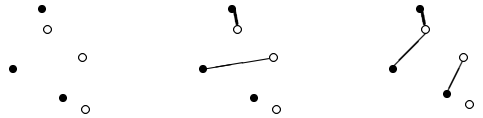


Fig. 3. Situation favoring the minimum link measure d_l over d_s and d_{fs}



(a) (b) (c)

Fig. 4a–c. Situation favoring the minimum link measure d_l over d_{dmd}

or d_{fs} is sketched in Fig. 3, where B is the plane and Δ is the euclidean distance between points.[3] (Notice that d_s and d_{fs} take the same value as each surjection between S_1 and S_2 is trivially fair.)

Each pair of the minimum linking between S_1 and S_2 is represented by a line between dots. The small value of d_l seems to represent the intuitive distance between the point sets better than the larger value of d_s (resp. d_{fs}). Note that the value of d_s (resp. d_{fs}) increases if the distance between the left and the right group of three vertices increases, while the value of d_l remains the same; this behavior seems to be more intuitive.

Comparing d_l to d_{md}, we find that the latter acts in a sense *locally* (or *pointwise*), since the distance between sets S_1 and S_2 is basically the sum of the distances of each point in the one set to the other set. On the other hand, d_l acts *globally* (or *setwise*), since it aims at minimizing the total distance value rather than the distance value of each point. In particular, it is possible that an optimal linking L between S_1 and S_2 contains a pair (e_1, e_2) such that $\Delta(e_1, e_2) > \Delta_m(e_1, S_2)$ and $\Delta(e_1, e_2) > \Delta_m(e_2, S_1)$ (see Fig. 4b). In some situations, setwise distance is more appropriate than pointwise.

For example, assume that point sets are the sites of organizations, and that an organization is interested in a working exchange program with another organization such that mutual connections between sites are established, involving all sites of both organizations. For example, organizations might be associations of universities from the same country. Here, the d_l measure is more appropriate for measuring the geographical distance between the organizations than the d_{md} measure. An example is given in Fig. 4. For the situation in (a), d_l and d_{md} amount to the connections between sites as shown in (b) and (c), respectively. The solution in (b) can be considered preferable; it seems to correspond more closely with the intuition.

The global operation mode of d_l makes it also appealing for measuring the distance between logical theories, based on a revision argument. Suppose that as in Sect. 1, the points of a metric space are interpretations of a logical language $\mathscr{L}$, and that theories are identified with the sets of their models (i.e., sets of points).[4] The distance between two theories S_1 and S_2 can be seen as the "cost" for a believer to change

[3] Similar situations can be found for the integral plane and the Manhattan metric.

[4] By the assumptions from above, only finitely many interpretations may be in the metric space. For example, if $\mathscr{L}$ is a first-order language over a finite vocabulary, all interpretations of $\mathscr{L}$ in a fixed finite domain might be included. In particular, a propositional $\mathscr{L}$ over a finite set of atoms has only finitely many interpretations. Such languages are relevant in the area of data and knowledge representation.

his current view, given by S_1, to S_2. For this purpose, each model in S_1 must be revised to some model in S_2, and all models of S_2 must be covered; it is reasonable to allow that the same model from S_1 is multiply revised to different models in S_2 (think of a model in S_1 which is very close to several models in S_2; see also Fig. 3), and that different models in S_1 can be revised to the same model in S_2. The cost of a particular revision process is the sum of the distances between each model of S_1 and its revisions. Notice that by inverting each revision step, S_2 can be revised to S_1 and thus the cost of revision is symmetric. The least cost for a revision process between S_1 and S_2 is thus a suggestive value for the distance between S_1 and S_2. However, this is precisely the value computed by d_l. Notice that d_{md} is less appealing for this application; due to pointwise minimization, this function reflects the cost of revising one set to a close part of the other set rather than to the whole set.

3 Link graph

We associate with each linking L between sets of points S and S' a graph $G(L)$, which we call the link graph of L. The link graph is useful for studying structural properties of linkings.

Assume that $S = \{s_1, \ldots, s_n\}$ and $S' = \{s'_1, \ldots, s'_m\}$ (note that $S_1 \cap S_2 \neq \emptyset$ is possible). Let $V_1 = \{v_1^1, \ldots, v_n^1\}$ and $V_2 = \{v_1^2, \ldots, v_m^2\}$ be disjoint sets of vertices. For every linking L between S and S', $G(L)$ is the undirected bipartite graph $(V_1 \cup V_2, E)$, where $E = \{\{v_i^1, v_j^2\} : (s_i, s'_j) \in L\}$.

We note some useful structural properties of the link graph.

Proposition 3.1 *For each surjection $\eta : S \to S'$, the graph $G(\eta)$ is the disjoint union of $|S'|$ lines and stars, whose centers are all from V_2.*

Proof. Each set $\{v_j^2\} \cup \{v_i^1 : s_i \in \eta^{-1}(s'_j)\}$, where $1 \leq j \leq m$, induces a line or star in $G(\eta)$. $G(\eta)$ is clearly the disjoint union of all these stars and lines. □

Proposition 3.2 *Let L be an optimal linking between S and S'. Then $G(L)$ is the disjoint union of stars and lines.*

Proof. If not, an edge can be removed from $G(L)$ so that the resulting graph is $G(L')$ for a linking $L' \subseteq L$ between S and S', which contradicts optimality of L. □

Thus, for every optimal linking L between S and S', we can imagine $G(L)$ as a forest of stars and lines. By taking the metric Δ into account, the following properties of the stars in this forest can be derived.

For a disjoint union $G(L)$ of a forest of stars and lines, denote by $cst(L)$ the set of all centers of the stars of $G(L)$. For example, in Fig. 5 we have $cst(L) = \{v_1^1, v_3^1, v_7^2\}$.

It can be easily seen that for an optimal linking L, each center x of a star in $G(L)$ is connected only to vertices y such that x corresponds to a closest point of S (resp. S') for y.

Proposition 3.3 *Let L be an optimal linking between S and S' and let $\{v_i^1, v_j^2\}$ be an edge of $G(L)$. If $v_i^1 \in cst(L)$, then $\Delta(s_i, s'_j) = \Delta_m(s'_j, S)$, and if $v_j^2 \in cst(L)$, then $\Delta(s_i, s'_j) = \Delta_m(s_i, S')$.*

An analogous proposition holds on optimal surjections.

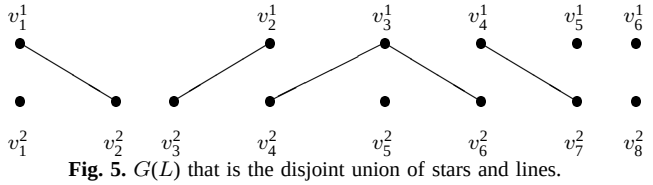


Fig. 5. $G(L)$ that is the disjoint union of stars and lines.

Proposition 3.4 *If $\eta : S \to S'$ is an optimal surjection between S and S', then for each i and j, $1 \leq i \leq n$, $1 \leq j \leq m$, such that $\{v_i^1, v_j^2\}$ is an edge of $G(\eta)$ but not a line, we have $\Delta(s_i, s'_j) = \Delta_m(s_i, S')$.*

4 Computing the distance measures

In this section, we investigate computing the distance functions from the previous section. While it is clear that under suitable assumptions computing d_{md} is simple and efficient, this is not obvious for d_s, d_{fs}, and d_l. It turns out that all these distance functions are computable in polynomial time, however.

4.1 Assumptions, graph matchings, and network flows

We need some assumptions for distance computation first. The input for computing $d(S, S')$ consists of the sets $S = \{s_1, \ldots, s_n\} \subseteq B$ and $S = \{s'_1, \ldots, s'_m\} \subseteq B$, such that $n \geq m \geq 1$. Furthermore, we assume that the distance function Δ on B takes rational values and can be computed in polynomial time.[5]

Our algorithms make use of graph matching and network flow techniques. A *matching* of an undirected weighted graph $G = (V, E, w)$ is a set of edges $M \subseteq E$ such that for each pair of distinct $e, e' \in M$, it holds that $e \cap e' = \emptyset$; the matching M is called *perfect* iff $\bigcup M = V$. The *weight* $w(M)$ of M is $w(M) = \sum_{e \in M} w(e)$. It is well-known that a perfect matching of minimum weight (hence, by our assumptions, also its weight) in G can computed in polynomial time (cf. [10]). Specialized algorithms have been developed for bipartite graphs and for other graph classes.

Proposition 4.1 (cf. [10, p.93,Theorem 14]) *Let $G = (V_1 \cup V_2, E, w)$ be a bipartite weighted graph with nonnegative weights from the reals. Then, a perfect matching M in G of minimal weight (if it exists) can be computed in time*

$$O\left(\frac{|V_1| \cdot |E| \cdot \log |V_1|}{\max(1, \log(|E|/|V_1|))}\right). \qquad \square$$

A *network* $N = (V, E, cap, c)$ is a directed weighted graph (V, E, c) where each edge $e \in E$ has assigned a capacity $cap(e) \in \Re^+$; the second weight function c assigns a cost $c(e)$ to each edge e. Let s and t be specified vertices from V. An s-t-flow (or simply flow, if s and t are understood) on N is a function $f : E \cup \{(x, y) : (y, x) \in E\} \to \Re^+$ such that

[5] This assumption is stricter than necessary, but avoids problems arising if the values of Δ are possibly difficult to compare to a number or among each other (cf. [7]).

(i) for each $(x, y) \in E$, $0 \leq f(x, y) \leq cap(x, y)$ and $f(y, x) = -f(x, y)$, and
(ii) for each $x \neq s, t$ we have $\sum_{(x,y)\in E} f(x, y) = \sum_{(z,x)\in E} f(z, x)$.

The first condition states that the flow on an edge is legal and that on a "loop" $(x, y), (y, x)$, the flow must be zero. The second condition states that what flows in, flows out of every node distinct from s and t. The value $|f|$ of flow f is defined as $|f| = \sum_{(s,x)\in E} f(s, x)$, and the cost $c(f)$ of f is defined as $c(f) = \sum_{e\in E} w(e)f(e)$.

A maximum flow is any flow f such that $|f|$ is a maximum over all flow values, and a minimum cost flow of value v is any flow f such that $|f| = v$ and $c(f)$ is minimum over the costs of all flows of value v. The *integrality theorem for minimum cost flows* states the following.

Theorem 4.2 *(*[24, p. 593]*) If all capacities in a network $N = (V, E, cap, c)$ are integers, then there exists a maximum flow which is integral and has minimum cost (over all maximum flows).*

4.2 Computing the surjection distance d_s

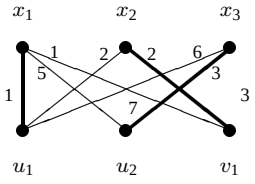
First we consider computing d_s for $S = \{s_1, \ldots, s_n\}$ and $S' = \{s'_1, \ldots, s'_m\}$. We show that computing $d_s(S, S')$ can be reduced to computing the cost of a minimum weight perfect matching in a graph in polynomial time.

Construct from S and S' an complete bipartite weighted graph $G = (X \cup Y, E, w)$ as follows. Let $k = n - m$. Let $X = \{x_1, \ldots, x_n\}$ and $Y = U \cup V$, where $U = \{u_1, \ldots, u_m\}$ and $V = \{v_1, \ldots, v_k\}$. The weight function w is defined as follows.

$$w(e) = \begin{cases} \Delta(s_i, s'_j) & \text{for } e = \{x_i, u_j\}, \quad 1 \leq i \leq n, 1 \leq j \leq m; \\ \Delta_m(s_i, S') & \text{for } e = \{x_i, v_j\}, \quad 1 \leq i \leq n, 1 \leq j \leq k. \end{cases}$$

The link graph $G(\eta)$ of an optimal surjection is a collection of stars and lines. The auxiliary vertices v_i allow us to obtain from $G(\eta)$ a graph with a perfect matching: from each star of $G(\eta)$, all vertices except one are linked to an auxiliary node.

Figure 6 shows an example of the construction, where $S = \{s_1, s_2, s_3\}$, $S' = \{s'_1, s'_2\}$, and the distance function Δ is assumed to yield the following values: $\Delta(s_1, s'_1) = 1$, $\Delta(s_2, s'_1) = 2$, $\Delta(s_3, s'_1) = 6$, $\Delta(s_1, s'_2) = 5$, $\Delta(s_2, s'_2) = 7$ and $\Delta(s_3, s'_2) = 3$. Notice that for a minimum weight perfect matching M of G, we have $w(M) = 6$.



$\Delta(s, s')$	s_1	s_2	s_3
s'_1	1	2	6
s'_2	5	7	3

Fig. 6. Graph G for $S = \{s_1, s_2, s_3\}$ and $S' = \{s'_1, s'_2\}$. The bold edges constitute a minimum weight perfect matching of G

Lemma 4.3 *Let M be an arbitrary minimum weight perfect matching of G and let η be an optimal surjection between S and S'. Then $w(M) = c(\eta)$.*

Proof. We show first that $w(M) \leq c(\eta)$. From an optimal surjection η we get a perfect matching M' of G such that $w(M') \leq c(\eta)$ as follows. We take all edges $\{x_{f(j)}, u_j\}$ such that $f(j)$ is the least index of any s_h in $\eta^{-1}(s'_j)$, and all edges $\{x_{g_i}, v_i\}$ for an arbitrary enumeration $x_{g_1}, x_{g_2}, \ldots, x_{g_k}$ of the remaining x vertices. M' consists of the following edges:

(*i*) $\{x_{f(j)}, u_j\}$ for every j with $1 \leq j \leq m$, where $f(j) = \min\{i : s_i \in \eta^{-1}(s'_j)\}$;
(*ii*) $\{x_{g_i}, v_i\}$ for every i with $1 \leq i \leq k$, where $\{x_{g_1}, \ldots, x_{g_k}\} = X - \{x_{f(1)}, \ldots, x_{f(m)}\}$.

It is not hard to see that M' is indeed a perfect matching in G. Since $w(x_i, v_l) \leq w(x_i, u_j)$, for all i, j, l with $1 \leq i \leq n$, $1 \leq j \leq m$, and $1 \leq l \leq k$, we have that $w(M') \leq c(\eta)$; hence, it follows that $w(M) \leq c(\eta)$.

On the other hand, $w(M) \geq c(\eta)$ holds. Indeed, from a matching M we can define a surjection $\eta' : S \to S'$ as follows. Let for all i with $1 \leq i \leq n$ be

$$\eta'(s_i) = \begin{cases} s'_j, & \text{if } \{x_i, u_j\} \in M; \\ \mu(s_i, S'), & \text{if } \{x_i, v_l\} \in M \text{ for some } l. \end{cases}$$

It is easy to see that η' is well-defined and indeed a surjection, and that $c(\eta') = w(M)$. Hence, it follows that $w(M) \geq c(\eta)$; the result follows. □

Theorem 4.4 *The distance function d_s is computable in polynomial time.*

Proof. Consider sets S and S'. The graph G for S and S' is clearly constructible in polynomial time. Since a minimum weight perfect matching M in G can be constructed in polynomial time, by our assumptions $w(M)$ can be computed in polynomial time. Hence the result follows. □

Remark: Since the graph G is bipartite, a minimum weight perfect matching in G can be computed in time $O(n^3)$ (cf. Proposition 4.1; $|V_1| = n, |E| = n^2$).

4.3 Computing the fair surjection distance d_{fs}

Next we show that d_{fs} can be efficiently computed by solving a network flow problem.

Given sets $S = \{s_1, \ldots, s_n\}$ and $S' = \{s'_1, \ldots, s'_m\}$ with $1 \leq m \leq n$, we construct a network $N = (V, E, cap, c)$ as follows. Let $V = X \cup Y \cup \{s, t\}$, where $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$ are disjoint. The set of edges E consists of four groups of edges e with capacity $cap(e)$ and cost $c(e)$ as follows. Let $c_0 = \lceil n/m \rceil$.

(*i*) (s, x_i), for every i with $1 \leq i \leq n$, for which $cap(s, x_i) = 1$ and $c(s, x_i) = 0$,
(*ii*) (x_i, y_j), for every i and j with $1 \leq i \leq n$, $1 \leq j \leq m$, for which $cap(x_i, y_j) = 1$ and $c(x_i, y_j) = \Delta(s_i, s'_j)$,
(*iii*) (y_i, t), for every i with $1 \leq i \leq m$, for which $cap(y_i, t) = c_0$ and $c(y_i, t) = 0$, and
(*iv*) (s, y_i), for every i with $1 \leq i \leq m$, for which $cap(s, y_i) = 1$ and $c(s, y_i) = \Omega$,

where Ω is an arbitrary real number greater than $\sum_{i,j} \Delta(x_i, y_j)$. Figure 7 shows an example of the construction, where S, S', and Δ are as above.

The intuition behind N is as follows. Mapping S to S' corresponds to forwarding from every vertex x_i one unit of flow coming from the source s to one of the vertices y_j, from which it is transported to the sink t. The capacity of the edge from y_j to t

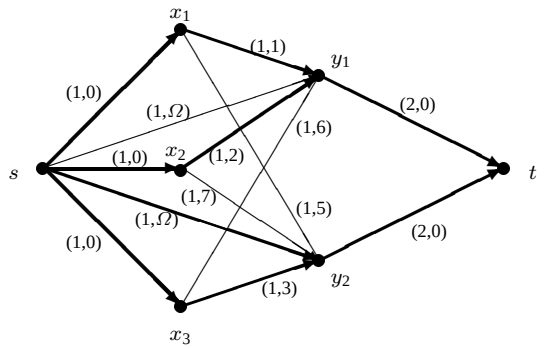


Fig. 7. Network N for $S = \{s_1, s_2, s_3\}$, $S' = \{s'_1, s'_2\}$. The flow f assigning $cap(e)$ to the bold edges and 0 to all others is a minimum cost maximum flow

is restricted so that no more units can be transported than elements from S can be mapped to s'_j by a fair surjection. The edges from s to the y_j vertices assure that at least as many units reach y_j from the x_i vertices as elements from S must at least be mapped to y_j in any fair surjection. Each such edge can transport one unit of flow, but only at the extremely high cost Ω. Thus, as few units as necessary will be transported via such edges in a maximum flow.

It is straightforward to check that for any S and S', a maximum s-t flow in N has value $c_0 m = n + k$, where

$$k = \begin{cases} 0, & \text{if } n = i \cdot m, \text{ for some integer } i \geq 0 \\ m - (n \bmod m), & \text{otherwise.} \end{cases}$$

In a maximum flow of minimum cost exactly k units are transported from s directly to the y_j vertices and all other units via the x_i vertices, such that the flow corresponds to an optimal fair surjection between S and S'. Since all capacities are integers, the integrality theorem for minimum cost flows asserts that an integral maximum s-t flow f of minimum cost on N exists. Clearly, each $e \in E$ with $cap(e) = 1$ has in f value 0 or 1, i.e., $cap(e)$. It can be easily seen that in f every other edge e also has value 0 or $cap(e)$.

For an example, consider Fig. 7. There, $k = 1$. The cost of the flow f is $c(f) = 6+1\cdot\Omega$. On the other hand, $\eta' = \{ (s_1, s'_1), (s_2, s'_1), (s_3, s'_2) \}$ is an optimal fair surjection between S and S' with $c(\eta) = 6$.

The next lemma states that the construction works correctly.

Lemma 4.5 *Let f be a maximum s-t flow of minimum cost on N and let η be an optimal fair surjection between S and S'. Then $c(f) = c(\eta) + k\Omega$.*

Proof. First observe that a surjection η between S and S' is fair if and only if $\lfloor n/m \rfloor \leq |\eta^{-1}(s'_j)| \leq \lceil n/m \rceil$ for every j, $1 \leq j \leq m$.

We show that $c(f) \geq c(\eta) + k\Omega$. By the integrality theorem for minimum cost flows, we can assume that f is integral. As easily seen, f assigns 1 to exactly k edges from $(s, y_1), \ldots, (s, y_m)$, to all edges (s, x_i), where $1 \leq i \leq n$, and to exactly one of the edges $(x_i, y_1), \ldots, (x_i, y_m)$ for every i with $1 \leq i \leq m$; let $(x_i, y_{g(i)})$ be this edge.

Define the mapping $\zeta : S \to S'$ by $\zeta(s_i) = s'_{g(i)}$, for every i with $1 \leq i \leq n$. We show that ζ is a fair surjection between S and S'. Assume this does not hold. Then, there must exist a j with $1 \leq j \leq m$, such that f assigns 1 to fewer than $c_0 - 1$ edges from $(x_1, y_j), \ldots, (x_n, y_j)$ and 0 to the others. The flow value $|f| = c_0 m$ implies that c_0 units reach y_j, i.e. $\sum_{(x,y_j)\in E} f(x, y_j) = c_0$. Note that y_j is reachable by an edge only from $x_1, \ldots, x_n$ and s. Since each of these edges has capacity 1, it follows that no more than $c_0 - 1$ units reach y_j, which is a contradiction. Thus ζ is a fair surjection. It is easy to check that $c(f) = c(\zeta) + k\Omega$.

Since η is an optimal fair surjection, we thus have that $c(f) \geq c(\eta) + k\Omega$.

Now we show that $c(f) \leq c(\eta) + k\Omega$. We construct from a fair surjection η a flow f_η on network N as follows.

$$f_\eta(e) = \begin{cases} c_0, & \text{if } e = (y_j, t) \text{ where } 1 \leq j \leq m; \\ 1, & \text{if } e = (s, x_i) \text{ where } 1 \leq i \leq n, \\ & \quad \text{or } e = (s, y_j) \text{ where } 1 \leq j \leq m \text{ is such that } |\eta^{-1}(s'_j)| < c_0, \\ & \quad \text{or } e = (x_i, y_{\eta(s_i)}) \text{ where } 1 \leq i \leq n; \\ 0, & \text{otherwise.} \end{cases}$$

Clearly, f_η is an s-t flow of value $c_0 m$ and hence maximum. As readily checked, $c(f_\eta) = c(\eta) + k\Omega$. Thus we get that $c(f) \leq c(\eta) + k\Omega$. The result follows. □

Theorem 4.6 *The function d_{fs} is computable in polynomial time.*

Proof. Consider sets S and S'. The network N can be easily constructed from S and S' in polynomial time. Since all capacities are integer, a minimum integral s-t flow of value $v = c_0 m$ on N is computable in time

$$O(ve_N(\log n_N)/\max\{\log(e_N/n_N), 1\}), \quad n_N = |V|, e_N = |E|$$

[10, p. 92, Theorem 13]. The result follows. □

4.4 Computing the link distance d_l

It remains to show that the link distance function d_l is computable in polynomial time. As in the case of d_s, we reduce this problem to a perfect matching problem.

Let again $S = \{s_1, \ldots, s_n\}$ and $S' = \{s'_1, \ldots, s'_m\}$. We define a complete bipartite weighted graph $G = (A \cup B, E, w)$ as follows: $A = \{a_1, \ldots, a_n\}$, $B = \{b_1, \ldots, b_m\}$, and for each edge $e = \{a_i, b_j\}$ of G, $w(e) = \Delta(s_i, s'_j)$. Intuitively, G is a graph in which each $s_i \in S$ (resp. $s'_j \in S'$) is connected to every s'_j (resp. s_i) at the cost of the distance $\Delta(s_i, s'_j)$. Let G' be a zero-weight copy of G, i.e., $G' = (A' \cup B', E', w')$ is a complete bipartite weighted graph where $A' = \{a'_1, \ldots, a'_n\}$, $B' = \{b'_1, \ldots, b'_m\}$, $(A \cup B) \cap (A' \cup B') = \emptyset$, and $w'(e) = 0$ for each edge e. From G and G' we define the graph G'' by taking the union of G and G' and connecting every a_i and b_j to a'_i and b'_j, respectively, by weight equal to its nearest neighbor in G. That is,

$$G'' = (V'', E'', w''),$$

where

$$
\begin{aligned}
V'' &= A \cup A' \cup B \cup B', \\
E'' &= E \cup E' \cup \{\{a_i, a_i'\}, \{b_j, b_j'\} : a_i \in A, b_j \in B\}, \\
w''(e) &= \begin{cases} w(e), & \text{for } e \in E; \\ w'(e), & \text{for } e \in E'; \\ \Delta_m(s_i, S'), & \text{for } e = \{a_i, a_i'\}, \quad 1 \le i \le n; \\ \Delta_m(s_i', S), & \text{for } e = \{b_j, b_j'\}, \quad 1 \le j \le m. \end{cases}
\end{aligned}
$$

Figure 8 shows an example of the construction; S, S', and Δ are as above. The set of

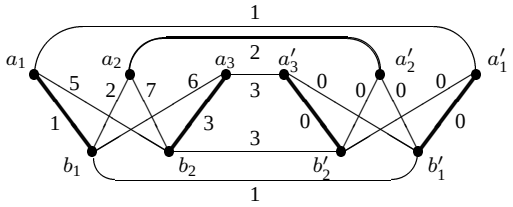


Fig. 8. The graph G'' for $S = \{s_1, s_2, s_3\}$, $S' = \{s_1', s_2'\}$. The bold edges constitute a minimum weight perfect matching

edges outlined in bold face constitute a perfect matching M of minimum weight; M has value $w(M) = 6$. Notice that the link measure between S and S' amounts to the same value: the linking $L = \{(s_1, s_1'), (s_2, s_1'), (s_3, s_2')\}$ is an optimal linking between S and S' with cost $c(L) = 6$.

Lemma 4.7 *Let L be an optimal linking between S and S', and let M be a minimum weight perfect matching in G''. Then $c(L) = w(M)$.*

Proof. We first show that $w(M) \le c(L)$. We get from L a perfect matching M' in G'' whose weight is equal to the cost of L as follows. Let the centers of the stars of $G(L)$ be $cst(L) = \{y_1, \ldots, y_k\}$, and let $x_1, \ldots, x_k$ be arbitrary vertices from $G(L)$ such that x_i is connected by an edge to y_i, for all $i = 1, \ldots, k$. Then, define that M' consists of the following edges:

(i) $\{a_i, b_j\}$ and $\{a_i', b_j'\}$, for all edges $\{v_i^1, v_j^2\}$ of $G(L)$ such that $v_i^1, v_j^2 \notin \{x_1, \ldots, x_k\}$;
(ii) $\{a_i, a_i'\}$, for every $i = 1, \ldots, n$ such that $v_i^1 \in \{x_1, \ldots, x_k\}$;
(iii) $\{b_j, b_j'\}$, for every $j = 1, \ldots, m$ such that $v_j^2 \in \{x_1, \ldots, x_k\}$.

By Proposition 3.2, $G(L)$ is the disjoint union of stars and lines. Hence, it is not hard to see that M' is a perfect matching in G''. Moreover, by the construction of G'' and the definition of M', it follows immediately from Proposition 3.3 that $w(M') = c(L)$. It follows that $w(M) \le c(L)$.

Now we show that also $w(M) \ge c(L)$. From M we can get a linking L' between S and S' whose cost is equal to $w(M)$ as follows. Define that L' contains all pairs (v_i^1, v_j^2), for each $i = 1, \ldots, n$ and $j = 1 \ldots, m$, such that one of the following conditions is satisfied:

(i) $\{a_i, b_j\} \in M$;
(ii) $\{a_i, a_i'\} \in M$ and $s_j' = \mu(s_i, S')$ (s_j' from S' is closest to s_i);
(iii) $\{b_j, b_j'\} \in M$ and $s_i = \mu(s_j', S)$.

Since M is a perfect matching in G'', it is easy to see that L' is indeed a linking between S and S', and that $c(L') = w(M)$. Hence, it follows that $c(L) \leq w(M)$. The result follows. □

We remark that a modified version of the graph G'' provides another possibility for computing the surjection measure. In fact, if all edges $\{b_j, b'_j\}$ are removed from G'', then the perfect matchings in the resulting graph G''' correspond to surjections between S and S', and the cost of a minimum weight perfect matching in G''' equals the cost of an optimal surjection between S and S'. However, G''' is more complex than the graph G in Sect. 4.2, on which a perfect matching algorithm is expected to perform better in general.

Theorem 4.8 *The distance function d_l is computable in polynomial time.*

Proof. Consider sets S and S'. The graph G'' for S and S'' can be constructed in polynomial time. A minimum cost perfect matching in G'' can be computed in polynomial time, cf. [10]; hence the result follows. □

The graph G'' is bipartite by the partition of the vertex sets constituted by $A \cup B'$ and $A' \cup B$. Therefore, a minimum cost perfect matching in G'' can be computed in time

$$O\left(n \cdot m \cdot (n+m) \cdot \log(n+m) \cdot (1/\max(1, \log \frac{2n \cdot m + n + m}{n+m}))\right)$$

(cf. Proposition 4.1; $|V_1| = n + m, |E| = 2nm + n + m$).

Let us remark at the end of this section that in case Δ is a metric on B, the efficiency of the proposed algorithms might be improved by using metric matching techniques (cf. [23]). The network flow and matching problems constructed in this section involve additional vertices to which the underlying metric space B does not extend; thus, the constructions would have to be suitably adapted. Notice that the main goal of this section is showing that certain distance functions are computable in polynomial time and not intractable. We confined ourselves for this purpose to using standard methods.

5 From distance functions to metrics

In certain contexts, it is natural or desired that a measure of distance between point sets is a metric, i.e., it satisfies the postulates of a distance function and in addition the triangle inequality.

For example, suppose that the amount of work that has to be done to change a set into another should be reflected in the interdistance value attached to these point sets; for a concrete instance, assume that the points are strings. In this case, the triangle inequality is a reasonable postulate for a suitable distance measure, since changing a set S_1 into a set S_2 is not more expensive than the sum of changing S_1 to S_3 and changing S_3 into S_2.

Each of the distance functions between two sets defined in the previous section, d_{md}, d_s, d_{fs}, and d_l, fails to satisfy the triangle inequality, as shown by the following example.

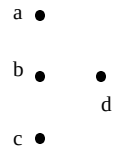


Fig. 9. Violations of the triangle inequality ($S_1 = \{a, c\}$, $S_2 = \{d\}$, $S_3 = \{b\}$)

Example 5.1 Consider $S_1 = \{a, c\}$, $S_2 = \{d\}$, and $S_3 = \{b\}$ in Fig. 9. Then $d_{md}(S_1, S_2) = 6/2 = 3$, $d_{md}(S_1, S_3) = 3/2$ and $d_{md}(S_3, S_2) = 2/2 = 1$; hence, $d_{md}(S_1, S_3) + d_{md}(S_3, S_2) = 5/2 < d_{md}(S_1, S_2)$, i.e., the triangle inequality fails. The same applies to the other distance functions d_α, for every $\alpha \in \{s, fs, l\}$, for which we have $d_\alpha(S_1, S_2) = 4$, $d_\alpha(S_1, S_3) = 2$, and $d_\alpha(S_3, S_2) = 1$.[6] □

In this section we describe a simple construction that produces from a distance function a metric which has certain appealing properties. More precisely, given an arbitrary distance function $d : \mathscr{P}_\emptyset(B)^2 \to \Re^+$, where B is finite, the construction produces a metric d^ω on $\mathscr{P}_\emptyset(B)$. For the distance functions d_{md}, d_s, d_{fs}, and d_l the respective metrics coincide on singletons with the metric Δ on B.

Call any finite sequence $P = (P_1, P_2, \ldots, P_m)$, where $m \geq 2$ and $P_i \subseteq B$, for all i with $1 \leq i \leq m$, a *path between P_1 and P_m*; the *length* of the path is m. We use $P(S, S')$ to denote the set of all paths between S and S'. The concatenation of paths $P = (P_1, \ldots, P_m)$ and $P' = (P_m, \ldots, P_n)$ is the path $PP' = (P_1, \ldots, P_n)$. The *cost* $c_d(P)$ of P under a distance function d is defined by $c_d(P) = \sum_{i=1}^{m-1} d(P_i, P_{i+1})$. A path $P \in P(S, S')$ is called *optimal under d* (or *d-optimal*) iff $c_d(P) = \min\{c_d(P') : P' \in P(S, S')\}$.

Define the function $d^\omega : \mathscr{P}_\emptyset(B)^2 \to \Re^+$ by

$$d^\omega(S, S') = \min\{c_d(P) : P \in P(S, S')\}.$$

We observe that d^ω has the following appealing properties. Let the functions $f : \mathscr{P}_\emptyset(B) \times \mathscr{P}_\emptyset(B) \to \Re^+$ ($\Re^+$ are the nonnegative reals) be partially ordered by $f \leq g$ iff $f(S_1, S_2) \leq g(S_1, S_2)$ for all S_1, S_2.

Theorem 5.1 *Let d be a distance function on $\mathscr{P}_\emptyset(B)$, where B is finite. Then,*

(a) d^ω defines a metric on $\mathscr{P}_\emptyset(B)$.
(b) d^ω is the unique maximum of the distance functions $f : \mathscr{P}_\emptyset(B)^2 \to \Re^+$ which satisfy

$$f(S, S') = \min(d(S, S'), \min_{S''}(f(S, S'') + f(S'', S'))) \tag{1}$$

(c) d^ω is the unique maximum metric function $f : \mathscr{P}_\emptyset(B)^2 \to \Re^+$ with $f \leq d$.

Proof. It is easy to see that d^ω is symmetric and that $d^\omega(S, S') = 0$ iff $S = S'$, hence d^ω is a distance function. Moreover, from the definition of d^ω it follows that the triangle inequality holds; hence (a) follows.

It is easy to see that d^ω satisfies Equation 1. We show that $f(S, S') \leq d^\omega(S, S')$ by induction on the length of the shortest d-optimal path $(P_1, \ldots, P_m) \in P(S, S')$.

[6] More complex examples show the failure of the triangle inequality for the versions of the measures from normalization by the size of the larger set.

If $m = 2$, then $d^\omega(S, S') = d(S, S')$ and $f(S, S') \leq d^\omega(S, S')$ holds. If $m > 2$, then $d^\omega(S, S') = d(S, P_2) + d^\omega(P_2, S')$, hence by the induction hypothesis and Equation 1, $d^\omega(S, S') \geq f(S, P_2) + f(P_2, S') \geq f(S, S')$, and the statement holds. Consequently, $f \leq d^\omega$ and d^ω is a unique maximum of the distance functions satisfying (1); thus (b) holds.

It remains to consider (c). Assume f is a metric such that $f \leq d$ and that $d^\omega(S, S') < f(S, S')$ for some S and S'. Now $d^\omega(S, S') = \sum_{i=1}^{m-1} d(P_i, P_{i+1})$ for some d-optimal path $(P_1, \ldots, P_m) \in P(S, S')$. By the triangle inequality $f(S, S') \leq \sum_{i=1}^{m-1} f(P_i, P_{i+1})$ and by assumption $f(P_i, P_{i+1}) \leq d(P_i, P_{i+1})$ for $1 \leq i < m$, thus we get $f(S, S') \leq d^\omega(S, S')$, a contradiction. □

In the light of Theorem 5.1(c), we refer to d^ω as the *metric infimum* of d.

Recall that our motivation was to extend a distance function Δ on B to $\mathscr{S}_\emptyset(B)$ so that $\Delta(\{x\}, \{y\}) = \Delta(x, y)$. Thus, if we want this distance function to be a metric, Δ must be a metric on B. If this is the case, then each of the metric infima of the distance functions d_{md}, d_s, d_{fs}, and d_l in Sect. 2 satisfies the property $d^\omega(\{x\}, \{y\}) = \Delta(x, y)$.

Proposition 5.2 *Let d and d_a be distance functions on $\mathscr{S}_\emptyset(B)$ such that $d_a \leq d$ and for all $x, y \in B$, $d(\{x\}, \{y\}) = \Delta(x, y)$ and $d_a^\omega(\{x\}, \{y\}) = \Delta(x, y)$. Then, $d^\omega(\{x\}, \{y\}) = \Delta(x, y)$ for all $x, y \in B$.*[7]

Proof. From $d_a \leq d$ it follows easily that $d_a^\omega \leq d^\omega$. Since $d_a^\omega(\{x\}, \{y\}) = \Delta(x, y)$, we have $\Delta(x, y) \leq d^\omega(\{x\}, \{y\})$. On the other hand, since $d(\{x\}, \{y\}) = \Delta(x, y)$, it follows (cf. Theorem 5.1(b)) that $d^\omega(\{x\}, \{y\}) \leq \Delta(x, y)$. The result follows. □

Corollary 5.3 *Assume that the distance function Δ satisfies the triangle inequality, i.e., Δ is a metric on B. Then, for each $\alpha \in \{md, s, fs, l\}$, we have $d_\alpha^\omega(\{x\}, \{y\}) = \Delta(x, y)$ for all $x, y \in B$.*

Proof. Choose the Hausdorff distance d_h for d_a (note that $d_h^\omega = d_h$) and d_α for d. □

Let us have a closer look at the properties of different d^ω measures. Interestingly, the metric infima of the distance functions d_s, d_{fs}, d_l collapse to a single function.

Theorem 5.4 $d_s^\omega = d_{fs}^\omega = d_l^\omega$.

Proof. Clearly, $d_l^\omega \leq d_s^\omega \leq d_{fs}^\omega$. (Every surjection between S and S' is a linking between S and S').

Notice that in order to show $d_\alpha^\omega \leq d_\beta^\omega$, it suffices to show that for all $S = \{s_1, \ldots, s_n\}$ and $S' = \{s'_1, \ldots, s'_m\}$ where $1 \leq m \leq n$ and $d_\beta(S, S') = d_\beta^\omega(S, S')$, there exists a path $P = (P_1, \ldots, P_m) \in P(S, S')$ such that $c_{d_\alpha}(P) \leq d_\beta(S, S')$.

The claim $d_s^\omega \leq d_l^\omega$ is shown as follows. Let L be an optimal linking between S and S' such that $c(L) = d_l^\omega(S, S')$. The idea is that a linking can be represented by two surjections. We define from L the surjections $\eta_S : S \to R$ and $\eta_{S'} : S' \to R$ to an intermediate set R. This set contains elements of S and S' corresponding to the centers of stars of $G(L)$ and the endpoints in V_1 of lines of $G(L)$, i.e.,

$$R = \{s_i : v_i^1 \in cst(L) \cap V_1 \text{ or } deg(v_i^1, G(L)) = 1\} \cup \{s'_j : v_j^2 \in cst(L) \cap V_2\}.$$

[7] This proposition can be generalized by replacing '$d(\{x\}, \{y\}) = \Delta(x, y)$ and $d_a^\omega(\{x\}, \{y\}) = \Delta(x, y)$' with the weaker condition '$d(\{x\}, \{y\}) \leq \Delta(x, y)$ and $d_a^\omega(\{x\}, \{y\}) \geq \Delta(x, y)$'.

The surjection η_S will map the elements of S that belong to a line of $G(L)$ or to a S'-star of $G(L)$ to the unique element they are linked by L. Similarly, $\eta_{S'}$ maps members of S' that belong to lines or S-stars of $G(L)$ to the unique element L links them with. Formally, for all i and j with $1 \leq i \leq n$, $1 \leq j \leq m$,

$$\eta_S(s_i) = \begin{cases} s'_k, & \text{if } v_i^1 \text{ is in a star of } G(L) \text{ with center } v_k^2, \\ & \quad \text{or if } \{v_i^1, v_k^2\} \text{ is a line of } G(L); \\ s_i, & \text{otherwise.} \end{cases}$$

$$\eta_{S'}(s'_j) = \begin{cases} s_k, & \text{if } v_j^2 \text{ is in a star of } G(L) \text{ with center } v_k^1, \\ & \quad \text{or if } \{v_k^1, v_j^2\} \text{ is a line of } G(L); \\ s'_j, & \text{otherwise.} \end{cases}$$

Then η_S and $\eta_{S'}$ are surjections which satisfy $c(\eta_S) + c(\eta_{S'}) = c(L)$. Consequently, $c_{d_s}(S, R, S') \leq c(L) = d_l^\omega(S, S')$. It follows that $d_s^\omega \leq d_l^\omega$.

The fact that $d_{fs}^\omega \leq d_s^\omega$ can be shown similarly. The intuition is that a surjection can be decomposed into a sequence of surjections between sets whose cardinality differs only by one. Such surjections will be necessarily fair. Assume that $\eta : S \to S'$ is an optimal fair surjection between S and S' and that $d_s(S, S') = d_s^\omega(S, S')$. We show by induction on the number k of stars and lines of $G(L)$, that there exists a path $P = (P_1, \ldots, P_r) \in P(S, S')$ such that $0 \leq |P_i| - |P_{i+1}| \leq 1$, $1 \leq i < r$, and $c_{d_{fs}}(P) \leq d_s(S, S')$.

If $k = 1$, then $m = 1$, i.e. $G(\eta)$ consists merely of one line or one star. The sequence $Q = (Q_1, \ldots, Q_{n+1})$, where $Q_1 = S$ and $Q_i = Q_{i-1} - \{s_{i-1}\} \cup \{s'_1\}$, for all i with $2 \leq i \leq n+1$, satisfies the properties of P. Indeed, $Q \in P(S, S')$, and clearly $0 \leq |Q_i| - |Q_{i+1}| \leq 1$, for all i, $1 \leq i \leq n$. Define $\eta_i : Q_i \to Q_{i+1}$, $1 \leq i \leq n$, by

$$\eta_i(x) = \begin{cases} s'_1, & \text{if } x = s_i; \\ x, & \text{if } x \in Q_i - \{s_i\}. \end{cases}$$

Each η_i is a fair surjection, and $\sum_{i=1}^n c(\eta_i) = c(\eta)$; hence, $c_{d_{fs}}(Q) \leq d_s(S, S')$. Thus the statement holds for $k = 1$.

Now consider the case $k > 1$. Let $S_1 = S - \eta^{-1}(s'_1)$ and $S'_1 = S' - \{s'_1\}$. We construct a sequence of fair surjections that will ultimately link S and S' by first decomposing an optimal surjection of $\eta^{-1}(S'_1)$ to $\{s'_1\}$ into a sequence of fair surjections (by the induction hypothesis). This sequence is extended to a sequence of fair surjections between S and $S_1 \cup \{s'_1\}$. The induction assumption is again applied, and we get a sequence of fair surjections from $S_1 \cup \{s'_1\}$ to S'. Formally, denote by $\eta[X]$ the restriction of η to X. We note first that

$$d_s(\eta^{-1}(s'_1), \{s'_1\}) = d_s^\omega(\eta^{-1}(s'_1), \{s'_1\}) = c(\eta[\eta^{-1}(s'_1)])$$

and

$$d_s(S_1, S'_1) = d_s^\omega(S_1, S'_1) = c(\eta[S_1])$$

must hold by the assumption $d_s(S, S') = d_s^\omega(S, S')$. Let $P' = (P'_1, \ldots, P'_s) \in P(\eta^{-1}(s'_1), \{s'_1\})$ and $P'' = (P''_1, \ldots, P''_t) \in P(S_1, S'_1)$ of the properties assured by the induction hypothesis on $\eta^{-1}(s'_1), \{s'_1\}$ and S_1, S'_1, respectively. Let

$$\begin{aligned} Q' &= (S_1 \cup P'_1, \ldots, S_1 \cup P'_s), \\ Q'' &= (P''_1 \cup \{s'_1\}, \ldots, P''_t \cup \{s'_1\}). \end{aligned}$$

Note that $S_1 \cup P_1' = S$ and $S_1 \cup P_s' = P_1'' \cup \{s_1'\}$. Hence, $Q = Q'Q''$ is a path in $P(S, S')$ and

$$\begin{aligned} c_{d_{fs}}(Q) &= c_{d_{fs}}(Q') + c_{d_{fs}}(Q'') \\ &\leq d_s(\eta^{-1}(s_1'), \{s_1'\}) + d_s(S_1, S_1') \\ &= c(\eta[\eta^{-1}(s_1')]) + c(\eta[S_1]) = d_s(S, S'). \end{aligned}$$

Clearly Q also satisfies the remaining property of P in the statement for S and S'. Hence, the statement holds on S, S'. It follows that $d_{fs}^{\omega} \leq d_s^{\omega}$.

The proposition follows from $d_l^{\omega} \leq d_s^{\omega} \leq d_{fs}^{\omega}$ and $d_{fs}^{\omega} \leq d_s^{\omega} \leq d_l^{\omega}$. □

One can find instances for which $d_l^{\omega} \neq d_{md}^{\omega}$, however. For an example, consider the sets $S_1 = \{00\}$, $S_2 = \{01, 11\}$, $S_3 = \{11\}$, $S_4 = \{01\}$ in Fig. 10. Here it is assumed that B consists of the four binary strings 00, 01, 10, and 11 whose distance Δ is their Hamming distance.

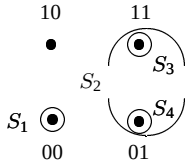


Fig. 10. Example showing that $d_{md}^{\omega} \neq d_l^{\omega}$

Then, $d_l^{\omega}(S_1, S_2) = d_l^{\omega}(S_1, S_3) = 2$. However, for the minimum distance measure we have $d_{md}^{\omega}(S_1, S_3) = 2$, but $d_{md}^{\omega}(S_1, S_2) \leq 3/2$, since one can go from S_2 to S_1 by first going to S_4 and then to S_1; we have $d_{md}(S_2, S_4) = 1/2$. This example shows that the behavior of the d_l^{ω} measure is more natural than that of the d_{md}^{ω} measure (at least in some situations). For this reason, we consider in the next section only the computation of the d_l^{ω} measure.

6 Computing the d_l^{ω} measure

Now let us consider the computational properties of the d_l^{ω} measure. A simple result is that under some general assumptions on the computational properties of Δ and B, the d_l^{ω} measure is computable.

Proposition 6.1 *Assume that Δ is computable and that B is computable, i.e., all points of B can be effectively generated in finite time. Then d_l^{ω} is computable.*

Proof. Compute d_l^{ω} by cycling through all paths of length at most $2^{|B|}$ and take the minimum of their costs. □

Computing d_l^{ω} appears to be more complex than computing d_l. We will show in this section that computing d_l^{ω} is NP-hard for a simple – and quite natural – instance. However, the associated recognition problem is in NP under very general conditions, which entails that computing d_l^{ω} is not much harder than the NP-complete problems.

6.1 Steiner forests and paths between sets

We need additional concepts. Let S be a nonempty subset of B. A weighted tree $T = (V, E, w)$ such that $V \subseteq B$ and w coincides with Δ is a *Steiner tree for* S if $S \subseteq V$; T is called *optimal* iff the weight $w(T)$ is minimum over the weights of all Steiner trees for S.

We generalize Steiner trees as follows. A *Steiner forest for* $S_1, S_2 \subseteq B$ is a family $\mathscr{F} = \{T_i : 1 \leq i \leq k\}$ of pairwise disjoint weighted trees $T_i = (V_i, E_i, w_i)$ such that $V_i \subseteq B$ and w_i coincides with Δ for all i with $1 \leq i \leq k$, and $S_1 \cup S_2 \subseteq \bigcup_i V_i$ and $S_i \cap V_j \neq \emptyset$, for $i = 1, 2$ and all j with $1 \leq j \leq k$. The weight $w(\mathscr{F})$ of $\mathscr{F}$ is defined as the weight of the graph that is the union of all trees in $\mathscr{F}$; $\mathscr{F}$ is optimal iff its weight is minimum over all Steiner forests for S_1 and S_2.

Note that if $\mathscr{F} = \{T\}$, then $\mathscr{F}$ is a Steiner forest for S and S' iff T is a Steiner tree for $S \cup S'$. In particular, if $S' = \{x\}$ and $x \in S$, then $\mathscr{F}$ is optimal iff T is optimal. The following holds for an optimal Steiner forest in the general case.

Proposition 6.2 *Let $\mathscr{F}$ be an optimal Steiner forest for S, S'. Then each $T = (V, E) \in \mathscr{F}$ is an optimal Steiner tree for $V \cap (S \cup S')$.*

Concerning the relation between Steiner forests and paths between sets, our first observation is that the cost of a path between sets S and S' is an upper bound for the weight of an optimal Steiner forest for S and S'.

Lemma 6.3 *Let $P = (P_1, \ldots, P_m) \in P(S, S')$ be a d_l-optimal path and let $\mathscr{F}$ be an optimal Steiner forest for S, S'. Then $w(\mathscr{F}) \leq c_{d_l}(P)$.*

Proof. (Sketch) Take an arbitrary sequence $L_1, \ldots, L_{m-1}$ of linkings L_i between P_i an P_{i+1}, and consider the undirected graph G whose edges correspond to the links in all L_i. Each connected component in G contains one point from S and one from S'. Thus any forest of spanning trees for the connected components in G is a Steiner forest for S, S'. It follows $w(\mathscr{F}) \leq c_{d_l}(P)$. □

Conversely, our next considerations show that the weight of an optimal Steiner forest for S, S' is an upper bound for the cost of any path between S and S' under d_l. Consequently, the cost of a d_l-optimal path between S and S' equals the weight of an optimal Steiner forest for S, S'.

6.2 Computing a path from a Steiner Tree

Given a Steiner tree $T = (V, E, w)$ for $S \cup S' \subseteq B$, $S, S' \neq \emptyset$, such that all its leaves, i.e. vertices of degree 1 are in $S \cup S'$, the algorithm **path** in Table 1 constructs a path $P \in P(S, S')$ such that $c_{d_l}(P) \leq w(T)$. The intuitive idea is that each step in the path takes one edge of the Steiner tree. We first do steps that remove each leaf of the tree that is in S. Then we select an arbitrary vertex x and an edge starting from that, and produce a next set on the path that in essence takes care of that edge.

Proposition 6.4 **path** (S, S', T) *computes $P = (P_1, \ldots, P_m)$ such that $P \in P(S, S')$ and $c_{d_l}(P) \leq w(T)$.*

Proof. (Sketch) It is not hard to see that the first **while** terminates (since the number of edges in T decreases in each iteration), and one can show that upon termination T is a tree such that all its leaves are in S'. After this, we clearly have $P_m \neq \emptyset$. The

Table 1. Algorithm for computing from T a path between S and S'

Algorithm path(S, S', T)

input: Steiner tree $T = (V, E, w)$ for $S \cup S' \subseteq B$,
$S, S' \neq \emptyset$, whose leaves are in $S \cup S'$.

output: $P = (P_1, \ldots, P_m) \in P(S, S')$ with $c_{d_l}(P) \leq w(T)$.

```
   /* Phase 1: move along edges of leaf nodes x ∉ S' */
P_1 := S; m := 1;
while T has a leaf x ∉ S' with an edge to y do
begin
   P_{m+1} := (P_m \ {x}) ∪ {y}; m := m + 1;          /* Link x to y */
   remove x from T;
end;  /* All leaves of T are in S' now */
   /* Phase 2: move towards S', and keep the reached nodes in R */
set R := {x} for any x ∈ P_m;
while E ≠ ∅ do begin
   select an x ∈ R with an edge to y;  R := R ∪ {y};
   if x ∉ S' and x is a leaf of T then
      P_{m+1} := (P_m \ {x}) ∪ {y};                   /* Link x to y */
   else  P_{m+1} := P ∪ {y};          /* Link x to x (cost 0) and y */
   m := m + 1;  remove edge {x, y} from T;
end;  /*  P_m = S' now */
if m = 1 then output(P_1, P_1) else output(P_1, ..., P_m);
```

vertex x selected from P_m serves as the root node of the remaining tree T, whose edges are removed one by one in the iterations of the second **while**. An edge $\{x, y\}$ can be removed only if $x \in R$, which holds after all edges on the path from the root to x have been removed. The second **while** terminates, again since the number of edges in T decreases in each iteration. Upon termination, $P_m = S'$ holds. Clearly, $m = |E| + 1$. It is easily checked that $m = 1$ iff $S = S' = \{v\}$ and $T = (\{v\}, \emptyset, w)$ for some v, hence $w(T) = 0$. If $m > 1$, then for each i with $1 \leq i < m$ there exists a linking L_i between P_i and P_{i+1} such that $c(L_i) = \Delta(x, y)$, where $\{x, y\}$ is the edge that has been removed from T when constructing P_{i+1}. Consequently, **path** outputs on input S, S' a path $P \in P(S, S')$ such that $c_{d_l}(P) \leq w(T)$. □

Lemma 6.5 *Let $\mathscr{F} = \{T_i = (V_i, E_i, w_i) : 1 \leq i \leq k\}$ be an optimal Steiner forest for S, S'. Then there exists a path $P \in P(S, S')$ such that $c_{d_l}(P) \leq w(\mathscr{F})$.*

Proof. (Sketch) This can be shown by induction on k. For $k = 1$, this follows from Proposition 6.4. In the general case, one can construct such a path by first going from $S \cap V_1$ to $S' \cap V_1$ while keeping the points in $S - V_1$ fixed, after that going from $S \cap V_2$ to $S \cap V_2'$ while keeping the points in $S' \cap V_1$ and $S - (V_1 \cup V_2)$ fixed and so on. The resulting path has cost $c_{d_l}(P) \leq w(\mathscr{F})$. □

Theorem 6.6 *Let $\mathscr{F}$ be an optimal Steiner forest for $S, S' \subseteq B$. Then $w(\mathscr{F}) = d_l^{\omega}(S, S')$.*

Proof. Follows immediately from Lemmas 6.3 and 6.5. □

6.3 An algorithm for computing d_l^ω and its complexity

The structural result in Theorem 6.6, combined with Proposition 6.2 is exploited first by the algorithm **dinflink** in Table 2. The subroutine **steiner**(S, B) called by this algorithm returns the weight of an optimal Steiner tree for S in B.

Table 2. Algorithm for computing d_l^ω

Algorithm dinflink(S, S')
input: $S, S' \subseteq B$, with $1 \leq |S| \leq |S'|$.
output: $d_l^\omega(S, S')$.

```
d := ∞;   /* upper bound on d_l^ω(S,S') */
for each partitioning B_1,...,B_k of S ∪ S' such that 1 ≤ k ≤ |S|
    and B_i ∩ S ≠ ∅, B_i ∩ S' ≠ ∅ for 1 ≤ i ≤ k do
begin   /* Σ_i w(T_i) of optimal Steiner trees T_i for B_i is ≥ d_l^ω(S,S') */
    w := 0;
    for i = 1 to k do w := w + steiner(B_i, B);
    d := min(d, w);
end;
 /* d has the cost of an optimal Steiner forest for S,S' */
output(d);
```

Proposition 6.7 dinflink *computes d_l^ω correctly.*

The generation of all $B_1, \ldots, B_k$, for each k, causes exponential cost of **dinflink**. However, as we will show, even if there is only one possible choice, computing d_l^ω can be an NP-hard problem. **dinflink** reflects this by calls of **steiner** for NP-hard instances.

On the other hand, under very general conditions computing d_l^ω is not "much harder" than the NP-complete problems, as one can show using the Steiner forest characterization. Recall that our motivation for defining d_l^ω was to extend a distance measure on B to a metric on $\mathscr{P}_\emptyset(B)$, which is only possible if Δ is a metric on B. In this case, the following holds on optimal Steiner trees.

Lemma 6.8 cf. [7] *Let $S \subseteq B$ be a nonempty subset of B. If Δ is a metric, then there exists an optimal Steiner tree (V, E, w) for S such that $|V| \leq 2|S| - 2$.*

Theorem 6.9 *Assume that each element of B can be represented in polynomial size and time with respect to the size of the input to d_l^ω, and that Δ is a metric.*[8] *Then deciding $d_l^\omega(S, S') \leq b$, where b is a rational number, is in NP.*

Proof. Let $\mathscr{F} = \{T_i = (V_i, E_i, w_i) : 1 \leq i \leq k\}$ be an optimal Steiner forest for S, S'. Then, $k \leq \min(|S|, |S'|)$. Since T_i is an optimal Steiner tree for $S_i = V_i \cap (S \cup S')$ and Δ is a metric, by Lemma 6.8 we have $|V_i| \leq 2|S_i| - 2$ for each i with $1 \leq i \leq k$. Hence, the number of vertices in $\mathscr{F}$, n_v, is bounded by

[8] Without this assumption on B, the result may not hold. E.g., if B consists of all Boolean formulas (likewise, first-order formulas, or trees) Φ of depth $\leq n$ on a given vocabulary, then the size of Φ is not necessarily polynomial in n or the size of other such formulas Φ_1 and Φ_2.

$$n_v = |\bigcup_{i=1}^{k} V_i| \leq 2\sum_{i=1}^{k}(|S_i| - 2) = 2|S \cup S'| - 2k \leq 2|S \cup S'| - 2.$$

Consequently, from Theorem 6.6 it follows that $d_l^\omega(S, S') \leq b$ iff there exists a Steiner forest $\mathscr{F}'$ for S and S' on at most n_v vertices. By our assumptions, the instance size of $\mathscr{F}'$ is bounded by a polynomial in the input size. Hence, guessing $\mathscr{F}'$, computing $w(\mathscr{F}')$, and comparing this value to b is possible in polynomial time. □

Corollary 6.10 *Computing $d_l^\omega(S, S')$ in the setting of Theorem 6.9 is NP-easy.*

Proof. $d_l^\omega(S, S')$ can be computed with an oracle for deciding $d_l^\omega(S, S') \leq b$ in polynomial time, by doing a binary search over the range of possible values. □

Notice that Theorem 6.9 and Corollary 6.10 generalize from metrics Δ to distance functions under which the instance size of an optimal Steiner forest for S and S' is bounded by a polynomial in the input size.

It is easy to see that computing d_l^ω is NP-hard even for simple cases.

Theorem 6.11 *Deciding whether $d_l^\omega(S, S') \leq b$ is NP-hard for subsets S, S' of the Euclidean integral plane with Manhattan metric and for integers b.*

Proof. This is shown by a reduction of the GEOMETRIC STEINER TREE problem (GST) [8] in the integral plane under the Manhattan (rectilinear) metric Δ_M. GST is as follows: Given a finite set $P = \{p_1, \ldots, p_n\}$ of points in the integral plane and an integer a, decide whether there exists under Δ_M a Steiner tree T for P such that $w(T) \leq a$. This problem is known to be NP-hard in the strong sense, i.e. even if all numbers are represented in unary notation [8]. It is easy to see that we may assume that in P only nonnegative coordinate values occur.

Now it suffices to note that there exists a Steiner tree T for P such that $w(T) \leq b$ if and only if $d_l^\omega(P, \{p_1\}) \leq b$. □

7 Conclusion

We have considered the problem of extending a distance function (or even metric) between points to a distance function or metric between point sets. We have investigated different approaches for this, and have analyzed the computational complexity of the resulting functions.

In our analysis, we assumed that the underlying set of points B is finite. However, in many cases B might be infinite. Under certain conditions, the results from above can be extended to this case as well. The functions d_{md}, d_f, d_{fs}, and d_l work well on finite subsets of B, but might not converge to a real number if one of the arguments is infinite; it is not straightforward how to overcome this problem. The function d^ω from Sect. 5, appropriately adapted by taking for (S, S') the infimum of $\{c_d(P) : P(S, S')\}$ rather than the minimum (which need not exist), yields a metric also for infinite B provided that no sequence of paths $P^1, P^2, \ldots$ between two different sets S and S' exists whose costs $c(P^1), c(P^2), \ldots$ converge to 0. In particular, Theorem 5.1 remains true in this case, and the metric infima of d_s, d_{fs} and d_l on the finite subsets of B collapse. An important fact is, however, that $d^\omega(S, S')$ may no longer take the cost of some optimal path between S and S', but rather the limit of the cost values of an infinite sequence of paths. This makes exact computation difficult if not impossible.

However, given e.g. that every set S has for every constant $c > 0$ only finitely many sets S' with $d(S, S') < c$ and that $d(S, S') > \alpha$ for some constant $\alpha > 0$ whenever S and S' are different, then $d^\omega(S, S')$ takes the cost value of an optimal path.[9] (This applies e.g. to the integral plane under a number of distance functions d on finite point sets from above.) Under this assertion, the computation of $d^\omega(S, S')$ reduces to a finite subset B' of B. For example, the computation of $d_l^\omega(S, S')$ in the integral plane trivially reduces to B' which contains only points x that are within distance $d_l(S, S')$ to some point in $S \cup S'$. The algorithms in Sect. 6 can thus be readily applied.

Besides proper extensions of the above distance functions to infinite point sets, several open problems remain.

We know that d_h, d_l^ω, and d_{md}^ω are not equivalent, but it is not quite clear what the properties of these metrics really are. Also, it would be interesting to know whether d_l^ω or d_{md}^ω can be approximated in polynomial time with reasonable performance bounds.

Acknowledgements. The authors would like to thank the referees for their comments and valuable suggestions that helped to improve the presentation of this paper.

References

1. H. Alt, K. Mehlhorn, H. Wagener, E. Welzl: Congruence, similarity and symmetries of geometric objects. Discrete Comput. Geom. **3**, 237–256 (1988)
2. M. R. Anderberg: Cluster Analysis for Applications. Academic Press, New York, 1973.
3. C. Berge: Graphs and Hypergraphs. Elsevier Science Publishers B.V. (North-Holland), 1989.
4. J. Dugundji: Topology. Allyn and Bacon, Inc., Boston, 1966.
5. U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (editors): Advances in Knowledge Discovery and Data Mining. AAAI Press, 1995. To appear.
6. P. Gärdenfors: Knowledge in Flux. Bradford Books, MIT Press, 1988.
7. M. Garey, R. Graham, D. Johnson: Some NP-complete Geometric Problems. In Proceedings STOC-78, pages 10–21, 1978.
8. M. Garey, D. S. Johnson: Computers and Intractability – A Guide to the Theory of NP-Completeness. W. H. Freeman, New York, 1979.
9. D. Huttenlocher, K. Kedem: Efficiently Computing the Hausdorff Distance for Point Sets under Translation. In Proceedings of the Sixth ACM Symposium on Computational Geometry, pp. 340–349 (1990)
10. K. Mehlhorn: Graph Algorithms and NP-Completeness, volume 2 of Data Structures and Algorithms. Springer, 1984.
11. S. Muggleton: Inductive Acquisition of Expert Knowledge. Addison Wesley, Reading, Massachusetts, 1990.
12. B. K. Natarajan: Machine Learning: A Theoretical Approach. Morgan Kaufmann, San Mateo, CA, May 1991.
13. I. Niiniluoto: Truthlikeness. D. Reidel Pub. Comp., Dordrecht, Holland, 1987.
14. G. Oddie: Verisimilitude and Distance in Logical Space. In I. Niiniluoto and R. Tuomela, editors, The Logic and Epistemology of Scientific Change, pages 243–264. North-Holland, 1979. (Acta Philosophica Fennica 30).
15. G. Oddie: Likeness to Truth. D. Reidel Pub. Comp., Dordrecht, Holland, 1986.
16. G. Piatetsky-Shapiro, W. J. Frawley (editors): Knowledge Discovery in Databases. AAAI Press / The MIT Press, Menlo Park, CA, 1991.
17. K. Popper: Some Comments on Truth and the Growth of Knowledge. In E. Nagel, P. Suppes, and A. Tarski, editors, Logic, Methodology, and Philosophy of Science, pp. 285–292 (1962)
18. K. Popper: A Note on Verisimilitude. British Journal for the Philosophy of Science **27**, 147–159 (1976)

[9] Examples show that neither of the two conditions can be dropped.

19. P. Z. Revesz: On the Semantics of Theory Change: Arbitration between Old and New Information. In Proceedings of the Twelth ACM SIGACT SIGMOD-SIGART Symposium on Principles of Database Systems (PODS-93), pages 71–79, June 1993.
20. H. Romerburg: Cluster Analysis for Researchers. Lifetime Learning, Belmont, CA, 1984.
21. H. Toivonen, M. Klemettinen, P. Ronkainen, K. Hätönen, H. Mannila: Pruning and grouping of discovered association rules. In MLnet Workshop on Statistics, Machine Learning, and Discovery in Databases, pages 47–52, Heraklion, Crete, Greece, April 1995.
22. R. Tuomela: Verisimilitude and Theory-Distance. Synthese **38**, 213–246 (1978)
23. P. Vaiyda: Geometry Helps in Matching. In Proceedings of the Twentiest ACM Symposium on the Theory of Computing (STOC-88), pages 422–425, 1988.
24. J. van Leeuwen: Graph Algorithms. In J. van Leeuwen, editor, Handbook of Theoretical Computer Science, volume A, chapter 10. Elsevier – North-Holland, Amsterdam 1990.
25. M. Winslett: Updating Logical Databases. Cambridge University Press, 1990.
26. S. Wrobel: On the Proper Definition of Minimality in Specialization and Theory Revision. In P. B. Brazdil, editor, Machine Learning: ECML-93. European Conference on Machine Learning, Lecture Notes in AI 667, pp 65–82. Springer-Verlag, 1993