Decorous Lower Bounds for Minimum Linear Arrangement

Alberto Caprara
DEIS, Università di Bologna, Viale Risorgimento 2, I-40136 Bologna, Italy, alberto.caprara@unibo.it
Adam N. Letchford
Department of Management Science, Lancaster University, Lancaster LA1 4YX, Great Britain, a.n.letchford@lancaster.ac.uk
Juan-José Salazar-González
DEIOC, Universidad de La Laguna, c/ Astrofísico Francisco Sánchez, s/n E-38271 La Laguna, Tenerife, Spain, jjsalaza@ull.es

Minimum Linear Arrangement is a classical basic combinatorial optimization problem from the 1960s, which turns out to be extremely challenging in practice. In particular, for most of its benchmark instances, even the *order of magnitude* of the optimal solution value is unknown, as testified by the surveys on the problem that contain tables in which the best known solution value often has one more digit than the best known lower bound value. In this paper, we propose a linear-programming based approach to compute lower bounds on the optimum. This allows us, for the first time, to show that the best known solutions are indeed not far from optimal for most of the benchmark instances.

Key words: programming, integer, algorithm, branch and bound; programming, integer, applications; networks-graphs, applications; analysis of algorithms

History: Submitted to *INFORMS J. on Comput.* September 2009; accepted February 2010.

1. Introduction

Given a graph $G = (V, E)$, with $V = \{1, \ldots, n\}$, an *arrangement* (also called a *permutation*, *labelling*, *ordering* or *layout*) is a one-to-one function $\psi : V \to V$. If we view ψ as a placing of the vertices on points $1, \ldots, n$ along the real line, the quantity $|\psi(i) - \psi(j)|$ corresponds to the Euclidean distance between vertices i and j. Several important combinatorial optimization problems, collectively known as *graph layout problems*, call for an arrangement minimizing a function of these distances (see the survey Díaz et al. 2002). Here, we are concerned with the *Minimum Linear Arrangement Problem* (MinLA for short, using a common acronym in the literature), in which the objective is to minimize the sum of the pairwise distances between all vertices joined by an edge, namely $\sum_{\{i,j\} \in E} |\psi(i) - \psi(j)|$.

1.1 Literature review and state-of-the-art

MinLA was originally proposed by Harper (1964). It was proven to be strongly $\mathcal{NP}$-hard by Garey et al. (1976), and this was later shown to hold even when G is bipartite (Garey and Johnson 1979). For general graphs, the fastest known exact algorithm (in terms of worst-case performance) is based on dynamic programming and runs in $\mathcal{O}(2^n m)$ time (Koren and

Harel 2002), where $m := |E|$. However, MinLA is known to be solvable in polynomial time on trees (Shiloach 1979), outerplanar graphs (Frederickson and Hambrush 1988) and certain Halin graphs (Easton *et al.* 1996). In fact, for some restricted classes of graphs, optimal layouts are known explicitly (Chung 1988, Harper 1964, Juvan and Mohar 1992, Mitcheson and Durbin 1986, Muradyan and Piliposjan 1980).

On the theoretical side, some recent progress has been made on the approximability of MinLA. Approximation algorithms with performance guarantee $\mathcal{O}(\log n)$ were introduced by Bornstein and Vempala (2004) and Rao and Richa (2005). Recently, an $\mathcal{O}(\sqrt{\log n} \log \log n)$ approximation algorithm was found (Charikar *et al.* 2006, Feige and Lee 2007). It has been shown that there does not exist a polynomial-time approximation scheme for the problem, unless $\mathcal{NP}$-complete problems can be solved in randomized subexponential time (Ambühlet al. 2007), and conjectured that MinLA cannot be approximated to within a constant factor in polynomial time (Devanur *et al.* 2006).

On the practical side, the problem appears to be extremely challenging. Before this work (and the subsequent work by Caprara et al. 2009, discussed in the conclusions), the best practical method to solve MinLA to proven optimality was the one based on dynamic programming mentioned above (which ruled out the possibility to solve instances with more than, say, 30 vertices). On the one hand, several heuristics were proposed for the problem (Koren and Harel 2002, Petit 2003a, Petit 2003b, Rodriguez-Tello et al. 2008, Safro et al. 2006) and tested on a well-established collection of benchmark instances (Petit 2001). On the other, it was so far impossible to certify that their solutions are close to optimal since, with the exception of three instances whose optimum is known by construction, the best lower bounds on the optimum are generally one *order of magnitude* smaller than the heuristic values.

The situation is illustrated in Petit (2003b) which, among other things, provides a clear picture of the situation concerning the practical solvability of MinLA. Moreover, this is testified in Table 1, in which we report the values of the best known heuristic solution (*best heur*), as reported in Rodriguez-Tello et al. (2008), and the best known lower bound value (*best LB*), taken from a table in Petit (2003b), along with the associated relative gap. Note that for instances *hc10*, *mesh33x33* and *bintree10* the optimal value is known by the structure of the problem (and reported in column *best heur*), and that the lower bound for *mesh33x33* by the mesh method illustrated in Petit (2003b) is equal to the optimum by definition.

1.2 Our contribution

In this paper, we focus our attention on the computation of lower bounds for the problem. Our main final contribution is given in Table 2, showing that, for the instances with less than (roughly) 5,000 edges in the benchmark, the current best solutions are very close to fairly close to the optimum, with gaps that are (roughly) between 5% and 20%. We can also find improved lower bounds for the larger instances but, since our lower bounding procedure does not converge within a reasonable time limit, we preferred not to report the results here.

Our approach is based on the solution of a suitable *Linear Program* (LP), which involves variables that represent distances between vertices in the layout and contains exponentially many constraints, handled through separation. In itself, this is a very natural idea that has already been exploited in the literature. However, the key to our approach is, on the

Table 1: Best known solution values and lower bounds for the MinLA benchmark instances (known optimal values are marked by '*').

name	n	m	best heur	best LB	% gap
gd95c	62	144	506	292	42.3
gd96a	1096	1676	95242	5155	94.6
gd96b	111	193	1416	702	50.4
gd96c	65	125	519	241	53.6
gd96d	180	228	2391	595	75.1
c1y	828	1749	62230	14101	77.3
c2y	980	2102	78757	17842	77.3
c3y	1327	2844	123145	23417	81.0
c4y	1366	2915	114936	21140	81.6
c5y	1202	2557	96850	19217	80.2
hc10	1024	5120	523776*	349525	33.3
mesh33x33	1089	2112	31680*	31680	0.0
bintree10	1023	1022	3696*	1277	65.4
randomA1	1000	4974	866968	140634	83.8
randomA2	1000	24738	6522206	4429294	32.1
randomA3	1000	49820	14194583	11463259	19.2
randomA4	1000	8177	1717176	601130	65.0
randomG4	1000	8173	140211	39972	71.5
3elt	4720	13722	357329	44785	87.5
airfoil1	4253	12289	272931	40221	85.3
crack	10240	30380	1491126	95347	93.6
whitaker3	9800	28989	1143645	144854	87.3

one hand, to limit to m the number of variables, and, on the other, to work with 'strong' constraints that arise from the projection into our variable space of natural inequalities in the space of dimension $\binom{n}{2}$ associated with the distances for all vertex pairs. This idea is implicit in Even et al. (2000), in which such an LP was introduced uniquely for theoretical purposes. In this paper, we significantly extend and explore computationally this idea. In doing so, we analyze the structure of the underlying polyhedron, deriving several classes of valid inequalities, proving that they are facet inducing, and discussing the associated separation problems.

The structure of the paper is as follows. In Section 2, we illustrate the LP that we will use, comparing it with the existing LP-based lower bounds for MinLA. In Section 3, we associate certain integer polyhedra with MinLA and derive various valid and facet-inducing inequalities. The complexity of separation for these inequalities is discussed in Section 4. In Section 5 we describe our cutting plane algorithm and the associated computational experiments. Conclusions are given in Section 6.

Table 2: Improved lower bounds found by our method for the MinLA benchmark instances.

name	our LB	% gap
gd95c	443	12.5
gd96a	77860	18.3
gd96b	1281	9.5
gd96c	402	22.5
gd96d	2021	15.5
c1y	59971	3.6
c2y	76253	3.2
c3y	113801	7.6
c4y	106942	7.0
c5y	88741	8.4
bintree10	3696	0.0

1.3 Notation

Throughout the paper, we use the following customary notation. First of all, as already mentioned, n and m denote the number of vertices and edges, respectively, in G. For a given vertex i, $\delta(i)$ denotes the set of edges incident on vertex i, and $N(i)$ the set of neighbours of i. We also use the standard binomial notation $\binom{a}{b} = \frac{a!}{b!(a-b)!}$. By subgraph G' of G we will always mean an *edge-induced* subgraph $G' = (V(G'), E(G'))$ such that $E(G') \subseteq E$ and $V(G') \subseteq V$ is the subset of vertices of G which are endpoints of at least one edge in $E(G')$. For a subgraph G' of G, we will let $\text{LA}(G')$ denote the optimal MinLA value for G'. We will sometimes consider the complete graph on n vertices, denoted by $K_n = (V, F)$, where F is the set of $\binom{n}{2}$ vertex pairs in V. In the hope of improving readability, whenever possible e will denote a vertex pair belonging to the edge set E of G whereas $\{i, j\}$ will denote a vertex pair not belonging to E. We let Ψ denote the collection of the $n!$ arrangements of G. Finally, given two vertices $i, j \in V$, we will let $\mathcal{P}_{ij}$ denote the collection of the paths (viewed as sets of edges) in G from i to j.

2. Linear Programming Lower Bounds

MinLA is naturally formulated as a *Integer LP* (ILP) as follows. For $i, j \in V$, let the binary variable x_{ij} take the value 1 if and only if vertex i is placed in position j (i.e., if and only if $\psi(i) = j$). Moreover, for $e = \{i, j\} \in E$, let $d_e \equiv d_{\{i,j\}}$ be a variable representing the distance between vertices i and j, i.e., $|\psi(i) - \psi(j)|$ (whose integrality does not need to be imposed explicitly). A straightforward ILP formulation is:

$$\min \sum_{e \in E} d_e,$$

$$\sum_{j \in V} x_{ij} = 1, \qquad i \in V,$$

$$\sum_{i \in V} x_{ij} = 1, \qquad j \in V,$$

$$d_{\{i,j\}} \geq |p-q|(x_{ip} + x_{jq} - 1), \qquad \{i,j\} \in E,\ p,q \in V,$$

$$x_{ij} \in \{0,1\}, \qquad i,j \in V.$$

A big disadvantage of this formulation, apart from the very large number of constraints, is that its LP relaxation admits the trivial solution $x_{ij} = 1/n$ for all $i,j \in V$ and $d_{ij} = 0$ for all $\{i,j\} \in E$, yielding a lower bound of zero. For this reason, the formulation appears to be of no practical use as it stands. However, for the closely-related Minimum Bandwidth Problem, suitable variants of this formulation turn out to be the basis of the best practical approaches to tackle the problem (Caprara and Salazar-González 2005). Unfortunately, these methods are heavily based on the fact that the objective function is of bottleneck type and, apparently, cannot be used in the MinLA context (note in particular that these methods carefully avoid solving an LP by a general-purpose solver).

The above formulation is the textbook linearization of the quadratic objective function

$$\min \sum_{\{i,j\} \in E} \sum_{p,q \in V} |p-q| x_{ip} x_{jq},$$

subject to the classical assignment constraints on the x_{ij} variables, i.e., that MinLA can be seen as a special case of the Quadratic Assignment Problem. Due to the generality and difficulty of the latter, and the relatively small size of the instances for which the state-of-the-art methods can find a provably optimal solution, or even only compute a good lower bound, it appears that tackling MinLA as a Quadratic Assignment Problem is not the best way to proceed (as also suggested by our preliminary computational results, based on the classical linearization-based lower bounds).

2.1 A sparse LP relaxation

A natural idea to get lower bounds is to stick to the distance variables d_e and to introduce new constraints that these variables must satisfy. In fact, if one does not insist on having a formulation for MinLA, it is natural to get rid of the variables x_{ij}, whose only aim is to specify the position of the vertices, significantly reducing the number of variables if the graph is sparse (as is the case for the benchmark instances).

Natural conditions to impose on the distance variables d_e are the following rank inequalities. Given a subgraph G' of G, the *rank inequality* associated with G' imposes that the sum of the distances associated with the edges in G' must be at least $\text{LA}(G')$ (several examples are given in the sequel). Considering a suitable collection $\mathcal{G}$ of subgraphs of G, the

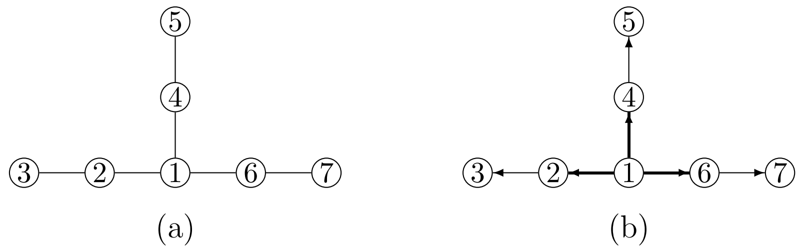

Figure 1: The simple graph G in Example 1 and an associated projected inequality.

corresponding LP relaxation is:

$$\begin{aligned} &\min \sum_{e \in E} d_e, \\ &\sum_{e \in E(G')} d_e \geq \mathrm{LA}(G'), \qquad G' \in \mathcal{G}. \end{aligned} \tag{1}$$

(Note that the trivial lower bounds $d_e \geq 1$ for $e \in E$ are implicit in (1) if all subgraphs induced by a single edge are in $\mathcal{G}$.) This LP relaxation was proposed in Liu and Vannelli (1995). Of course, the strength and the solvability of the relaxation depend on $\mathcal{G}$. For instance, if $G \in \mathcal{G}$, the right-hand-side of the associated rank inequality is $\mathrm{LA}(G)$, thus solving the relaxation is equivalent to solving MinLA on G. On the other hand, if $\mathcal{G}$ is the collection of all subgraphs induced by a single edge, the relaxation has the trivial solution $d_e = 1$ for $e \in E$, yielding the useless lower bound m on $\mathrm{LA}(G)$. In order to get useful lower bounds, it is natural to consider in $\mathcal{G}$ subgraphs G' for which not only computation of $\mathrm{LA}(G')$ is easy, but also the separation of the constraints can be handled in practice (possibly via heuristics). A classical and relevant example is the one in which $\mathcal{G}$ contains all the (edge-induced) *stars* of G, noting that for a vertex $i \in V$ there are $2^{|N(i)|} - 1$ edge-induced stars, one for each nonempty subset of $N(i)$, all of which may be useful for the LP relaxation. As discussed next, the separation of star inequalities is very easy (and the value $\mathrm{LA}(G')$ for a star G' is given by a simple formula). In Liu and Vannelli (1995), a few other simple structures in addition to stars are considered, even if the separation problem for these is not addressed explicitly, so presumably separation is heuristic in the computational results.

Example 1 Consider the simple graph G with $n = 7$ and $m = 6$ illustrated in Figure 1(a), for which it is easy to check that $\mathrm{LA}(G) = 8$. If we restrict $\mathcal{G}$ to be the collection of the stars of G, the nondominated inequalities in (1) are $d_{\{1,2\}} + d_{\{1,4\}} + d_{\{1,6\}} \geq 4$ and the trivial lower bounds $d_e \geq 1$ for $e \in E$, and the corresponding optimal value of LP (1) is 7.

2.2 A dense LP relaxation

The main advantage of LP relaxation (1) is the small number m of variables. However, if one extends the set of distance variables to all $\binom{n}{2}$ vertex pairs, one may obtain a much tighter LP relaxation for the same collection of subgraph classes (e.g., if only stars are considered). Recall that $K_n = (V, F)$ is the complete graph on n vertices, and let $\mathcal{K}$ now be a suitable collection of edge induced subgraphs K' of K_n. Let $F(K')$ denote the set of edges in subgraph

K'. Introduce a binary variable $d_{\{i,j\}}$ for each edge $\{i,j\} \in F$. The LP relaxation, in which only the objective function depends on G, is:

$$\min \sum_{e \in E} d_e$$
$$\sum_{\{i,j\} \in F(K')} d_{\{i,j\}} \geq \text{LA}(K'), \qquad K' \in \mathcal{K}. \tag{2}$$

For instance, in case $\mathcal{K}$ is the collection of all stars of K_n, for each vertex $i \in V$ there are $2^{n-1} - 1$ inequalities in (2), one for each nonempty subset of $V \setminus \{i\}$.

As it is, LP relaxation (2) has no advantage with respect to (1) since, by setting $d_{\{i,j\}}$ to a sufficiently large value for $\{i,j\} \notin E$, we can satisfy all the constraints when K' is not a subgraph of G. However, we get an LP relaxation much stronger than (1) if we also impose the natural condition that the d variables define a metric, by the following *triangle inequalities*:

$$d_{\{i,j\}} \leq d_{\{i,k\}} + d_{\{k,j\}}, \qquad \{i,j,k\} \subseteq V. \tag{3}$$

An LP relaxation analogous to (2)-(3) was considered in Amaral and Letchford (2006) for a generalization of MinLA.

Example 1 (cont.) If we restrict $\mathcal{K}$ to be the collection of the stars of K_n, (2) contains, among others, the inequality

$$d_{\{1,2\}} + d_{\{1,3\}} + d_{\{1,4\}} + d_{\{1,5\}} + d_{\{1,6\}} + d_{\{1,7\}} \geq 12 \tag{4}$$

that, jointly with the triangle inequalities $d_{\{1,3\}} \leq d_{\{1,2\}} + d_{\{2,3\}}$, $d_{\{1,5\}} \leq d_{\{1,4\}} + d_{\{4,5\}}$, $d_{\{1,7\}} \leq d_{\{1,6\}} + d_{\{6,7\}}$ and the trivial lower bounds, leads to an optimal value of 7.5 for LP (2)-(3).

For LP relaxation (1), the metric condition can be imposed by the following *path inequalities*, which are exponentially many but can easily be dealt with as their separation simply requires computing shortest paths in G:

$$d_{\{i,j\}} \leq \sum_{e \in P_{ij}} d_e, \qquad \{i,j\} \in E, \; P_{ij} \in \mathcal{P}_{ij}, \tag{5}$$

recalling that $\mathcal{P}_{ij}$ denotes the collection of all paths in G from i to j.

2.3 A projected LP relaxation

We can combine the advantages of the two previous LP relaxations, namely the small number of variables of (1) and the relative tightness of (2)-(3), essentially by taking the latter and projecting it onto the variable space of the former. Roughly speaking, this amounts to replacing, in each rank inequality, a variable $d_{\{i,j\}}$ by the sum of the variables d_e in the shortest path from i to j in G (and to removing the triangle inequalities). This projected relaxation can be handled in a way analogous to LP (2), in that the separation of rank inequalities is the same modulo pre-computing the shortest paths between all vertex pairs, as explained in the following.

Formally, using the same variables as in (1), as well as the same notation as in (2), the projected LP relaxation is:

$$\min \sum_{e \in E} d_e$$
$$\sum_{\{i,j\} \in F(K')} \sum_{e \in P_{ij}} d_e \geq \text{LA}(K'), \qquad K' \in \mathcal{K},\ \{i,j\} \in F(K'),\ P_{ij} \in \mathcal{P}_{ij}, \tag{6}$$

where we now have an inequality for each subgraph $K' \in \mathcal{K}$ and for each choice of paths in G joining the endpoints of the edges in K'. Note that these inequalities are not necessarily of rank type, namely an edge e may appear in more than one path and therefore d_e may have a coefficient larger than one.

Example 1 (cont.) The projected LP relaxation (6) contains, among others, the inequality $2d_{\{1,2\}} + d_{\{2,3\}} + 2d_{\{1,4\}} + d_{\{4,5\}} + 2d_{\{1,6\}} + d_{\{6,7\}}, \geq 12$, associated with (4) and with the unique paths from 1 to all vertices, yielding again an optimal value of 7.5 for LP (6). This inequality is illustrated in Figure 1(b), in which the edges with a coefficient of 2 are drawn as thick lines (the reason for drawing the edges as arrows will be clear in Section 4).

A formulation equivalent to (6), with $\mathcal{K}$ restricted to be the collection of all stars of K_n, was given in Even et al. (2000), where the (equivalent) constraints are called *spreading constraints* and the feasible solutions *spreading metrics*. In Even et al. (2000), it is observed that, with that choice of $\mathcal{K}$, the LP can be solved in polynomial time. The main result of Even et al. (2000) is that the worst-case ratio between the MinLA optimum and the lower bound given by this LP is $\mathcal{O}(\log n \log \log n)$. In Rao and Richa (2005), it is shown that this ratio is in fact $\Theta(\log n)$.

Proposition 1 *The feasible region of LP* (6) *is the projection over* $\mathbb{R}^E$ *of the feasible region of LP* (2)-(3).

Proof. Clearly, for each solution d^* of (2)-(3), taking only the components of d^* corresponding to the edges in E yields a feasible solution of (6), since the triangle inequalities guarantee that $\sum_{e \in P_{ij}} d^*_e \geq d^*_{\{i,j\}}$ for each vertex pair $\{i,j\} \subseteq V$ and path $P_{ij} \in \mathcal{P}_{ij}$. On the other hand, for each solution d^* of (6), it is easy to check that extending this solution to vertex pairs $\{i,j\} \notin V$, by defining $d^*_{\{i,j\}}$ to be the length of the shortest path from i to j in G with respect to edge lengths d^*, yields a feasible solution of (2)-(3). □

Corollary 1 *LP* (6) *is equivalent to LP* (2)-(3).

The number of constraints of (6) is much larger than the one of (2), as is customary for projections, since exponentially many inequalities in the former correspond to each rank inequality in the latter. However, as we will discuss in detail in Section 4, the separation problem for (6) can be polynomially reduced to the separation problem for (2).

3. A Polyhedral Study

In this section, we illustrate classes of rank inequalities to be used to concretely define and solve the LP relaxations of Section 2. We do this by studying certain integer polyhedra associated with MinLA. We refer the reader to Nemhauser and Wolsey (1988) for an introduction to polyhedral theory and its application to combinatorial optimization.

A polyhedral study of the convex hull of distance vectors representing permutations, i.e., the polyhedron

$$P(K_n) := \text{conv}\left\{d \in \mathbb{Z}_+^F : \text{ there exists } \psi \in \Psi \text{ such that } d_{\{i,j\}} = |\psi(i) - \psi(j)| \text{ for all } \{i,j\} \in F\right\},$$

can be found in Amaral and Letchford (2006). (In fact, Amaral and Letchford (2006) study a more general class of polyhedra, associated with the so-called Single-Row Facility Layout Problem, which contains MinLA as a special case.) The affine hull was determined and several families of valid and facet-inducing inequalities were derived.

Our goal here is to work with only m distance variables d_e for each $e \in E$. At first sight, it seems natural to work with the projection of the polytope $P(K_n)$ onto the subspace defined by the edges in E, i.e., with the following integer polytope:

$$P(G) := \text{conv}\left\{d \in \mathbb{Z}_+^E : \text{ there exists } \psi \in \Psi \text{ such that } d_{\{i,j\}} = |\psi(i) - \psi(j)| \text{ for all } \{i,j\} \in E\right\}.$$

However, $P(G)$ has a fairly complex structure. We have found it helpful to study its *dominant*, which is the Minkowski sum of $P(G)$ and the non-negative orthant $\mathbb{R}_+^E$. That is:

$$D(G) := \left\{d \in \mathbb{R}_+^E : \text{ there exists } d' \in P(G) \text{ such that } d \geq d'\right\}.$$

Since the objective function in MinLA is non-negative, optimizing over $D(G)$ is equivalent to optimizing over $P(G)$. However, $D(G)$ is much easier to work with. Indeed, we have the following three elementary results.

Proposition 2 *$D(G)$ is a full-dimensional, unbounded polyhedron.*

Proof. Unboundedness is obvious. As to full dimensionality, consider an arbitrary point $d \in D(G)$. By setting, in turn, each component to a sufficiently large value M while leaving the other components unchanged, it is easy to check that the resulting m vectors, belonging to $D(G)$ by definition, together with d form a set of $m + 1$ affinely independent vectors. □

Recall that an inequality $\alpha^T d \geq \beta$ that is valid for $D(G)$ is *face-inducing* if there exists at least one point $d \in D(G)$ such that $\alpha^T d = \beta$.

Proposition 3 *If the inequality $\sum_{e \in E} \alpha_e d_e \geq \beta$ is valid for $D(G)$, then $\alpha_e \geq 0$ for $e \in E$. Moreover, if it is face-inducing, then $\beta \geq \sum_{e \in E} \alpha_e$.*

Proof. Nonnegativity of α follows since $D(G)$ is, by definition, of blocking type, i.e., for each $d \in D(G)$, we have $d' \in D(G)$ for every $d' \geq d$. The lower bound on β follows since all components of each $d \in D(G)$ are at least one. □

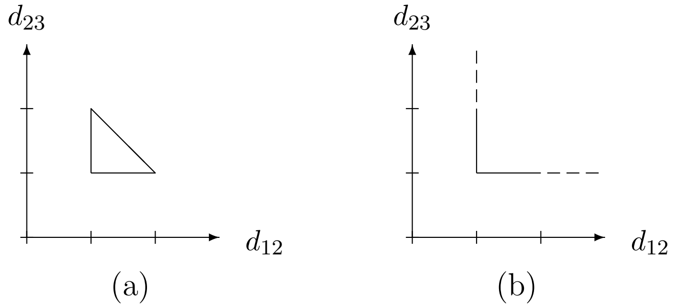

Figure 2: $P(G)$ and $D(G)$ for a trivial graph.

Proposition 4 *For each subgraph G' of G, the inequality $\sum_{e \in E(G')} \alpha_e d_e \geq \beta$ is valid (or face-inducing, or facet-inducing) for $D(G')$ if and only if it is valid (respectively, face-inducing, facet-inducing) for $D(G)$.*

Proof. Note that $D(G')$ is the projection of $D(G)$ onto the subspace defined by the edges of G'. The statement is easily checked to be valid in general for a polyhedron of blocking type, its projection onto a subspace, and an inequality having nonzero coefficients only for the variables in this subspace. □

When $G = K_n$, the relationship between $P(G)$ and $D(G)$ is clear:

Proposition 5 *$P(K_n)$ is the unique bounded facet of $D(K_n)$, induced by the equation $\sum_{\{i,j\} \in F} d_{\{i,j\}} = \binom{n+1}{3}$.*

Proof. It is shown in Amaral and Letchford (2006) that $P(K_n)$ has dimension $\binom{n}{2} - 1$ and that its affine hull is defined by the given equation. Since $D(K_n)$ is the dominant of $P(K_n)$ and has dimension $\binom{n}{2}$, $P(K_n)$ is a bounded facet of $D(K_n)$. All other facets of $D(K_n)$ must contain half-lines and therefore be unbounded. □

However, for general graphs $P(G)$ need not even be a face of $D(G)$, as the following example illustrates:

Example 2 Consider the (trivial) graph with $n = 3$ and $m = 2$, in which E contains the edges $\{1, 2\}$ and $\{2, 3\}$. The possible values taken by $(d_{\{1,2\}}, d_{\{2,3\}})$ in a feasible arrangement are $(1, 1)$, $(1, 2)$ and $(2, 1)$. Thus, $P(G)$ is a triangle in $\mathbb{R}^2_+$, as shown in Figure 2(a). $D(G)$, on the other hand, has only one extreme point, namely $(1, 1)$, as shown in Figure 2(b). □

From now on we concentrate on $D(G)$. In the following subsections, we present various valid and facet-inducing inequalities, mainly of rank type, as defined in Section 2. Given an inequality that is valid for $D(G)$, we will say that an arrangement $\psi \in \Psi$ is *tight* for the inequality if the associated distance vector satisfies it at equality. Moreover, we will sometimes use 'arrangement' to mean the associated distance vector.

3.1 Star inequalities

MinLA is trivial when G is a *star* (i.e., a graph in which all edges are incident on a common vertex). In this case, the optimal MinLA solution has cost $\lfloor n^2/4 \rfloor$. This leads to the following *star* inequalities:

Proposition 6 *For any $i \in V$ and any $S \subseteq N(i)$, the* star *inequality*

$$\sum_{j \in S} d_{\{i,j\}} \geq \lfloor (|S|+1)^2/4 \rfloor \tag{7}$$

is valid for $D(G)$, and facet-inducing if $|S| \neq 2$.

Proof. By Proposition 4, it suffices to prove the result for the case in which G is itself a star. When $|S| = 1$, the star inequality reduces to the trivial lower bound $d_{\{i,j\}} \geq 1$, which is facet-inducing since $D(G) = \{d_{\{i,j\}} \in \mathbb{R} : d_{\{i,j\}} \geq 1\}$. When $|S| = 2$, the star inequality is the sum of two trivial lower bounds and therefore not facet-inducing.

Now suppose $|S| \geq 3$. Let $\alpha^T d = \beta$ be an equation which is satisfied by all arrangements that are tight for the star inequality. Now assume that $|S|$ is even, i.e., $|S| = 2k$ for some integer k. Without loss of generality, suppose that $i = k+1$. Now let ψ be the identity arrangement (i.e., $\psi(j) = j$ for all j) and let ψ' be the arrangement obtained by switching vertices 1 and 2, noting that both ψ and ψ' are tight. A comparison of the two arrangements shows that $\alpha_{\{1,i\}} = \alpha_{\{2,i\}}$. By symmetry, this shows that $\alpha_{\{i,j\}}$ takes the same value for all $j \in S$. Therefore $\alpha^T d \geq \beta$ can be converted into the star inequality by a suitable scaling. The proof for $|S|$ odd is analogous; it suffices to suppose that $i = k$. □

Vertex i is called the *center* of the star. In Amaral and Letchford (2006) it was noted that star inequalities do not in general induce facets of $P(K_n)$.

3.2 Clique inequalities

MinLA is also trivial when $G = K_n$, since any arrangement satisfies the equation $d(E) = \binom{n+1}{3}$. This leads to the following *clique* inequalities:

Proposition 7 *For any $n \geq 2$, and for any $S \subseteq V$ inducing a clique in G, the* clique *inequality*

$$\sum_{\{i,j\} \subseteq S} d_{\{i,j\}} \geq \binom{|S|+1}{3} \tag{8}$$

is valid and facet-inducing for $D(G)$.

Proof. By Proposition 4, we can assume that $G = K_n$ and $S = V$. The result then follows from Proposition 5. □

Note that the clique inequalities with $|S| = 2$ are the trivial lower bounds $d_e \geq 1$.

3.3 Circuit inequalities

MinLA is also trivial when G is a *circuit* (i.e., a simple cycle), the optimal MinLA solution having cost $2n-2$. This leads to the following *circuit* inequalities:

Proposition 8 *For any $C \subseteq E$ inducing a circuit in G, the* circuit *inequality*

$$\sum_{e \in C} d_e \geq 2|C| - 2 \tag{9}$$

is valid and facet-inducing for $D(G)$.

Proof. By Proposition 4, it suffices to prove the result for the case in which G is itself a circuit. Without loss of generality, we assume that $E = \{\{1,2\},\ldots,\{n-1,n\}\} \cup \{1,n\}$. Let $\alpha^T d = \beta$ be an equation which is satisfied by all arrangements that are tight for the circuit inequality. Now let ψ be the identity arrangement, and let ψ' be the 'shifted' arrangement with $\psi'_i = i + 1 \bmod n$ for $i = 1,\ldots,n$, noting that both ψ and ψ' are tight. A comparison of the two arrangements shows that $\alpha_{\{1,n\}} = \alpha_{\{n-1,n\}}$. By symmetry, this shows that $\alpha_{\{i,i+1\}}$ takes the same value for all i. Therefore $\alpha^T d \geq \beta$ can be converted into the circuit inequality by a suitable scaling. □

3.4 Bipartite and double star inequalities

A class of graphs more general than stars for which there is a closed-form expression for the MinLA optimal value is that of the *complete bipartite graphs*. We denote the complete bipartite graph by $K_{p,q}$ and assume without loss of generality that $p \leq q$. In Juvan and Mohar (1992) it is shown that the optimal solution to MinLA for $K_{p,q}$ is $r(p,q) := p(3q^2 + 6pq - p^2 + 4)/12$ (when $p+q$ is even) and $r(p,q) := p(3q^2 + 6pq - p^2 + 1)/12$ (when $p+q$ is odd). This leads to the following *bipartite* inequalities:

Proposition 9 *Let $S \subseteq V$ and $T \subseteq V \setminus S$ be such that each vertex in S is adjacent to each vertex in T, with $p := |S|$, $q := |T|$, and $p \leq q$. The* bipartite *inequality*

$$\sum_{i \in S} \sum_{k \in T} d_{\{i,k\}} \geq r(p,q) \tag{10}$$

is valid for $D(G)$. It is facet-inducing if and only if one of the following three conditions holds:

- $p = 1$ *and* $q \neq 2$,
- $p = 2$ *and q is even,*
- $p \geq 3$.

Proof. When $p = 1$, the bipartite inequalities reduce to star inequalities, and are therefore facet-inducing if and only if $q \neq 2$.

Now suppose that $p \geq 3$. By Proposition 4, it suffices to prove the result for the case in which $G = K_{p,q}$. Let $\alpha^T d = \beta$ be an equation which is satisfied by all arrangements that are tight for the bipartite inequality.

In Juvan and Mohar (1992) it was shown that any arrangement of the following form is tight: the first $\lceil \frac{q-p}{2} \rceil$ and the last $\lceil \frac{q-p}{2} \rceil$ positions are occupied by vertices in T, whereas the positions in the center are occupied by vertices in S and T in alternation. Notice that, if we take such an arrangement and exchange the positions of the vertices in T in positions $\lceil \frac{q-p}{2} \rceil$ and $\lceil \frac{q-p}{2} \rceil + 2$, say k and ℓ, we obtain another tight arrangement. Letting $i \in S$ denote the vertex in position $\lceil \frac{q-p}{2} \rceil + 1$ and using symmetry, this shows that

$$\sum_{j \in S \setminus \{i\}} \alpha_{\{j,k\}} = \sum_{j \in S \setminus \{i\}} \alpha_{\{j,\ell\}}, \qquad i \in S, \ \{k, \ell\} \subseteq T.$$

Standard linear algebra calculations show that this is equivalent to

$$\alpha_{\{i,k\}} = \alpha_{\{i,\ell\}}, \qquad i \in S, \ \{k, \ell\} \subseteq T.$$

Thus, the equation must take the form:

$$\sum_{i \in S} \lambda_i \sum_{k \in T} d_{\{i,k\}} = \beta.$$

Now notice that, if we take a tight arrangement of the same form and exchange the positions of the vertices in S in positions $\lceil \frac{q-p}{2} \rceil + 1$ and $\lceil \frac{q-p}{2} \rceil + 3$, say i and j, we obtain another tight arrangement. The exchange moves vertex i further away from $\lceil \frac{q-p}{2} \rceil$ vertices in T, but closer to $q - \lceil \frac{q-p}{2} \rceil - 1$ other vertices in T. The reverse holds for vertex j. This implies that

$$(p-1)\lambda_i = (p-1)\lambda_j, \qquad \{i, j\} \subseteq S \tag{11}$$

if $q - p$ is even, and

$$(p-2)\lambda_i = (p-2)\lambda_j, \qquad \{i, j\} \subseteq S \tag{12}$$

if $q - p$ is odd. These equations imply that $\lambda_i = \lambda_j$ for all $\{i, j\} \subseteq S$. So, the inequality $\alpha^T d \geq \beta$ can be converted into the bipartite inequality by a suitable scaling.

A similar argument shows that the bipartite inequality induces a facet when $p = 2$ and q is even. When $p = 2$ and q is odd, however, the equations (12) reduce to the trivial equations $0 = 0$, and we cannot conclude that the bipartite inequality induces a facet. In fact, it does not, because every tight arrangement satisfies the equation

$$\sum_{k \in T} d_{\{i,k\}} = \sum_{k \in T} d_{\{j,k\}},$$

where $S = \{i, j\}$. □

In the remaining case, in which $p = 2$ and q is odd, the bipartite inequality can be derived by summing together two of the following *double star* inequalities, and dividing the resulting inequality by 3.

Proposition 10 *Let* $\{i,j\} \subseteq V$ *and* $T \subseteq V \setminus \{i,j\}$ *be such that* i *and* j *are adjacent to every vertex in* T*, and* $|T|$ *is odd and at least 3. The* double star *inequality*

$$\sum_{k \in T} (2d_{\{i,k\}} + d_{\{j,k\}}) \geq 3(|T|^2 + 4|T| - 1)/4 \tag{13}$$

is valid and facet-inducing for $D(G)$.

Proof. Let $t := |T|$. By Proposition 4, it suffices to prove the result for the case in which $G = K_{2,t}$.

Consider any feasible arrangement, and without loss of generality suppose that i is to the left of j (i.e., $\psi(i) < \psi(j)$). Suppose that there are n_1 vertices to the left of i and n_2 vertices to the right of j (and therefore $t - n_1 - n_2$ vertices between i and j). The left-hand side of the double star inequality, computed with respect to the given arrangement, is easily checked to be:

$$2n_1^2 + n_2^2 - 2n_1 t - n_2 t + n_1 + 2n_2 + 3t(t+1)/2.$$

A tedious but straightforward calculation shows that this is minimized when $n_1 = (t-1)/2$ and $n_2 \in \{(t-3)/2, (t-1)/2\}$, at which points it takes the value $3(t^2 + 4t - 1)/4$. Thus, the inequality is valid.

Let $\alpha^T d = \beta$ be an equation which is satisfied by all arrangements that are tight for the double star inequality. Exactly the same exchange argument used in the proof of the previous proposition shows that $\alpha_{\{i,k\}} = \alpha_{\{i,\ell\}}$ and $\alpha_{\{j,k\}} = \alpha_{\{j,\ell\}}$ for all $\{k,\ell\} \subseteq T$. Thus, the equation must take the form:

$$\sum_{k \in T} \left(\lambda d_{\{i,k\}} + \mu d_{\{j,k\}}\right) = \beta.$$

Now let k and ℓ be vertices in T. Consider a tight arrangement in which k lies between i and j, and ℓ lies to the immediate right of j. If we exchange the positions of j and ℓ, we obtain another tight arrangement. A comparison of the two arrangements shows that $\lambda = 2\mu$. Therefore the inequality $\alpha^T d \geq \beta$ can be converted into the double star inequality by a suitable scaling. □

Double star inequalities are the only inequalities not of rank type addressed in this paper.

3.5 Tree inequalities

All rank inequalities illustrated so far are associated with graphs G' for which there is a closed-form expression of $\text{LA}(G')$. For trees, such an expression is not known but MinLA can be solved efficiently (Shiloach 1979), although the algorithm is far from trivial and the structure of the optimal solution often fairly complex. Therefore, although it appears to be hopeless to prove that they are facet inducing, one may consider the following *tree* inequalities in a cutting plane approach.

Proposition 11 *For any* $F \subseteq E$ *inducing a tree* T *in* G*, the* tree *inequality*

$$\sum_{e \in F} d_e \geq \text{LA}(T) \tag{14}$$

is valid for $D(G)$ *and its right-hand-side can be computed in polynomial time.*

3.6 Projected inequalities

One limitation of the above classes of valid inequalities is that they rely on G containing subgraphs of certain pre-specified types (cliques, circuits, and so on). In the following theorem, we describe a 'projection' operation that enables one to convert any valid inequality for $D(K_n)$ into one or more valid inequalities for $D(G)$, regardless of the structure of G. This is the basic idea behind the definition of LP relaxation (6) from (2).

Proposition 12 *Let*

$$\sum_{\{i,j\} \in F} \alpha_{\{i,j\}} d_{\{i,j\}} \geq \beta, \tag{15}$$

with α and β non-negative, be a valid inequality for $D(K_n)$, and $G = (V, E)$ be any connected subgraph of K_n. Moreover, for each $\{i, j\} \in F$, let $P_{ij} \subseteq E$ be the edge set of an arbitrary path from i to j in G. Then the projected *inequality*

$$\sum_{\{i,j\} \in F} \alpha_{\{i,j\}} \sum_{e \in P_{ij}} d_e \geq \beta \tag{16}$$

is valid for $D(G)$.

Proof. If the inequality $\sum_{\{i,j\} \in F} \alpha_{\{i,j\}} d_{\{i,j\}} \geq \beta$ is valid for $D(K_n)$, it must also be valid for $P(K_n)$ (since $P(K_n)$ is a facet of $D(K_n)$). Now, for any $\{i, j\} \in F$ and any path P_{ij} from i to j, the inequality $\sum_{e \in P_{ij}} d_e \geq d_{\{i,j\}}$ is valid for $P(K_n)$, since the distances between points must obey the triangle inequality. This shows that the projected inequality is valid for $P(K_n)$. Since it has non-negative coefficients, it is also valid for $D(K_n)$. From Proposition 4, it is also valid for $D(G)$. □

The condition that G be connected in Proposition 12 is not a significant limitation, since MinLA decomposes into one or more independent subproblems when G is disconnected. We also remark that there is no guarantee that the projected version of a facet-inducing inequality will also be facet-inducing, or even only face-inducing. This will be discussed in detail for projected star inequalities, anticipating that basically all such inequalities turn out to be face-inducing in practice.

4. Separation

Since all of the classes of valid inequalities presented in Section 3 are exponential in number, we need exact or heuristic *separation algorithms*, i.e., routines for detecting when an inequality in a given class is violated by the solution to the current LP relaxation (Nemhauser and Wolsey 1988, Grötschel et al. 1988). In this section, we discuss the complexity of separation for the inequalities introduced in the previous section. Throughout, the LP solution is denoted by $d^* \in \mathbb{R}_+^E$.

4.1 Separation of non-projected inequalities

Proposition 13 *The separation problem for star inequalities (7) can be solved in $\mathcal{O}(m \log n)$ time.*

Proof. It suffices to do the following for each vertex $i \in V$. Sort the vertices $j \in N(i)$ in non-decreasing order of $d^*_{\{i,j\}}$. Iteratively insert vertices into S in the sorted order, checking the inequality for violation at each iteration. For a given vertex i, the time taken to sort is proportional to $|N(i)| \log |N(i)|$. The total time is therefore proportional to $\sum_{i \in V} |N(i)| \log |N(i)|$, which is $\mathcal{O}(m \log n)$. □

Proposition 14 *The separation problem for double star inequalities (13) can be solved in $\mathcal{O}(nm \log n)$ time.*

Proof. It suffices to do the following for each ordered pair $(i,j) \in V$. Sort the vertices $k \in N(i) \cap N(j)$ in non-decreasing order of $2d^*_{\{i,k\}} + d^*_{\{j,k\}}$. Iteratively insert vertices into T in the sorted order, checking the inequality for violation whenever $|T|$ is odd. For a given pair (i,j), the time taken to sort is proportional to $|N(i) \cap N(j)| \log |N(i) \cap N(j)|$, if $N(i) \cap N(j)$ has been precomputed and stored once for all. This implies that the overall time to check, for a given vertex i, all pairs (i,j) is $\mathcal{O}(m \log n)$. The total time is then $\mathcal{O}(nm \log n)$. □

Proposition 15 *The separation problem for clique inequalities (8) is strongly $\mathcal{NP}$-complete.*

Proof. An elementary calculation shows that an inequality (8) violated by d^* corresponds to a clique S^* in G such that $\sum_{\{i,j\} \subseteq S^*} (|S^*|+1-3d^*_{\{i,j\}}) > 0$. We reduce the well-known strongly $\mathcal{NP}$-complete problem of testing if G has a clique of size p to the problem of finding such a set S^* for a suitable d^*. For $e \in E$, let $d^*_e := (p+1-1/n)/3$. If G contains a clique S^* with p vertices, then we have $\sum_{\{i,j\} \subseteq S^*} (|S^*|+1-3d^*_{\{i,j\}}) = \binom{p}{2}(p+1-p-1+1/n) > 0$. On the other hand, for each clique S^* with no more than $p-1$ vertices, we have $\sum_{\{i,j\} \subseteq S^*} (|S^*|+1-3d^*_{\{i,j\}}) \leq \binom{|S^*|}{2}(p-p-1+1/n) < 0$. □

Proposition 16 *The separation problem for circuit inequalities (9) is strongly $\mathcal{NP}$-complete.*

Proof. An inequality (9) violated by d^* corresponds to a circuit C^* in G such that $\sum_{e \in C^*} (2 - d^*_e) > 2$. We reduce the well-known strongly $\mathcal{NP}$-complete problem of testing if G is Hamiltonian to the problem of finding such a C^* for a suitable d^*. For $e \in E$, let $d^*_e := 2(1-1/n-1/n^2)$. If G contains a Hamiltonian circuit C^*, then we have $\sum_{e \in C^*} (2-d^*_e) = 2n(1/n + 1/n^2) > 2$. On the other hand, for all circuits C^* with no more than $n-1$ edges, we have $\sum_{e \in C^*} (2 - d^*_e) \leq 2(n-1)(1/n + 1/n^2) < 2$. □

For bipartite inequalities, we are also convinced that separation is difficult, although we did not find a formal proof.

Conjecture 1 *The separation problem for bipartite inequalities (10) is strongly $\mathcal{NP}$-complete.*

For the complexity of separation of tree inequalities we do not dare to conjecture, given the very complex form of the right-hand-side.

Heuristic algorithms for the separation of inequalities (8), (9), (10), (14) are illustrated in Section 5.1.

4.2 Separation of projected inequalities

As anticipated in Section 2, in this section we discuss the separation of the projected version of the various inequalities that we consider.

Proposition 17 *Any polynomial-time algorithm for the separation of a class of valid inequalities (15) for $D(K_n)$ yields a polynomial-time algorithm for the separation of the associated projected inequalities (16).*

Proof. Given $d^* \in \mathbb{R}^E$, suppose that there is an inequality (16) violated by d^*. Letting P^*_{ij} denote the shortest path in G between i and j (with respect to lengths d^*_e), also the inequality (16) associated with these shortest paths is violated. Accordingly, we can proceed as follows: we compute the P^*_{ij} paths by an all-pair shortest-path algorithm, let $d^{**}_{\{i,j\}}$ be the length of P^*_{ij} for all vertex pairs $\{i,j\} \in F$, and run the separation algorithm for (15) on this new $d^{**} \in \mathbb{R}^F$ (note in particular that we may have $d^{**}_e < d^*_e$ for some $e \in E$, meaning that the corresponding path inequality (5) is violated). If a violated inequality is found, the associated violated inequality (16) is easily obtained by replacing each edge $\{i,j\} \in F$ by the edges in P^*_{ij}. □

In other words, given $d^* \in \mathbb{R}^E$, the separation of the projected inequalities is obtained by defining the complete graph K_n with edge lengths d^{**} equal to the shortest paths in G with respect to d^* and then running the separation algorithm over this complete graph. Given that d^{**} can be found once for all inequalities, the time complexities in the following corollaries do not take into account the associated shortest-path computation.

Corollary 2 *The separation problem for projected star inequalities can be solved in $\mathcal{O}(n^2 \log n)$ time.*

Corollary 3 *The separation problem for projected double star inequalities can be solved in $\mathcal{O}(n^3 \log n)$ time.*

Note that the converse of Proposition 17 does not hold in general, i.e., the separation of the projected version of a class of inequalities may be easier than the separation of the non-projected version, given that in the former case the edge lengths d^{**} satisfy the triangle inequalities. This is not the case, however, for the circuit inequalities.

Corollary 4 *The separation problem for projected circuit inequalities is strongly $\mathcal{NP}$-complete.*

Proof. Analogous to the proof of Proposition 16, considering the separation of circuit inequalities over the complete graph K_n, letting $d^{**}_{\{i,j\}} := 2(1 - 1/n - 1/n^2)$ if $\{i,j\} \in E$ and $d^{**}_{\{i,j\}} := 2$ otherwise, noting that d^* satisfies the triangle inequalities and that, again, there is a violated inequality if and only if E contains a Hamiltonian circuit. □

Moreover, we suspect that also for clique and bipartite inequalities separation is not easier when the projected version is considered.

Conjecture 2 *The separation problem for projected clique inequalities is strongly $\mathcal{NP}$-complete.*

Conjecture 3 *The separation problem for projected bipartite inequalities is strongly $\mathcal{NP}$-complete.*

Again, for the complexity of separation of projected tree inequalities we do not dare to conjecture.

4.3 On the strength of projected inequalities

As already mentioned, the projected version of a facet-inducing inequality is not even necessarily face-inducing, i.e., with maximal right-hand side in '$\geq$' form, as the example discussed next will show. Given that our method is widely based on the use of projected inequalities, this is an important issue to be addressed since, even if the separation of such inequalities is generally doable only for the original right-hand side, the right-hand side should be changed into maximal when they are added to the current LP.

Example 1 (cont.) The projected star inequality $2d_{\{1,2\}} + d_{\{2,3\}} + 2d_{\{1,4\}} + d_{\{4,5\}} + 2d_{\{1,6\}} + d_{\{6,7\}} \geq 12$ is not face-inducing, since the minimum of the left-hand-side over all feasible arrangements is in fact 13. With this value as right-hand-side, the inequality together with the trivial lower bounds would yield a value of 8 for the LP relaxation (6), equal to $\textsc{la}(G)$.

In this section, we discuss this issue for projected star inequalities, which are (by far) the most relevant inequalities in our method. For these inequalities, although computation of a maximal right-hand side appears to be nontrivial in general, we will derive a relatively simple sufficient condition for the original right-hand side to be maximal. Our computational experiments show that this condition is satisfied basically for all projected star inequalities separated, which implies that the original projected inequalities are fine in practice.

Consider a projected star inequality associated with $i \in V$, $S \subseteq N(i)$, and paths $P_{ij} \in \mathcal{P}_{ij}$ for $j \in S$:

$$\sum_{j \in S} \sum_{e \in P_{ij}} d_e \geq \lfloor (|S| + 1)^2/4 \rfloor. \tag{17}$$

First of all, we derive a suitable subclass of these inequalities which dominates the whole class. For each path P_{ij}, let $\vec{P}_{ij}$ denote the directed path obtained by orienting its edges from i to j, as illustrated in Figure 1(b). We denote an edge $\{h, k\}$ oriented from h to k by (h, k).

Proposition 18 *The subclass of projected star inequalities (17) such that:*

(i) for each $j \in S$, if P_{ij} contains an intermediate vertex $h \notin \{i, j\}$, we have that $h \in S$;

(ii) $\bigcup_{j \in S} \vec{P}_{ij}$ induces a directed tree T in G (rooted at i);

dominates the whole class of projected star inequalities.

Proof. Consider a projected star inequality not satisfying (i), i.e., such that there exists a vertex h visited by P_{ij} for which $h \notin S$. This inequality is dominated as it can be obtained by summing together: (a) the inequality obtained by replacing S by $S \setminus \{j\} \cup \{h\}$ and P_{ij} by its subpath from i to h and (b) the (slack) lower bounds $d_e \geq 0$ for the edges in the subpath of P_{ij} from h to j. By iterating the procedure one gets a dominating inequality satisfying (i).

Now consider a projected star inequality satisfying (i) but not (ii), i.e., such that there exists a vertex $h \in S$ that is visited by a subpath $P'_{ih} \subseteq P_{ij}$ for $j \neq h$ and $P'_{ih} \neq P_{ih}$. This inequality is dominated as it can be obtained by summing together and dividing by two: (a) the inequality obtained by replacing P'_{ih} by P_{ih} in the path from i to j and (b) the inequality obtained by replacing P_{ih} by P'_{ih} in the path from i to h. By iterating the procedure on the new inequalities until such a vertex h does not exist, one obtains a collection of inequalities satisfying (i) and (ii) that dominate the original inequality. (In particular, note that, for all the inequalities in the collection, there is no edge $\{h, k\}$ which is oriented from h to k in one path and from k to h in another one.) □

Observe that the proof of Proposition 18 yields a polynomial-time algorithm to construct, from a projected star inequality violated by the current LP solution, a violated inequality satisfying (i) and (ii). We now state the above-mentioned sufficient condition under which such an inequality is face-inducing (which, as already mentioned, turns out to be almost always satisfied in practice).

Proposition 19 *Consider a projected star inequality satisfying (i) and (ii) in Proposition 18, let $\{1, \dots, s\}$ be the neighbors of i in T and, for $j = 1, \dots, s$, T^j be the subtree of T rooted at j. If there exists a partition of $\{1, \dots, s\}$ into S^1 and S^2 such that $|\sum_{j \in S^1} |T^j| - \sum_{j \in S^2} |T^j|| \leq 1$, then the inequality is face-inducing.*

Proof. We let $w_e := |\{j \in S : \vec{P}_{ij} \ni e\}|$ for each oriented edge $e \in T$ be the number of paths containing e. Moreover, for each vertex h in T, we let $\delta^+_T(h)$ denote the set of edges leaving h in T. Note that, if (ii) is satisfied, condition (i) is equivalent to stating that, for each oriented edge $(h, k) \in T$,

$$w_{(h,k)} = 1 + \sum_{e \in \delta^+_T(k)} w_e \tag{18}$$

(in words, out of all paths visiting k, one ends in k and the other ones continue).

Note that the left-hand side of the inequality has the form $\sum_{e \in T} w_e d_e$, i.e., it can be seen as the objective function of the weighted variant of MinLA over a tree. We show that there exists a layout of the vertices in T in which the value of this objective function is $\lfloor (|T|+1)^2/4 \rfloor$, which implies that the right-hand side is as large as possible. In fact, we show that this is the cost of any layout such that (a) the vertices in $\bigcup_{j \in S^1} T^j$ (resp., $\bigcup_{j \in S^2} T^j$) appear to the left (resp., right) of i and (b) for $j \in S^1$ (resp., $j \in S^2$), for each oriented edge $(h, k) \in T^j$, h is to the right (resp., left) of k.

Indeed, consider a layout satisfying (a) and (b). By keeping this layout and the set of vertices unchanged, we transform T into a sequence of different trees for which the layout cost is the same, ending with a star having i as a center and all edge weights equal to 1, for which the cost of the layout is indeed the required value since, by (a), $\sum_{j \in S^1} |T^j|$ vertices are to the left of i and $\sum_{j \in S^2} |T^j|$ vertices are to the right of i (i.e., the same number of vertices, modulo one, are to the left and the right of the star center, meaning that the star layout is optimal).

If $w_e = 1$ for $e \in T$, T is already a star and there is nothing to be done. Otherwise, consider an oriented edge $(h, k) \in T$ such that $w_{(h,k)} > 1$ (possibly with $h = i$), along with another oriented edge $(k, \ell) \in T$, noting that such an edge exists by (18) (possibly with $w_{(k,\ell)} = 1$). Recalling (b), we focus on the case in which h, k, ℓ appear in this order in the layout from left to right (i.e., $\{h, k, \ell\} \subseteq \bigcup_{j \in S^2} T^j \cup \{i\}$), the opposite case being identical. We remove from T edge (k, ℓ), add to T edge (h, ℓ) with $w_{(h,\ell)} := w_{(k,\ell)}$, and redefine $w_{(h,k)} := w_{(h,k)} - w_{(k,\ell)}$. It is easy to check that the new weights still satisfy (18), and that the cost of the layout is the same for the new tree (and weights). By iterating the procedure, we end up with a tree T such that $w_e = 1$ for $e \in T$. □

5. Computational Experiments

Our algorithm was implemented in C and run on a PC with processor Intel Core 2 Duo 3.33GHz and 2 GB RAM under Microsoft Windows XP Professional Version 2002 SP2, using CPLEX 11.2 as LP solver. As the results in this section show, our approach may be quite time consuming, so we imposed a time limit of 1 day (86400 seconds) CPU time for each instance.

5.1 Separation heuristics

Triangle, path, (projected) star and double star inequalities are separated efficiently in our code by the methods illustrated in Section 4. In this section, we illustrate the heuristic procedures used to separate the remaining inequalities for the complete graph K_n. According to Proposition 17, these procedures can be used also for their projected versions. The general impression is that it is fairly easy to find violated inequalities whatever the heuristic used.

For clique inequalities (8), we use a simple greedy heuristic that defines the clique S by starting with the vertex i such that $\sum_{j \in V \setminus \{i\}} d^*_{\{i,j\}}$ is minimum and then, at each iteration, adding the vertex j such that a suitable weighted combination of $\sum_{i \in S} d^*_{\{i,j\}}$ and $\sum_{i \in V \setminus (S \cup \{j\})} d^*_{\{i,j\}}$ is minimum. For each set S considered in the various iterations (and for its complement $V \setminus S$), we test violation of the corresponding clique inequality.

For bipartite inequalities (10), note that, once the set of p vertices in one side of the bipartition is fixed, the separation is easy, by using the same observations as in the separation of (double) star inequalities. Accordingly, we first enumerate all sets for $p = 2$ (and q even) and check the associated bipartite inequalities. Then, letting S be the initially empty set of vertices on one side of the bipartition, we perform a sequence of iterations adding to S the vertex such that $\sum_{i \in V \setminus (S \cup \{j\})} d^*_{\{i,j\}}$ is minimized, and, if $p = |S| > 2$, we check the associated bipartite inequalities.

We use two simple heuristic procedures for the separation of tree inequalities (14). In the first one, for each vertex $i \in V$, we consider each partial (shortest path) tree T in one of the iterations of Dijkstra's algorithm with source i and edge weights d^*, and check the tree inequality for T (the right-hand-side $\text{LA}(T)$ is computed by our implementation of the algorithm in Shiloach (1979)). The second procedure is analogous, considering each partial (shortest spanning) tree T in one of the iterations of Prim's algorithm with source i and edge weights d^*.

Finally, for circuit inequalities (9), we find a maximum-weight collection of vertex-disjoint circuits in K_n (some of the vertices may not be contained in any circuit) by letting the weight of each edge e be equal to $2 - d_e^*$ (recall the proof of Proposition 16). Such a collection can be found efficiently by matching techniques. Then, we check the inequality associated with each circuit in the collection.

5.2 Implementation details

We report the results for the LP relaxations in Section 2. In our cutting plane approach, we initialized these relaxations with the n star inequalities (7) having, for $i \in V$, vertex i as center and $S = N(i)$. When the dense LP relaxation (2) is solved, also the clique inequality (8) with $S = V$ (given that V itself is a clique of K_n) is added.

We solve all the LPs in the cutting plane process by dual simplex. After the solution of each LP relaxation, we separate inequalities in the following order.

For the dense LP relaxation (2), the first inequalities that are separated (by complete enumeration) are the triangle inequalities (3). For the sparse LP relaxation (1) and the projected LP relaxation (6), we may or may not separate the path inequalities (5), since, according to the discussion in Section 2 these inequalities do not improve the final lower bound if the time limit is not reached, but their explicit addition may speed-up the solution process. We will indicate in the results if (5) were separated.

After having separated (3) in the dense case and, possibly, (5) in the projected case, we separate, in this order:

- star inequalities, by the exact algorithm in Proposition 13;
- clique inequalities, by the heuristic in Section 5.1;
- circuit inequalities, by the heuristic in Section 5.1;
- tree inequalities, by the two heuristics in Section 5.1;
- double star inequalities, by the exact algorithm in Proposition 14;
- bipartite inequalities, by the heuristic in Section 5.1.

As the results below will show, the star inequalities, whose separation is very easy, are by far the most important ones in our approach. Moreover, for the projected LP relaxation, we often separate several projected star inequalities (17) associated with the same star inequality (7) of K_n. This is due to the fact that the shortest path (according to distances d^*) from the center i of the star to the vertices in S changes from iteration to iteration. In order to limit this phenomenon, if there were violated projected star inequalities with center $i \in V$ for a given number of consecutive cutting plane iterations, before adding a further projected star inequality (17) associated with i and $S \subseteq N(i)$, we consider all edges $e = \{i, j\}$ for $j \in S$, and: (a) if d_e is not yet a variable of the current LP (this is possible only if $\{i, j\} \notin E$), we add it; (b) if not present, we add the path inequality (5) associated with edge $\{i, j\}$ and with the current shortest path P_{ij} from i to j. (Given that we may add path inequalities associated with edges not in E, this is important also in case we separate explicitly (5) for the edges in E.)

5.3 Benchmark instances

We first considered the well-known instances from the Petit test set (Petit 2003b), available at `http://www.lsi.upc.edu/~jpetit/MinLA/Experiments/`. On the largest instances of this test set, the time limit was exceeded even when trying to solve the sparse LP relaxation (1) with $\mathcal{G}$ equal to the set of stars of G, which is essentially the simplest and fastest-to-compute lower bound discussed in this paper. Accordingly, we restricted our attention to the instances for which this lower bound could be computed. In fact, these are the instances with fewer than 3,000 edges (the next largest instance, *randomA1*, has almost 5,000 edges).

We also considered the bandwidth instances addressed in Caprara and Salazar-González (2005), available at `http://www.informs.org/site/IJOC/article.php?id=42`. Out of the associated connected graphs, we removed *bcsstk02*, as it is complete. Moreover, since all the results are identical for the six *e05r∗* instances and the two *rdb∗ instances* (although the graphs are not, but they may be isomorphic), we give the results only for *e05r0000* and and *rdb200*.

5.4 Comparison of the LP relaxations in Section 2

Table 3: Lower bounds and running times for the LP relaxations in Section 2 with star inequalities only (time limit of 86400 seconds).

name	LP (1)			LP (2)-(3)			LP (6)			LP (6)-(5)		
	LB	% gap	time	LB	% gap	time	LB	% gap	time	LB	% gap	time
gd95c	312	38.3	0.3	424	16.2	1889.6	424	16.2	0.3	424	16.2	0.3
gd96a	6708	93.0	2237.8	8755	90.8	limit	70674	25.8	limit	77860	18.3	limit
gd96b	1199	15.3	1.5	1199	15.3	limit	1261	10.9	5.2	1261	10.9	9.3
gd96c	193	62.8	0.1	376	27.6	2822.6	376	27.6	1.2	376	27.6	0.5
gd96d	920	61.5	4.6	912	61.9	limit	1966	17.8	28.3	1966	17.8	27.0
c1y	25431	59.1	293.7	27093	56.5	limit	59948	3.7	4511.9	59948	3.7	10259.8
c2y	29604	62.4	465.4	31295	60.3	limit	76229	3.2	5617.3	76229	3.2	8684.8
c3y	37184	69.8	961.6	38529	68.7	limit	113801	7.6	limit	113739	7.6	limit
c4y	28137	75.5	869.2	29528	74.3	limit	106942	7.0	limit	106627	7.2	limit
c5y	29773	69.3	687.7	31237	67.7	limit	88741	8.4	limit	86755	10.4	limit
mesh33x33	3136	90.1	662.2	5394	83.0	limit	19691	37.8	limit	20042	36.7	limit
bintree10	1362	63.1	909.0	2098	43.2	limit	2847	23.0	47.1	2847	23.0	45.9

In Table 3 we compare the lower bounds found by the LP relaxations illustrated in Section 2, using star inequalities only. For the projected LP relaxation (6) we report the results with and without the explicit separation of the path inequalities (5).

The table shows that the sparse LP relaxation (1) can be solved within relatively short time but the lower bounds produced are fairly poor. The situation does not change substantially if, in addition to star inequalities, also other inequalities are added to this relaxation. In other words, considering inequalities associated with subgraphs of the complete graph (rather than the specific graph considered) appears to be essential to find good lower bounds.

Moreover, the table shows that the dense LP relaxation (2)-(3), which would produce the same lower bounds as (6) if solved to optimality, is much (a few orders of magnitude) slower to be solved, exceeding the time limit in all but two instances. For the cases in which (2)-(3) reaches the time limit, with the exception of instance *gd96b* the lower bound it finds is much worse than the one found by (6).

Although it often reaches the time limit, the lower bounds produced by (6) are very good for instances *c∗y*, for which the previous relative gap between best known heuristic and lower bound values was roughly 80%, and fairly good for instances *gd∗*, for which the previous relative gap was roughly 50-75%, with the exception of *gd96a*, for which it was roughly 95%. The lower bound for *bintree10* is not bad, whereas for *mesh33x33* the solution may be far from convergence when the time limit is reached, although these two instances are of limited interest since their optimal value is known.

All in all, the table clearly shows that (6) widely outperforms the other two. As to the explicit separation of (5), sometimes it pays off, notably for instance *gd96a*, and sometimes it does not. According to the present discussion, in the remainder of this section we will not consider LPs (1) and (2)-(3) any more, and report the results found by (6) both without and with the explicit separation of (5).

5.5 Results with the addition of the inequalities in Section 3

Table 4: Lower bounds and running times for LP relaxation (6) with all inequalities in Section 3 (time limit of 86400 seconds).

name	LP (6)			LP (6)-(5)		
	LB	% gap	time	LB	% gap	time
gd95c	443	12.5	113.6	443	12.5	68.3
gd96a	70674	25.8	limit	77860	18.3	limit
gd96b	1281	9.5	493.5	1281	9.5	889.8
gd96c	402	22.5	218.1	402	22.5	390.3
gd96d	2021	15.5	1669.0	2021	15.5	1642.2
c1y	59970	3.6	limit	59971	3.6	limit
c2y	76251	3.2	limit	76253	3.2	limit
c3y	113801	7.6	limit	113739	7.6	limit
c4y	106942	7.0	limit	106627	7.2	limit
c5y	88741	8.4	limit	86755	10.4	limit
mesh33x33	19691	37.8	limit	20042	36.7	limit
bintree10	3696	0.0	limit	3696	0.0	limit

Table 4 presents the results of LP relaxation (6) when, besides star inequalities, all the other inequalities in Section 3 are separated by the heuristic procedures outlined above. For instance *bintree10* the separation of tree inequalities leads to the optimal value (which is not surprising), and then the method keeps on finding violated inequalities until the time limit. For the four small *gd** instances, the lower bound improvement is notable, although not impressive, but the running time increases by a couple of orders of magnitude. Note that our separation heuristics find plenty of violated inequalities, and our impression is that even with an exact separation the lower bounds would be similar. For instances *c1y* and *c2y*, although about 90% of the time limit is available when the solution of the LP with star inequalities only is complete, the lower bound improvement from then to the time limit is negligible. Of course, there is no improvement for the instances for which the time limit was reached already by separating star inequalities only. Summarizing, the other inequalities, which are the most natural ones to consider in addition to stars and are also facet inducing, give limited improvements for these instances, and only for the instances for which the LP relaxation with stars only is solved within a tiny fraction of the time limit. This is not the case for the instances from Caprara and Salazar-González (2005), as mentioned below.

5.6 Results for the instances in Caprara and Salazar-González (2005)

In order to find a MinLA heuristic solution for the instances from Caprara and Salazar-González (2005), we applied a multistart local search procedure kindly provided us by Gerd Reinelt (2009).

In Table 5, we report the results of LP relaxation of (6) for these instances, both restricting attention to star inequalities only and separating all the inequalities in Section 3. (Also for these instances models (1) and (2)-(3) give much poorer results, which are not reported. The only exception is instance *bcsstk01*, for which model (2)-(3) finds a lower bound of value 971 within minutes.)

These results show that the quality of our lower bounds is variable for these instances, with a relative gap ranging from 0% (we could solve to proven optimality instance *dwt—66*) to about 35%. Moreover, the quality of the bounds increases significantly with the addition of other inequalities besides stars, a notable difference with respect to the classical MinLA instances discussed so far. In any case, for about a third of the instances the gap is less than 10%, representing a first significant step towards proving that the solutions found by the common heuristics are near-optimal.

6. Conclusions

The MinLA instances in the literature are a well-established benchmark for the problem. In this paper, we showed for the first time that the best known heuristic solutions in the literature are indeed near-optimal for most of these instances. We hope that our contribution will stimulate further research in this direction. This result is obtained by combining two LP relaxations that were already tested previously with limited success, namely the sparse LP relaxation (1) and the dense LP relaxation (2), into a unique projected LP relaxation (6) which combines the advantages of both.

LP relaxation (6) has the following two main limits. First of all, although we have put a lot of effort into speeding-up the computation, for many instances even only solving the relaxation with star inequalities is very time consuming, and essentially does not lead to any significant result for instances with more than 5,000 edges within one day of computation

Table 5: Results for the bandwidth benchmark graphs in Caprara and Salazar-González (2005) (time limit of 86400 seconds).

name	n	m	LP (6) (star)			LP (6) (all)			LP (6)-(5) (star)			LP (6)-(5) (all)		
			LB	%gap	time	LB	%gap	time	LB	%gap	time	LB	%gap	time
bcspwr01	39	46	84	20.8	0.0	91	14.2	0.8	84	20.8	0.0	91	14.2	0.7
bcspwr02	49	59	130	19.3	0.1	144	10.6	2.0	130	19.3	0.0	144	10.6	1.8
bcspwr03	118	179	567	16.5	1.4	588	13.4	189.5	567	16.5	0.9	588	13.4	254.6
bcspwr04	274	669	3623	23.0	789.2	3696	21.4	limit	3623	23.0	105.9	3700	21.4	limit
bcsstk01	48	176	778	31.3	2.7	971	14.2	23232.8	778	31.3	1.7	972	14.1	38481.1
bcsstk02	66	2145	35937	25.0	limit	35937	25.0	limit	35937	25.0	6908.2	35937	25.0	limit
bcsstk04	132	1758	25762	13.6	5674.6	27518	7.7	limit	25762	13.6	2434.9	27569	7.5	limit
bcsstk05	153	1135	9614	13.1	1710.6	9653	12.7	limit	9614	13.1	1742.1	9653	12.7	limit
can—24	24	68	179	14.8	0.1	203	3.3	2.8	179	14.8	0.1	203	3.3	3.4
can—61	61	248	1083	4.7	3.7	1119	1.6	1221.4	1083	4.7	1.3	1119	1.6	538.0
can—62	62	78	171	19.3	0.1	187	11.8	4.2	171	19.3	0.1	187	11.8	5.1
can—73	73	152	797	27.5	6.8	971	11.7	2016.8	797	27.5	2.6	971	11.7	4855.5
can—96	96	336	1609	40.5	154.2	2105	22.1	27786.0	1609	40.5	39.9	2105	22.1	8280.6
can–144	144	576	1752	45.7	142.2	2304	28.5	19608.4	1752	45.7	2.4	2304	28.5	1710.6
can–161	161	608	4568	31.8	1788.9	5657	15.5	limit	4568	31.8	628.9	5569	16.8	limit
can–187	187	652	3053	41.2	17521.6	3827	26.2	limit	3051	41.2	limit	3051	41.2	limit
can–229	229	774	6155	36.6	11094.8	7461	23.1	limit	6155	36.6	6707.4	7461	23.1	limit
can–256	256	1330	18211	20.9	33942.8	20627	10.4	limit	18162	21.1	limit	18162	21.1	limit
can–268	268	1407	17377	19.6	limit	17377	19.6	limit	17304	19.9	limit	17304	19.9	limit
can–292	292	1124	14144	28.2	18653.5	14981	24.0	limit	14144	28.2	4354.2	15012	23.8	limit
dwt—59	59	104	222	23.2	0.2	258	10.7	55.4	222	23.2	0.1	258	10.7	88.2
dwt—66	66	127	192	0.0	0.0	192	0.0	1.7	192	0.0	0.0	192	0.0	1.7
dwt—72	72	75	143	14.4	0.1	150	10.2	6.7	143	14.4	0.1	149	10.8	7.0
dwt—87	87	227	879	5.7	4.5	897	3.8	20761.1	879	5.7	3.8	897	3.8	18430.7
dwt–162	162	510	1832	24.6	4667.1	2032	16.4	limit	1832	24.6	3901.4	2029	16.5	limit
dwt–193	193	1650	21021	15.0	24126.8	23073	6.8	limit	21021	15.0	8653.6	23154	6.4	limit
dwt–209	209	767	5731	10.3	2001.5	5905	7.5	limit	5731	10.3	516.3	5905	7.5	limit
dwt–221	221	704	3571	5.5	2721.9	3603	4.7	limit	3571	5.5	634.6	3603	4.7	limit
dwt–245	245	608	3403	11.8	382.1	3422	11.3	limit	3403	11.8	128.2	3422	11.3	limit
bwm200	200	298	484	2.4	403.4	495	0.2	2409.1	484	2.4	234.4	495	0.2	33409.2
e05r0000	236	2847	41814	29.2	limit	41814	29.2	limit	43377	26.5	limit	43377	26.5	limit
fidap001	216	2079	32419	15.0	limit	32419	15.0	limit	32325	15.2	limit	32325	15.2	limit
fidap005	27	126	402	2.9	0.2	412	0.5	5.4	402	2.9	0.2	412	0.5	4.2
fidapm05	42	239	975	2.8	2.3	998	0.5	7492.9	975	2.8	0.9	998	0.5	805.2
lshp-265	265	744	5137	15.6	13761.3	5497	9.7	limit	5137	15.6	22256.4	5602	8.0	limit
lund-a	147	1151	9489	16.2	6679.3	10760	5.0	limit	9489	16.2	2530.1	10772	4.9	limit
lund-b	147	1147	9465	15.4	5441.1	10692	4.5	limit	9465	15.4	3156.9	10712	4.3	limit
nos4	100	247	943	8.5	9.9	976	5.3	59118.0	943	8.5	5.7	976	5.3	60497.5
pde225	225	420	2145	29.4	725.5	2539	16.5	70519.0	2145	29.4	287.4	2538	16.5	limit
rdb200	200	460	2581	31.5	1144.3	3052	19.0	limit	2581	31.5	267.6	3066	18.6	73463.6
saylr1	238	445	2313	25.6	1208.7	2673	14.0	limit	2313	25.6	257.7	2676	13.9	limit
steam1	240	1761	21017	26.3	57429.7	21559	24.4	limit	21017	26.3	limit	21017	26.3	limit
steam3	80	424	1360	4.0	30.8	1382	2.4	limit	1360	4.0	8.3	1406	0.7	limit
tub100	100	148	236	4.1	1.9	245	0.4	74.3	236	4.1	0.2	245	0.4	71.5

on a PC. Second, even with the addition of the integrality requirement on the variables, the model would not be a valid formulation of MinLA, and testing the MinLA feasibility of a given integer solution would be nontrivial (in fact, we conjecture that this is an NP-complete recognition problem).

In order to overcome the second limit above, in Caprara et al. (2009) a different formulation of the problem is proposed, with a much larger number of variables. The solution of the associated LP relaxation uses many of the ideas presented in this paper for the solution of LP (6). This approach could solve to proven optimality the two smallest instances in the MinLA benchmark, namely *gd95c* and *gd96c*, and find better lower bounds than ours for *gd96b* and *gd96d*. On the other hand, recalling the first limit above, this formulation seems to be absolutely too large for the other MinLA benchmark instances, as well as for any graph with more than a few hundred edges.

Acknowledgments

The second author was supported by the Engineering and Physical Sciences Research Council under grant EP/D072662/1. The third author was supported by the Ministerio de Ciencia e Innovación under grant MTM2009-14039-C06-01. We are grateful to André Amaral for illuminating discussions and to Gerd Reinelt for providing us with the source code of a multistart local search procedure for MinLA implemented at the University of Heidelberg.

References

Amaral, A.R.S., A.N. Letchford. 2006. A polyhedral approach to the single row facility layout problem. Working paper, Lancaster University.

Ambühl, C., M. Mastrolilli, O. Svensson. 2007. Inapproximability results for sparsest cut, optimal linear arrangement, and precedence constrained scheduling. *Proc. of the 48th Annual IEEE Symposium on Foundations of Computer Science*, 329–337.

Bornstein, C., S. Vempala. 2004. Flow metrics. *Theoretical Computer Science* **321** 13–24.

Caprara, A., J.J. Salazar-González. 2005. Laying out sparse graphs with provably minimum bandwidth. *INFORMS Journal on Computing* **17** 356–373.

Caprara, A., M. Jung, M. Oswald, G. Reinelt, E. Traversi. 2009. A betweenness approach for solving the linear arrangement problem. Working paper, DEIS, Università di Bologna.

Charikar, M., M.T. Hajiaghayi, H. Karloff, S. Rao. 2010. l_2^2 spreading metrics for vertex ordering problems. *Algorithmica* **56** 577–604.

Chung, F.R.K. 1988. Labellings of graphs. In L. Beineke, R. Wilson (eds.) *Selected Topics in Graph Theory 3*, 151–168. San Diego: Academic Press.

Devanur, N.R., S.A. Khot, R. Saket, N.K. Vishnoi. 2006. On the hardness of minimum linear arrangement. Working paper, College of Computing, Georgia Tech.

Díaz, J., J. Petit, M. Serna. 2002. A survey of graph layout problems. *ACM Computing Surveys* **34** 313–356.

Easton, T., S.B. Horton, R.G. Parker (1996) A solvable case of the optimal linear arrangement problem on Halin graphs. *Congressus Numerantium* **119** 3–17.

Even, G., J. Naor, S. Rao, B. Schieber. 2000. Divide-and-conquer approximation algorithms via spreading metrics. *Journal of the ACM* **47** 585–616.

Feige, U., J.R. Lee. 2007. An improved approximation ratio for the minimum linear arrangement problem. *Information Processing Letters* **101** 26–29.

Frederickson, G.N., S.E. Hambrush. 1988. Planar linear arrangements of outerplanar graphs. *IEEE Transactions on Circuits and Systems* **35** 323–332.

Garey, M.R., D.S. Johnson. 1979. *Computers and Intractability: An Introduction to the Theory of $\mathcal{NP}$-Completeness*. New York: Freeman.

Garey, M.R., D.S. Johnson, L.J. Stockmeyer. 1976. Some simplified $\mathcal{NP}$-complete graph problems. *Theoretical Computer Science* **1** 237–267.

Grötschel, M., L. Lovász, A.J. Schrijver. 1988. *Geometric Algorithms and Combinatorial Optimization*. New York: Wiley.

Harper, L.H. 1964. Optimal assignment of numbers to vertices. *SIAM Journal* **12** 131–135.

Juvan, M., B. Mohar. 1992. Optimal linear labellings and eigenvalues of graphs. *Discrete Applied Mathematics* **36** 153–168.

Koren, Y., D. Harel. 2002. A multi-scale algorithm for the linear arrangement problem. In L. Kucera (ed.) *Proc. of 28th International Workshop on Graph-Theoretic Concepts in Computer Science*, 293–306. Lecture Notes in Computer Science **2573**. Berlin: Springer.

Liu, W., A. Vannelli. 1995. Generating lower bounds for the linear arrangement problem. *Discrete Applied Mathematics* **59** 137–151.

Mitcheson, G., R. Durbin. 1986. Optimal numberings of an $n \times n$ array. *SIAM Journal on Discrete Mathematics* **7** 571–582.

Nemhauser, G.L., L.A. Wolsey. 1988. *Integer and Combinatorial Optimization*. Chichester: Wiley.

Muradyan, D.O., T.E. Piliposjan. 1980. Minimal numberings of vertices of a rectangular lattice. *Akademii Nauk Armianskoi SSR* **70** 21–27 (in Russian).

Petit, J. 2001. *Layout Problems*. Doctoral thesis, Department of Languages and Information Systems, Universitat Politècnica de Catalunya .

Petit, J. 2003a. Combining spectral sequencing and parallel simulated annealing for the MinLA problem. *Parallel Processing Letters* **13** 77–91.

Petit, J. 2003b. Experiments on the linear arrangement problem. *Journal of Experimental Algorithmics* **8** article 2.3.

Rao, S., A.W. Richa. 2005. New approximation techniques for some linear ordering problems. *SIAM Journal on Computing* **34** 388–404.

Reinelt, G. 2009. Personal communication.

Rodriguez-Tello, E., J.-K. Hao, J. Torres-Jimenez. 2008. An effective two-stage simulated annealing algorithm for the minimum linear arrangement problem. *Computers and Operations Research* **35** 3331–3346.

Safro, I., D. Ron, A. Brandt. 2006. Graph minimum linear arrangement by multilevel weighted edge contractions. *Journal of Algorithms* **60** 24–41.

Shiloach, Y. 1979. A minimum linear arrangement algorithm for undirected trees. *SIAM Journal on Computing* **8** 15–32.