



## List graphs and distance-consistent node labelings

Håkan Lennerstad, Mattias Eriksson

<sup>a</sup>*Dept. of Mathematics and Science, Blekinge Institute of Technology, S-37179 Sweden*

hakan.lennerstad@bth.se, mattias.eriksson@bth.se

### Abstract

In this paper we consider node labelings  $c$  of an undirected connected graph  $G = (V, E)$  with labels  $\{1, 2, \dots, |V|\}$ , which induce a *list distance*  $c(u, v) = |c(v) - c(u)|$  besides the usual graph distance  $d(u, v)$ . Our main aim is to find a labeling  $c$  so  $c(u, v)$  is as close to  $d(u, v)$  as possible. For any graph we specify algorithms to find a *distance-consistent labeling*, which is a labeling  $c$  that minimize  $\sum_{u, v \in V} (c(u, v) - d(u, v))^2$ . Such labelings may provide structure for very large graphs.

Furthermore, we define a labeling  $c$  fulfilling  $d(u_1, v_1) < d(u_2, v_2) \Rightarrow c(u_1, v_1) \leq c(u_2, v_2)$  for all node pairs  $u_1, v_1$  and  $u_2, v_2$  as a *list labeling*, and a graph that has a list labeling is a *list graph*. We prove that list graphs exist for all  $n = |V|$  and all  $k = |E| : n - 1 \leq k \leq n(n - 1)/2$ , and establish basic properties. List graphs are Hamiltonian, and show weak versions of properties of path graphs.

*Keywords:* graph labeling, graph distance, extremal combinatorics

Mathematics Subject Classification: 05C12, 05C45, 05C85

DOI: 10.5614/ejgta.2018.6.1.11

### 1. Introduction

In this paper, all graphs  $G = (V, E)$  with  $|V| = n$  and  $|E| = k$  are assumed to be simple, undirected and connected.

We refer to a bijection  $c(u)$  from  $V$  to  $\{1, 2, \dots, n\}$  as a  $V$ -labeling. Define the *list distance* as  $c(u, v) = |c(u) - c(v)|$ , and denote the usual graph distance by  $d(u, v)$ , i.e. the minimal number

Received: 21 April 2017, Revised: 24 December 2017, Accepted: 9 March 2018.

of edges for a path from  $u$  to  $v$ . A main aim of this study is to find a labeling  $c$  so  $c(u, v)$  is close to  $d(u, v)$ . This is a novel approach (see Section 2) with potential use both in graph theory and in various network application areas.

A strict condition for consistency between list distance and graph distance is that  $d(u_1, v_1) < d(u_2, v_2) \Rightarrow c(u_1, v_1) < c(u_2, v_2)$  for all nodes  $u_1, v_1, u_2, v_2$  - which we may call the strong list condition. It is fulfilled for a  $V$ -labeling only if  $G$  is a path graph  $P_n$ . In this paper we provide results answering to two different list-distance consistency problems:

1. (Section 3) If  $G$  has a *list labeling*, i.e. a  $V$ -labeling  $c$  fulfilling the weak list condition  $d(u_1, v_1) < d(u_2, v_2) \Rightarrow c(u_1, v_1) \leq c(u_2, v_2)$  for all nodes  $u_1, v_1, u_2, v_2$ , then  $G$  is called a *list graph* (Definition 2.5). We note some basic properties for list graphs, and prove that list graphs exist for all  $n$  and  $k : n - 1 \leq k \leq n(n - 1)/2$  (Theorem 3.3).
2. (Section 4) For any graph, a  $V$ -labeling  $c$  is *list-distance consistent* (Definition 2.1) if it minimizes  $\sum_{u,v \in V} (c(u, v) - d(u, v))^2$  within the set of  $V$ -labelings  $c$ . We provide algorithms to find list-distance consistent labelings for any graph (Theorem 4.3).

Furthermore, since minimizing  $\sum_{u,v \in V} (c(u, v) - d(u, v))^2$  is equivalent to maximizing  $d(A) = \sum_{i,j} |i - j| |a_{i,j}|$  (Lemma 4.1) by column permutations of  $G$ 's distance matrix  $A$ , it is possible to alternatively maximize  $d_p(A) = \sum_{i,j} |i - j|^p |a_{i,j}|$  for a suitable choice of  $p : 0 < p < \infty$ . List-distance consistent labelings are important also for list graphs, to explicitly find labelings.

A list labeling gives a Hamiltonian path, so the set of list graph constitute a middle set between general Hamiltonian graphs and path graphs. List graphs retain a version of the obvious list property of a path graph  $P_n$  of having no more than two ends. A graph that has three or more nodes of degree one it is not a list graph, and if it has two nodes of degree one, then if it is a list graph these nodes must have the labels 1 and  $n$  (see Theorem 3.2). Many list graphs have no nodes of degree one, such as a complete graph which is a list graph.

List graphs and distance-consistence labelings for any graph provide insight in the structure of distances between nodes in a graph in a way where all distances are considered simultaneously. Recall that the center of a graph consists of nodes  $u$  where  $\max_{v \in V} d(u, v)$  is minimal. Which nodes are center nodes is not directly affected by other distances in the graph than those realizing  $\max_{v \in V} d(u, v)$ . A comparison of the center of a graph with the nodes receiving close to  $(1 + |V|)/2$  labels in a distant-consistent labeling may be significant for understanding a specific graph's structure. This is potentially useful in facility location problems [20], [18] and in cluster analysis [3], [17].

Compared to other labelings, a list-distance consistent labeling may decrease or remove node labeling biases, which may provide clarity about fundamental network structure. In large and very large networks, a major problem is to identify important qualitative properties. A distance-consistent labeling may also help in dividing a large network in subnetworks in a relevant way.

Furthermore, comparing distant-consistent labelings may reveal new properties of well known graphs and allow new types of graph comparisons, as is discussed in Section 5.

## 2. Previous research and overview

The present approach is not investigated previously since few graph labelings aims at describing graph distances other than adjacencies, and those which do use other label sets than  $\{1, 2, \dots, |V|\}$ . The survey [7] describes over 200 different labelings of graphs, none of which has as main purpose to describe graph distances.

Adjacency labelings [2], [4], [5] is a way to assign labels to the vertices of an undirected graph so that given the labels of two vertices and no other information regarding the graph, it is possible to decide whether or not the vertices are adjacent. Results are presented concerning the minimal size of such labels.

A labeling scheme that is not restricted to the integers  $1, 2, \dots, |V|$  as labels, but concerned with graph distances, are proximity-preserving labelings [16]. Here the problem considered is to label the nodes so that it is possible to compute the distance between any pair of nodes directly from their labels. Again results describe minimal length of such labels for certain families of graph. In [8] such labeling schemes are called distance labeling schemes, and the results are generalized.

A related problem is considered in [11]. Here short labels are assigned to the nodes of a graph of  $n$  nodes in such a way that given the labels of any two nodes, one can decide whether they are  $k$ -vertex connected (not the same  $k$  as in this paper). If they are, there exist  $k$  vertex disjoint paths connecting the nodes. An upper bound  $k^2 \log n$  is established for the number of bits used in such a label. We also remark that a distance-consistent labeling certainly can be combined with other labelings.

Graph labelings are important in numerous applications, such as in [12] for communications networks, in [9] for radio channel assignment, in [10] for automation engineering, in [14] for data mining and fraud detection, in [19] for geographic navigation and internet routing, in [6] for moving object detection, in [15] for information visualization, in [13] for chemical inference, and in [1] for vehicle detection in traffic management.

### 2.1. Setup

Based on definitions from the introduction, we remark that the distance matrix  $A(G, c) = \{a_{ij} = d(c^{-1}(i), c^{-1}(j))\}$ , where  $c^{-1}$  is the inverse mapping of  $c : V \rightarrow \{1, 2, \dots, n\}$ , describes the graph distances in terms of the node labeling  $c$ . Note also that a change of node labeling is equivalent to a permutation of columns and rows of the symmetric matrix  $A$ .

This paper focuses on the following type of  $V$ -labelings.

**Definition 2.1.** For a graph  $G = (V, E)$ , a  $V$ -labeling  $c$  that minimizes

$$\sum_{u,v \in V} (c(u, v) - d(u, v))^2$$

among all possible  $V$ -labelings is called a **distance-consistent labeling** for  $G$ .

Note that the path graph  $P_n$  is the only graph where there is a  $c$  so that  $c(u, v) = d(u, v)$  for all vertices  $u, v$ , allowing the minimum to be zero. A path graph obviously has the same structure as a list of items, i.e. two distinct items with one neighbour, and all other have two neighbours.

Section 4 provides algorithms to find distance-consistent labelings for a graph  $G$ . The minimization problem is equivalent to maximizing the quantity  $\sum_{i,j} |i - j| a_{i,j}$  (Lemma 4.1). Other ways to measure the distance-consistency of a labeling, such as  $\sum_{u,v \in V} |c(u, v) - d(u, v)|$ , are discussed in Section 4, where we consider the generalized quantity  $\sum_{i,j} |i - j|^p a_{i,j}$  for any  $0 < p < \infty$ .

Note that for any matrix  $A$  with fixed sum  $\sum_{i,j} |a_{i,j}| > 0$ , the quantity  $d(A) = \sum_{i,j} |i - j| |a_{i,j}|$  is maximal with maximum  $(n - 1)(|a_{1n}| + |a_{n1}|)$  if all entries except  $a_{1n}$  and  $a_{n1}$  are zero. We refer to the positions  $(1, n)$  and  $(n, 1)$  as the *matrix corners*. We define as follows, with  $p = 1$  as main case.

**Definition 2.2.** For  $p : 0 < p < \infty$ , the  *$p$ -diagonal dispersion* of a non-zero matrix  $A$  is

$$d_p(A) = \sum_{i,j} |i - j|^p |a_{i,j}|,$$

and the *normalized  $p$ -diagonal dispersion* is

$$D_p(A) = \frac{\sum_{i,j} |i - j|^p |a_{i,j}|}{(n - 1)^p \sum_{i,j} |a_{i,j}|}.$$

We denote  $d(A) = d_1(A)$  and  $D(A) = D_1(A)$ . Note that  $D_p(A)$  fulfills  $0 \leq D_p(A) \leq 1$  for any matrix  $A$ . The quantity  $D_p(A)$  is relevant when comparing matrices, but  $d_p(A)$  is usually more practical when working with column permutations of a fixed matrix. Large exponent  $p$  favours a choice of many small differences  $|c(u, v) - d(u, v)|$  of  $A$ , while small  $p$  favours few large differences.

Note that the quantity  $C(A) = 1 - D(A)$ ,  $0 \leq C(A) \leq 1$ , can be seen as a diagonal concentration, since  $A$  is diagonal  $\Leftrightarrow C(A) = 1$ , and  $C(A) \geq \frac{1}{2}$  if  $A$  is diagonally dominant ( $\sum_{i:i \neq j} |a_{i,j}| \leq a_{ii}$  for all  $j$ ). Diagonal concentration  $C(A)$  has potential relevance also in numerical linear algebra.

For maximizing the diagonal dispersion, there is a particularly easy special case.

**Definition 2.3.** A matrix  $A$  is *monotone* if

$$\begin{aligned} a_{i,j} &\leq a_{i,j+1} \text{ for all } j : i \leq j \leq n - 1, \\ a_{i,j} &\geq a_{i,j+1} \text{ for all } j : 1 \leq j \leq i - 1, \\ a_{i,j} &\leq a_{i+1,j} \text{ for all } i : j \leq i \leq n - 1, \\ a_{i,j} &\geq a_{i+1,j} \text{ for all } i : 1 \leq i \leq j - 1. \end{aligned}$$

If the distance matrix is monotone, the graph is described by a distance-consistent labeling, since no permutation of rows and columns can increase  $d_p(A)$  for any  $p > 0$ . Monotonicity for a distance matrix  $A$  follows from the following stronger condition in terms of a labeling.

**Definition 2.4.** A  $V$ -labeling  $c$  of a connected graph  $G = (V, E)$  is a **list-distance labeling**, or a **list labeling**, if

$$d(u_1, v_1) < d(u_2, v_2) \Rightarrow c(u_1, v_1) \leq c(u_2, v_2)$$

for all nodes  $u_1, v_1, u_2, v_2 \in V$ .

Thus, for a list labeling only, there are no **list-distance contradictions**, by which we mean two pairs of nodes  $u_1, v_1$  and  $u_2, v_2$  so that  $d(u_1, v_1) < d(u_2, v_2)$  but  $c(u_1, v_1) > c(u_2, v_2)$ . The graph  $L$  given before Theorem 3.3 in Section 3 is an example of a non-list graph that has a monotone distance matrix - monotonicity is a slightly weaker condition.

**Definition 2.5.** A **list-distance graph**, or a **list graph**, is a graph that has a list labeling.

In adjacency matrix terms, the list graphs is the subclass of graphs that can be labeled to achieve a monotone matrix. Examples of non-list graphs are the star graph  $S_n$  and the cycle graph  $C_n$ ,  $n \geq 4$  (see Section 3). Extremal cases of list graphs are the path graph  $P_n$  and the complete graph  $K_n$ . In the latter case all labelings are list labelings.

For any connected graph and a  $V$ -labeling  $c$ , we may define a measure of how much the labeling deviates from a list:

**Definition 2.6.** The **list discrepancy** is  $l(G, c) = \max_{u,v \in V} |c(u, v) - d(u, v)|$ .

For  $P_n$  and  $K_n$  we have  $l(P_n, c) = 0$  and  $l(K_n, c) = n - 2$  if  $c$  is a list labeling.

Note that the diagonal dispersion for a distance matrix of a connected graph is maximal for  $P_n$  which has  $D(A(P_n)) = \frac{1}{2} \frac{n}{n-1} \searrow \frac{1}{2}$  as  $n \rightarrow \infty$ . As a comparison  $D(A(K_n)) = \frac{1}{3} \frac{n+1}{n-1} \searrow \frac{1}{3}$  as  $n \rightarrow \infty$ . Hence:

**Lemma 2.1.** The normalized diagonal dispersion for the distance matrix of a graph  $G$  with a list labeling and  $n$  nodes fulfill

$$\frac{1}{3} \frac{n+1}{n-1} \leq D(A(G)) \leq \frac{1}{2} \frac{n}{n-1}.$$

### 3. Properties of list graphs

We denote by  $H = (U, F)$  the induced subgraph of  $G = (V, E)$ , i.e. the edge set  $F$  contains all edges in  $G$  with both endpoints in  $U$ .

**Definition 3.1.** A labeling  $c : \{u, v, w\} \rightarrow \mathbb{N}$  of  $P_3 = (\{u, v, w\}, \{uv, vw\})$  is consistent if  $c(u) < c(v) < c(w)$  or  $c(u) > c(v) > c(w)$ .

The following lemma follows trivially.

**Lemma 3.1.** Consider a  $V$ -labeling  $c$  of  $G$ . If a path graph  $P_3$  is a subgraph of  $G$  and  $c$  restricted to  $P_3$  is not consistent, then the labeling is not a list labeling of  $G$ .

So, in a list labeling, all subpaths  $P_3$  must be labeled consistently.

- Theorem 3.2.**
1.  $G$  is not a list graph if it contains a cycle graph  $C_n$  for any  $n \geq 4$  as an induced subgraph.
  2.  $G$  is not a list graph if it contains a star graph  $S_n$  for any  $n \geq 3$  as an induced subgraph.
  3.  $G$  is not a list graph if it contains three vertices of degree one.
  4. If  $G$  is a list graph and has two vertices  $u$  and  $v$  of degree one, then  $c(u) = 1$  and  $c(v) = n$ , or  $c(u) = n$  and  $c(v) = 1$ .
  5. If  $G$  is a list graph, it is Hamiltonian.

*Proof.* 1. Consider a subgraph  $C_n$  where the nodes are labeled by distinct numbers  $c(v)$ . Then there must be a node  $u$  with maximal  $c(u)$  among the nodes in  $C_n$ . Then  $u$  together with its two neighbours in  $C_n$  form a subgraph  $P_3$  that is not labeled consistently.

2. A star graph  $S_3 = (\{u, v, w, x\}, \{ux, vx, wx\})$  has three distinct path graphs  $P_3$  that are subgraphs. It is easy to check that all three cannot be labeled consistently.

3. Assume that the three vertices  $\{u, v, w\}$  that has degree one has the neighbours  $\{u', v', w'\}$  respectively, and assume that  $c(u) < c(v) < c(v') < c(w)$ . Then  $d(u, v') < d(u, v)$  but  $|c(u) - c(v')| > |c(u) - c(v)|$ . The other cases are proven similarly.

4. Assume that  $u'$  is the neighbour vertex of  $u$  and consider the case that  $c(u) = 2$ . Then if  $c(u') = 1$ , the pairs  $u, w$  and  $u', w$  contradicts that  $G$  is a list graph, using any other node  $w$ . If  $c(v) = 1$  where  $v \neq u'$ , the pairs  $u, v$  and  $u', v$  provide a contradiction. The other cases are proven similarly.

5. The labeling  $c$  provides a Hamiltonian path since the nodes  $u = c^{-1}(i)$  and  $v = c^{-1}(i + 1)$  must be adjacent. If not, nodes adjacent to  $u$  must be labeled  $i - 1$  and to  $v$  by  $i + 2$ , i.e.  $u$  and  $v$  must have degree 1. By continuing this argument with the next adjacent nodes, finally the nodes meet since the graph is connected, and we have a contradiction.  $\square$

Certainly, statements 2 and 3 of the theorem follow from  $G$  being Hamiltonian. Statements 3 and 4 suggest that like a full list like  $P_n$ , also a list graph retains at most two ends, if there are nodes of degree one. The main problem of constructing list graphs for any  $n$  and  $k$  is that the edges need to be distributed rather evenly, in a specific sense. An example of a non-list graph of a different type than those of Theorem 3.2 is the graph  $N = (V, E)$  with  $V = \{1, 2, 3, 4, 5, 6, 7\}$  and  $E = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 6\}, \{6, 7\}, \{1, 3\}, \{3, 5\}\}$ , i.e. the path graph  $P_7$  where the two edges  $\{1, 3\}$  and  $\{3, 5\}$  have been added. It is not a list graph since  $c(1, 5) = 4$  and  $d(1, 5) = 2$ , but  $c(4, 7) = d(4, 7) = 3$ . However, the distance matrix of  $L$  is monotone.

Replacing the edge  $\{3, 5\}$  with  $\{5, 7\}$  gives a list graph.

The existence question can however be answered affirmatively for all  $n$  and  $k$ , where the proof can be seen as a generalization of this example.

**Theorem 3.3.** For any integers  $n \geq 2$  and  $k : n - 1 \leq k \leq n(n - 1)/2$ , there exists a list graph with  $n$  nodes and  $k$  edges.

*Proof.* In this proof we start with a  $V$ -labeling  $c$  of the nodes  $1, \dots, n$ , and construct a list graph with that labeling by adding  $k$  edges to the nodes, for any  $k : n - 1 \leq k \leq n(n - 1)/2$ .

For any  $i$ , an edge  $(i, i + a)$  will be referred to as an  $a$ -edge. Note that the number of 1-edges is  $n - 1$ , the number of 2-edges is  $n - 2$  and so on, so the total number of 1-edges, 2-edges, up to

all  $a$ -edges is  $na - a(a + 1)/2$ . Given  $k$  edges to distribute, we first add all 1-edges, 2-edges up to all  $(a - 1)$ -edges to the graph, until we reach the smallest  $a$  so that  $k \leq na - a(a + 1)/2$ . Then  $a = \left\lceil n - \frac{1}{2} - \frac{1}{2}\sqrt{(2n - 1)^2 - 8k} \right\rceil$ . Then  $b = k - n(a - 1) - a(a - 1)/2$  edges remain, which all are  $a$ -edges. These  $b$  edges will in this proof be distributed in a way that provides a list graph. Thus, all  $(a - 1)$ -edges,  $(a - 2)$ -edges, ... , 2-edges and 1-edges belong to all graphs in this proof. We may assume that  $a \geq 2$  since the graph is connected and the case  $a = 1$  is the graph of only 1-edges, which is  $P_n$ , which is a list graph

In all cases we consider a pair  $u, v$  with  $c = c(u, v)$  and a pair  $w, z$  with  $c(w, z) = c - 1$ , and construct a graph where  $\min d(u, v) \geq \max d(w, z)$  for all such pairs, and for any  $c$ . Then the graph is a list graph. Denote a shortest path from  $u$  to  $v$  a  $c$ -path. As in the example preceding the theorem, where  $a = 2$ , a critical case is a  $c$ -path which uses 2  $a$ -edges more than another  $c$ -path  $(c - 1)$ -path. We show that this cannot happen in any of the graphs described in this proof.

The simplest case occurs when  $k = na - a(a + 1)/2$  and thus  $b = n - a$ . Then all possible  $a$ -edges belong to the graph, and it is a list graph since  $d(u, v) = \left\lceil \frac{c}{a} \right\rceil$  for any  $c = c(u, v)$ , so  $\min d(u, v) = \left\lceil \frac{c}{a} \right\rceil \geq \left\lceil \frac{c-1}{a} \right\rceil = \max d(w, z)$ .

In the more complicated cases we need to refer to a set of  $a$  consecutive  $a$ -edges

$$\{\{i, i + a\}, \{i + 1, i + a + 1\}, \dots, \{i + a - 1, i + 2a - 1\}\}$$

as a *complete cluster*. These nodes have no other edges, and have the advantage that a  $c$ -path passing it only uses one  $a$ -edge, for any  $c$ . A complete cluster consists of  $a$  edges and involves  $2a$  nodes. An *incomplete cluster* has  $j < a$  consecutive  $a$ -edges  $\{\{i, i + a\}, \{i + 1, i + a + 1\}, \dots, \{i + j - 1, i + a + j - 1\}\}$  for some  $j : 2 \leq j \leq a - 1$ , and involves  $a + j$  nodes. The *starting node* of a cluster, complete or incomplete, is node  $i$ .

**Case 1.** If  $b \leq a$  a list graph is produced by forming a cluster of all  $a$ -edges. Then  $u, v$  with  $c = c(u, v)$  minimizes  $d(u, v)$  when using one cluster edge and otherwise  $(a - 1)$ -edges, so  $\min d(u, v) = 1 + \left\lceil \frac{c-a}{a-1} \right\rceil = \left\lceil \frac{c-1}{a-1} \right\rceil$ . The pair  $w, z$  maximizes  $d(u, v)$  if it can avoid a cluster edge, so  $\max d(w, z) \leq \left\lceil \frac{c-1}{a-1} \right\rceil$ . We get  $\min d(u, v) \geq \max d(w, z)$ , so this graph is a list graph for all  $n$ .

**Case 2.** Now suppose that  $a < b \leq 2a$ . If  $n \geq 2a + b$  we may construct a list graph with two clusters, one complete cluster and one incomplete, where the complete cluster has  $2a$  nodes, and the incomplete  $b$  nodes. The complete cluster starts at node  $t + 1$  and the incomplete cluster at node  $j = n - t - b + 1$ . The two subgraphs containing the consecutive nodes  $\{1, 2, \dots, t\}$  and  $\{n - t + 1, \dots, n\}$ , respectively, will be referred to as the *tails* of the graph. Thus, both tails has  $t$  nodes. The total number of tail nodes,  $2t$ , will be referred to as the *tail space*, also in other cases in this proof.

Denote the number of nodes between the cluster nodes - the distance between the clusters - by  $e$ . Then if  $c(u, v) = c < e + 2a - 1$  only one cluster can be used, and we are in Case 1. If  $c(u, v) = c \geq e + 2a - 1$  we get  $\min d(u, v) = 2 + \left\lceil \frac{c-2a}{a-1} \right\rceil = \left\lceil \frac{c-2}{a-1} \right\rceil$ . For a pair  $w, z$  with  $c(w, z) = c - 1$  we obtain  $\max d(w, z) = \left\lceil \frac{c-1}{a-1} \right\rceil$  if there exists a pair  $w$  and  $z$  that do not use any of the cluster edges, which would imply that the graph is not a list graph. However, such a pair  $w, z$  with  $c(w, z) \geq e + 2a - 2$  does not exist if  $e + 2a - 2 > t + a - 1$  - i.e. if the tail space is small enough. So by choosing  $t \leq a + e$ , placing the clusters close enough to the nodes 1 and  $n$ , the graph is a list graph.

The following cases generalize this situation, where we for all  $b$  and  $n$  find similar graphs with  $t \leq a + e$ .

If  $n < 2a + b$  there are too few nodes to form two clusters. Then we start by constructing two clusters in the far ends of the graph with  $\lfloor n/2 \rfloor - a$  and  $\lceil n/2 \rceil - a$  edges, respectively. The remaining edges to add are  $\{i, i + a\}$  for  $i = \lfloor n/2 \rfloor - a + 1, \dots, \lfloor n/2 \rfloor + b - n - 3a$ . In this graph, no  $c$ -path can use 2  $a$ -edges more than any  $(c - 1)$ -path, so it is a list graph.

Hence also in Case 2 a list graph can be constructed for all  $n$ .

**Case 3.** Suppose next that  $b > 2a$  and  $\lfloor b/a \rfloor \leq 2(a + e) + 1$ . We here construct as many complete clusters as possible, i.e.  $\lfloor b/a \rfloor$  complete clusters and possibly one incomplete with  $(b - a \lfloor b/a \rfloor)$  edges, which is the last cluster, say next to the right tail. The distance between any pair of neighbouring clusters is  $e$ . Hence, the distance from a starting node of a cluster to the next cluster's starting node is  $2a + e$ . Then we can take advantage of the proof of Case 2 since increasing  $c$  by  $i(2a + e)$  for some  $i \geq 1$  implies the same change for the pair  $u, v$  as for the pair  $w, z$ , preserving the difference  $d(u, v) - d(w, z)$  as in Case 2.

Denote the number of complete clusters by  $f = \lfloor b/a \rfloor$ . Then for  $n = af + a + b$  we have a list graph with  $e = 0$ . Now we can add nodes to the tails as long as  $t \leq a + e$ , providing list graphs for  $n = af + a + b + i$  for  $i = 0, \dots, 2(a + e)$ . Next we may increment  $n$  further by incrementing  $e$ , i.e. incrementing all  $f - 1$  distances between clusters, and simultaneously reducing the number of nodes in the tails. This is possible if  $f - 1 \leq 2(a + e)$ , so in Case 3 we have list graphs for all  $n$ .

**Case 4.** For  $b > 2a$ ,  $f = \lfloor b/a \rfloor > 2(a + e) + 1$  and large  $n$ , we produce a graph consisting of a set of complete clusters, one incomplete cluster, and a set of single edges, which we for simplicity call single clusters. We choose the same distance between the starting points of all clusters, complete, incomplete and single:  $2a + e$ . This graph is a list graph by the same arguments as in Case 3.

If one edge is extracted from a cluster and made into a single cluster, the size of the tail-free graph increases by at most  $a + e$ . Since the total allowed tail space  $2(a + e)$  is larger than this, we can always extract edges and form single clusters to construct a list graph. For very large  $n$ , also  $e$  need to be increased. Thus there exist list graphs for all  $n$  also in Case 4.

Since the cases cover all  $n$ ,  $a$  and  $b$ , the proof is complete. □

This proof demonstrates rather well some characteristics of list graphs.

#### 4. Finding a distance-consistent labeling

This section describes algorithms that produce distance-consistent labelings for any simple, undirected connected graph.

**Lemma 4.1.** *A labeling  $c$  minimizes*

$$\sum_{u,v \in V} (c(u, v) - d(u, v))^2$$

*if and only if it maximizes*

$$\sum_{i,j} |i - j| a_{i,j}.$$

*Proof.* This follows immediately by a change of notation and expanding the square:

$$\begin{aligned} \sum_{u,v \in V} (c(u,v) - d(u,v))^2 &= \sum_{i,j=1}^n (|i-j| - a_{ij})^2 \\ &= \frac{1}{6}n^2(n^2 - 1) + \sum_{i,j=1}^n a_{ij}^2 - 2 \sum_{i,j} |i-j| a_{i,j}. \end{aligned}$$

In the last expression, the two first terms are independent of graph labelings, from which the lemma follows. □

If the graph is a list graph, the distance-consistent labeling found by the algorithm to be presented in this section is a list labeling. This follows from Lemma 4.1 and the fact that only list graphs have a monotone distance matrix, i.e. that is free from list-distance contradictions.

We have defined a list-distance contradiction as two pairs of nodes  $u_1, v_1$  and  $u_2, v_2$  so that  $d(u_1, v_1) < d(u_2, v_2)$  but  $c(u_1, v_1) > c(u_2, v_2)$ . *The size of a contradiction* is the quantity  $c(u_1, v_1) - c(u_2, v_2)$ . In some instances a labeling with few large contradictions may be preferred, and in other many small contradictions, a choice that can be made by choosing an appropriate value of  $p$  in  $D_p(A)$ .

In the case  $C_4$ , for example,  $p > 1$  favours several small contradictions, while  $p < 1$  favours labelings with few large contradictions. Note that since a list graph has a monotone distance matrix, maximization of  $d_p(A)$  give the same distance-consistent labelings, which are list labelings, for all  $p > 0$ .

Each step in the algorithm, described in Theorem 4.3, switches the labels between two nodes  $m$  and  $l$ , which means switching distance matrix  $A$  into  $A'$ , where

$$\begin{aligned} a'_{mj} &= a_{lj} \text{ and } a'_{lj} = a_{mj} && \text{for all } j \\ a'_{im} &= a_{il} \text{ and } a'_{il} = a_{im} && \text{for all } i \\ a'_{ij} &= a_{ij} && \text{otherwise.} \end{aligned}$$

Such a row and column switch is here for simplicity called a *label transposition*  $m, l$  of  $A$ .

**Definition 4.1.** *The gain in diagonal dispersion  $d_p(A)$  is defined as*

$$g_{ml} = \sum_{i=1}^n (|m-i|^p (a_{il} - a_{im}) + |l-i|^p (a_{im} - a_{il})).$$

*The gain matrix  $G$  has the entries  $g_{ml}$ ,  $1 \leq m, l \leq n$ .*

The gain matrix  $G$  is obviously symmetrical and has zeroes in the main diagonal. If  $g_{ml} > 0$ , the label transposition  $m, l$  increases the diagonal dispersion to  $d_p(A') = d_p(A) + g_{ml}$ .

Thus, if all  $g_{ml} \leq 0$ , no label transposition can improve the diagonal dispersion. The next lemma, Lemma 4.2, is a preparation in more general terms for Theorem 4.2, which establishes that the  $\binom{n}{2}$  permutations are enough to consider, so the entire set of  $n!$  permutations must not be investigated to find maximal diagonal dispersion.

Consider now a matrix  $A$  of type  $m \times n$  and permutations of the columns of  $A$ . Denote by  $p(A)$  the matrix  $A$  subjected to a column permutation  $p$ . For given non-negative weights  $w_{ij}$ , we want to find a permutation  $p$  so that  $f(p(A)) = \sum_{i=1}^m \sum_{j=1}^n w_{ij} a_{i,p(j)}$  is maximal. It turns out that a value of  $f$  that cannot be improved by any transposition cannot be improved by any permutation.

**Lemma 4.2.** *Suppose that  $A$  is a matrix of type  $m \times n$  and  $w_{ij}$   $1 \leq i \leq m, 1 \leq j \leq n$  are non-negative real numbers. Consider the problem of finding a maximum of the linear function  $f(A) = \sum_{i=1}^m \sum_{j=1}^n w_{ij} a_{ij}$  by permuting columns in  $A$ . Then if*

$$f(A) \geq f(t(A))$$

for all column transpositions  $t$ , then

$$f(A) \geq f(p(A))$$

for all column permutations  $p$ .

*Proof.* Consider first the case  $m = 1$ , when  $f(A) = \sum_{j=1}^n w_j a_j$ . Then the transposition condition says that

$$(w_i - w_j)(a_i - a_j) \geq 0$$

for all  $1 \leq i, j \leq n$ . We assume without loss of generality that the  $w_i$ :s are in size order:  $w_i \geq w_{i+1}$  for all  $i : 1 \leq i \leq n - 1$ . Then it follows that also  $a_i \geq a_{i+1}$  for all  $i : 1 \leq i \leq n - 1$ .

Consider now a permutation  $c = (m_1, \dots, m_n)$  of  $(1, \dots, n)$ . It is enough to prove that

$$\sum_{j=1}^n w_j a_j \geq \sum_{j=1}^n w_j a_{m_j}, \text{ i.e.}$$

$$\sum_{j=1}^n w_j (a_j - a_{m_j}) \geq 0.$$

We will show that it is possible to group the  $n$  terms of  $\sum_{j=1}^n w_j (a_j - a_{m_j})$  in a way so that the sum of each group is non-negative.

The only possibly non-positive terms are  $w_j (a_j - a_{m_j})$  where  $j > j_m$ . Denote such an index by  $j = j^1$ , so  $w_{j^1} (a_{j^1} - a_{m_{j^1}}) < 0$ . Then we find the index  $m_{j^2} = j^1$  and the term  $w_{j^2} (a_{j^2} - a_{m_{j^2}})$ .

Now there are three cases. If  $j^2 = m_{j^1}$  we have  $w_{j^2} (a_{j^2} - a_{j^1}) + w_{j^1} (a_{j^1} - a_{j^2}) = (w_{j^2} - w_{j^1})(a_{j^2} - a_{j^1}) \geq 0$ . Otherwise we calculate

$$\begin{aligned} w_{j^2} (a_{j^2} - a_{m_{j^2}}) + w_{j^1} (a_{j^1} - a_{m_{j^1}}) &\geq w_{j^1} (a_{j^2} - a_{m_{j^2}}) + w_{j^1} (a_{j^1} - a_{m_{j^1}}) \\ &= w_{j^1} (a_{j^2} - a_{m_{j^1}}). \end{aligned}$$

If  $j^2 < m_{i^1}$ , this term is positive. We then turn to the next negative term  $w_j(a_j - a_{m_j})$  in the same way as the procedure described for  $j^1$ . If  $j^2 > m_{i^1}$  we find the index  $m_{j^3} = j^2$  and the term  $w_{j^3}(a_{j^3} - a_{m_{j^3}})$ , similarly producing the term  $w_{j^1}(a_{j^3} - a_{m_j})$ . We are then back to the situation with tree cases. One of the two first cases must happen since each orbit of a permutation ends where it started, and the orbits are disjoint.

Now consider the case  $m > 1$ . With the notation  $W_i = \sum_{j=1}^n w_{ij}$  and  $A_i = \sum_{j=1}^n w_{ij}$ , the transposition condition gives  $(W_i - W_j)(A_i - A_j) \geq 0$  for all  $i$  and  $j$ , and thus the proof can be completed in the same way as for  $m = 1$ . The proof is complete.  $\square$

**Theorem 4.2.** *Suppose that  $c^*$  is a node labeling where all  $g_{ml} \leq 0$ . Then*

$$d_p(c^*, A) \geq d_p(c, A)$$

for all node labelings  $c$ .

*Proof.* Aiming at Lemma 4.2 we choose  $w_{ij} = |j - i|^p$ ,  $m = n$ . However, in contrast to the column permutations of the lemma, label permutations involve simultaneous row permutations. Since  $w_{ij} = w_{ji}$  and  $a_{ij} = a_{ji}$ , the row transposition can be transformed into a column transposition, and the statement follows from Lemma 4.2 for  $m = 1$ .  $\square$

Note that if all  $g_{ml} < 0$  the distance-consistent labeling is unique. Otherwise, label transpositions  $m, l$  where  $g_{ml} = 0$  produces all distance-consistent labelings.

The following theorem summarizes the algorithm for finding distance-consistent labelings.

**Theorem 4.3.** *A maximally distant-consistent labeling is reached by in each step strictly increasing the diagonal dispersion  $d_p(c, A)$  by effectuating one of the  $\binom{n}{2}$  transpositions  $t$  of columns  $m$  and  $l$  in  $A$  with  $g_{ml} > 0$ , producing the matrix  $A' = t(A)$ .*

*If one of  $i$  or  $j$  equals  $m$  or  $l$ , the new quantities in  $G'$  in terms of entries in  $A$  are*

$$g'_{ij} = g'_{ij} = \sum_{m=1}^n (|m - i|^p (a_{mj} - a_{mi}) + |m - j|^p (a_{mi} - a_{mj})).$$

*For all other  $i$  and  $j$  we have*

$$g'_{ij} = g_{ij} + |l - j|^p (a_{mj} - a_{lj}) + |m - j|^p (a_{lj} - a_{mj}) + |l - i|^p (a_{mi} - a_{li}) + |m - i|^p (a_{li} - a_{mi}).$$

*Each step  $A, G \rightarrow A', G'$  requires at most  $4n^2$  multiplications.*

*Proof.* The formulas of the theorem follows directly from Definition 4.1. The first formula requires  $2n(2n - 1)$  multiplications, the second  $4(n - 1)^2$ , totally slightly less than  $8n^2$ . By the symmetry of matrices, the estimate  $4n^2$  multiplications follows.  $\square$

## 5. Conclusions and open questions

List graphs and distance-consistence labelings provide insight in the entire distance structure in a graph in terms of local information: the label of each node. Studying similar problems for weighted and directed graphs can be expected to provide concept versions closer to networks in applications.

Distance-consistence labelings quantifies how close a graph is to a path graph, i.e. how close the graph structure is to a list structure. For example, if we define  $N(G)$  as the minimal number of list-distance contradictions taken over all labelings  $c$  on a graph  $G = (V, E)$ , the number  $N(G)$  provides a categorization of all simple graphs, where list graphs only have  $N(G) = 0$ . The maximal possible  $N(G)$  for a certain  $n = |V|$  and  $k = |E|$  is not yet known, although star graphs of playing this role.

A different obvious measure on the "listness" of a graph  $G$  is the quantity

$$L(G) = \min_c \left( \sum_{u,v \in V} (c(u, v) - d(u, v))^2 \right),$$

which is zero only for path graphs. Non-trivial bounds for

$$\begin{aligned} L_{\max}(n, k) &= \max_{G:|E|=k} L(G) \text{ and} \\ L_{\min}(n, k) &= \min_{G:|E|=k} L(G) \end{aligned}$$

are not known, nor which graphs that realize the maximas and minimas. It is however easy to check that

$$L_{\max}(2n + 1, 2n) = L(S_{2n+1}) = \frac{1}{3}n(n - 1)(8n^2 - 10n + 15),$$

and

$$L(K_n) = \frac{1}{12}n(n - 1)^2(n - 2).$$

A natural conjecture is that  $L_{\min}(n, k) = L(G)$  if  $G = (V, E)$  is a list graph for all  $|V| = n$  and  $|E| = k > n - 1$ .

- [1] A. Ali and S. Afghani, Shadow based on-road vehicle detection and verification using Haar wavelet packet transform, *Proceedings of the First International Conference on Information and Communication Technologies (ICICT 05)* (2005).
- [2] S. Alstrup, H. Kaplan, M. Thorup and U. Zwick, Adjacency labeling schemes and induced-universal graphs, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, (2015), pp 625-634.
- [3] P. Arabie, L.J. Hubert and G. De Soete, *Clustering and Classification*, World Scientific Publishing, Singapore, 1999.
- [4] M.A. Brewer, Coding the vertex of a graph, *IEEE Trans. of Info. Theory* **12** (1966), 148–153.

- [5] M.A. Brewer and J. Folkman, An unexpected result on coding vertices of a graph, *J. Math. Anal. Appl.* **20** (1967), 583–600.
- [6] R. Fablet, P. Bouyhemy and M. Gelgon, Moving object detection in color image sequences using region-level graph labeling, *Proceedings of International Conference on Image Processing (ICIP 99)* **2** (1999), 939–943.
- [7] J.A. Gallian, A dynamic survey of graph labeling, *Electronic Journal of Combinatorics* **18** (2011).
- [8] C. Gavoille and D. Peleg, Pérennes S., Raz R., Distance labeling in graphs, *Proceedings of SODA'01*, Philadelphia, PS, (2001), 210–219.
- [9] J. van den Heuvel, R.A. Leese and M.A. Shepherd, Graph Labeling and Radio Channel Assignment, *J. Graph Theory* **29** (4) (1998), 263–283.
- [10] G.W. Klau and P. Mutzel, Combining Graph labeling and compaction, *Lecture Notes in Computer Science* **1731**, Springer, Berlin, Heidelberg (1998), 27–37.
- [11] A. Korman, Labeling schemes for vertex connectivity, *ACM Trans. Algorithms* **6** (2), (2010), 39:1-39:10.
- [12] N.L. Prasanna, Application of Graph Labeling in Communication Networks, *Oriental Journal of Computer Science and Technology* **7** (1), (20).
- [13] L.M. Masinter, N.S. Sridharan, J. Lederberg and D.H. Smith, Applications of artificial intelligence for chemical inference. XII. Exhaustive generation of cyclic and acyclic isomers, *J. Am. Chem. Soc.*, **96** (25), (1974), 7702–7714.
- [14] M. McGlohon, M.G. Anderle, D.M. Stele and C. Faloutsos, SNARE: a link analytic system for graph labeling and risk detection, *KDD '09 Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, (2009), 1265–1274.
- [15] G. Neyer, Map labeling with application to graph drawing, *Lecture Notes in Computer Science* **2025**, Springer, Berlin, Heidelberg, (2001), 247–273.
- [16] D. Peleg, Proximity-preserving labeling schemes and their applications, Graph-theoretic Concepts in Computer Science, *The 25th International Workshop, Lecture Notes in Computer Science* **1665**, Springer, Berlin, Heidelberg, (1999), 30–41.
- [17] C.J. Stam and J.C. Reijneveld, Graph theoretical analysis of complex networks in the brain, *Nonlinear Biomed. Phys.* **1** (1), (2007).
- [18] T. Thai and P.M. Pardalos, *Handbook of Optimization in Complex Networks: Theory and Applications*, Springer, Berlin, Heidelberg (2012).

- [19] H. Wang, H. He, J. Yang, P.S. Yu and J.X. Yu, Dual Labeling: Answering Graph Reachability Queries in Constant Time, *Proceedings of 22nd International Conference on Data Engineering (ICDE'06)*, (2006).
- [20] J. Xu, L. Yao and X. Zhao, A multi-objective chance-constrained network optimal model with random fuzzy coefficients and its application to logistics distribution center location problem, *Fuzzy Optim. Decis. Ma.* **10** (3) (2011), 255–285.