Open access • Journal Article • DOI:10.1007/S10107-012-0533-Z

A polyhedral approach to the single row facility layout problem — Source link

André Renato Sales Amaral, Adam N. Letchford

Institutions: Universidade Federal do Espírito Santo, Lancaster University

Published on: 01 Oct 2013 - Mathematical Programming (Springer Berlin Heidelberg)

Topics: Heuristic

A Polyhedral Approach to the Single Row Facility Layout Problem

André R.S. Amaral* Adam N. Letchford†

January 2011

Abstract

The *Single Row Facility Layout Problem* (SRFLP) is the NP-hard problem of arranging facilities on a line, while minimizing a weighted sum of the distances between facility pairs. In this paper, a detailed polyhedral study of the SRFLP is performed, and several huge classes of valid and facet-inducing inequalities are derived. Some separation heuristics are presented, along with a primal heuristic based on multi-dimensional scaling. Finally, a branch-and-cut algorithm is described and some encouraging computational results are given.

Keywords: facility layout – polyhedral combinatorics – branch-and-cut

1 Introduction

Suppose n facilities are to be arranged on a straight line. Each facility $i \in N = \{1, \ldots, n\}$ has a positive integer length ℓ_i. For each $\{i, j\} \subset N$, c_{ij} denotes the traffic intensity between facilities i and j. The *Single-Row Facility Layout Problem* (SRFLP) asks for a *layout* of the facilities, i.e., a permutation π of the set N, that minimizes the weighted sum of the distances between all facility pairs, i.e., the quantity:

$$\min_{\pi \in \Pi} \sum_{\{i,j\} \subset N} c_{ij} d^{\pi}_{ij}, \tag{1}$$

where Π denotes the set of all layouts and d^{π}_{ij} denotes the distance between the centroids of facilities i and j in the layout π.

Suppose, for example, that $n = 3$, $(\ell_1, \ell_2, \ell_3) = (3, 5, 6)$ and $(c_{12}, c_{13}, c_{23}) = (4, 8, 9)$. An optimal layout π^* is $(3, 1, 2)$. As shown in Figure 1, this corresponds to the distances $d^{\pi^*}_{12} = 4$, $d^{\pi^*}_{13} = 4.5$ and $d^{\pi^*}_{23} = 8.5$. The cost of π^* is $(4 \times 4) + (8 \times 4.5) + (9 \times 8.5) = 128.5$.

*Instituto Superior Técnico, Universidade Técnica de Lisboa, Av. Rovisco Pais, 1049-001 Lisboa, Portugal. E-mail: `andre.r.s.amaral@ist.utl.pt`

†Department of Management Science, Lancaster University, Lancaster LA1 4YX, United Kingdom. E-mail: `A.N.Letchford@lancaster.ac.uk`

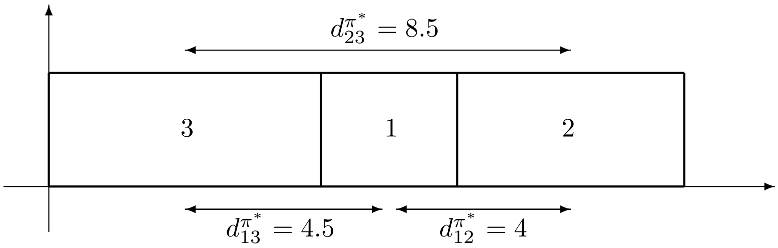

Figure 1: Optimal layout π^* for an SRFLP instance with $n = 3$.

The SRFLP has many important practical applications [15, 23, 26]. Moreover, it contains the well-known *Minimum Linear Arrangement Problem* (MinLA) as a special case, obtained when $\ell_i = 1$ for all $i \in N$ and $c_{ij} \in \{0, 1\}$ for all $\{i, j\} \subset N$. (See Díaz *et al.* [10] for a survey of MinLA and other graph layout problems.)

MinLA is NP-hard in the strong sense (Garey *et al.* [12]), and therefore so is the SRFLP. In practice, the SRFLP is similar to the well-known *Quadratic Assignment Problem* (QAP), in that instances with $n \geq 20$ can pose a serious challenge. For this reason, many authors have concentrated on heuristics; see, e.g., [11, 15, 16, 28].

To solve the SRFLP exactly, authors have suggested using branch-and-bound [26, 27], dynamic programming [17, 23], mixed-integer linear programming [1, 2, 16, 20], non-linear programming [16], semidefinite programming [4, 5, 6] and 0-1 linear programming [3]. The leading exact algorithms at present are the ones of Anjos *et al.* [5, 6], which are based on semidefinite programming (SDP), and the one of Amaral [3], which is based on linear programming (LP).

Perhaps surprisingly, no researchers have performed an in-depth polyhedral study of the SRFLP. In this paper, we perform such a study. As well as deriving valid inequalities and facets for the problem, we present some effective exact and heuristic separation algorithms, describe a branch-and-cut algorithm and present extensive computational results.

The structure of the paper is as follows. In Section 2, we define a family of polytopes associated with the SRFLP and establish some fundamental properties of them. It turns out that there is a connection between our polytopes and the well-known *cut polytope*, which has been studied in depth (see Deza & Laurent [9]). In Section 3, we derive five huge (exponentially large) families of valid inequalities, and provide conditions for them to induce facets. In Section 4, we describe a branch-and-cut algorithm for the SRFLP, that uses all of the inequalities presented in Section 3. In Section 5, we present extensive computational results. The results demonstrate that our branch-and-cut algorithm can solve instances with $n \leq 30$ in reasonable computing times, and provides tight lower and upper bounds at the root node. Finally, concluding remarks are made in Section 6.

Throughout the paper, $\binom{a}{b}$ denotes the usual binomial term $\frac{a!}{b!(a-b)!}$ and, for any $S \subseteq N$, $\ell(S)$ denotes $\sum_{i \in S} \ell_i$. Moreover, at several points in the paper we will find the following polynomial identities useful. They can be easily proved by either binomial expansion or induction.

Proposition 1 *For any subset $S \subseteq N$ we have:*

$$\sum_{\{i,j\} \subset S} \ell_i \ell_j = \left(\ell(S)^2 - \sum_{i \in S} \ell_i^2 \right) / 2 \tag{2}$$

$$\sum_{\{i,j,k\} \subset S} \ell_i \ell_j \ell_k = \left(\ell(S)^3 + 2 \sum_{i \in S} \ell_i^3 - 3\ell(S) \sum_{i \in S} \ell_i^2 \right) / 6. \tag{3}$$

2 Distance Polytopes: Fundamentals

2.1 Definition

For any given positive integer n and vector $\ell \in \mathbb{Z}_+^n$, we define the *distance polytope* $P(n, \ell)$ as the convex hull of valid distance vectors. That is:

$$P(n, \ell) := \mathrm{conv} \left\{ d \in \mathbb{R}_+^{\binom{n}{2}} : \exists \pi \in \Pi : \ d_{ij} = d_{ij}^\pi \ \forall \, 1 \leq i < j \leq n \right\}.$$

Note that $P(n, \ell)$ is the convex hull of $n!/2$ points, since each distance vector corresponds to two layouts (due to symmetry). Thus, $P(n, \ell)$ is a bounded polyhedron, i.e., a polytope. It is however not an *integral* polytope, since the distances d_{ij} need not be integral (as Figure 1 illustrates).

2.2 Dimension

Next, we show that $P(n, \ell)$ is not full-dimensional:

Lemma 1 *All layouts satisfy the equation*

$$\sum_{\{i,j\} \subset N} \ell_i \ell_j d_{ij} = \frac{1}{6} \left(\ell(N)^3 - \sum_{i \in N} \ell_i^3 \right). \tag{4}$$

Proof. First, we show that it is satisfied by the identity layout $\pi = (1, \ldots, n)$. To see this, note that $d_{ij}^\pi = (\ell_i + \ell_j)/2 + \sum_{k=i+1}^{j-1} \ell_k$ for all

$\{i,j\} \subset N$, and therefore:

$$\begin{aligned}
\sum_{\{i,j\}\subset N} \ell_i \ell_j d^{\pi}_{ij} &= \sum_{\{i,j\}\subset N} \ell_i \ell_j \left((\ell_i + \ell_j)/2 + \sum_{k=i+1}^{j-1} \ell_k \right) \\
&= \frac{1}{2} \sum_{\{i,j\}\subset N} \ell_i \ell_j (\ell_i + \ell_j) + \sum_{\{i,j,k\}\subset N} \ell_i \ell_j \ell_k \\
&= \frac{1}{6} \left(\ell(N)^3 - \sum_{i\in N} \ell_i^3 \right).
\end{aligned}$$

where the last equation follows from the identity (3). This value is clearly invariant with respect to permutation. □

The following theorem states that the equation (4) is the only one needed:

Theorem 1 *$P(n,\ell)$ is of dimension $\binom{n}{2} - 1$, and its affine hull is described by the implicit equation (4).*

Proof. To show that (4) is the only implicit equation (up to scaling by a constant), we use a standard 'indirect' proof. That is, we show that any implicit equation $\alpha^T d = \beta$ is equivalent to (4). For any two facilities i and j, let π be any layout such that i and j are in the first two positions, and let π' be the layout obtained from π by exchanging facilities i and j. A comparison of the two layouts shows that

$$\ell_i \sum_{k\in N\setminus\{i,j\}} \alpha_{jk} = \ell_j \sum_{k\in N\setminus\{i,j\}} \alpha_{ik} \qquad (\forall \{i,j\} \subset N). \tag{5}$$

Similarly, for any three facilities i, j and k, let π be any layout in which the first three positions are occupied by facilities k, i and j, respectively, and let π' be the layout obtained from π by exchanging facilities i and j. A comparison of the two layouts shows that

$$\ell_j \alpha_{ik} + \ell_i \sum_{p\in N\setminus\{i,j,k\}} \alpha_{jp} = \ell_i \alpha_{jk} + \ell_j \sum_{p\in N\setminus\{i,j,k\}} \alpha_{ip} \qquad (\forall \{i,j,k\} \subset N).$$

Together with (5), this implies

$$\ell_j \alpha_{ik} = \ell_i \alpha_{jk} \qquad (\forall \{i,j,k\} \subset N).$$

The ratios between all pairs of left hand side coefficients in the equation $\alpha^T d = \beta$ are now fixed. The equation $\alpha^T d = \beta$ can therefore be converted into (4) by a suitable scaling. □

2.3 Clique inequalities

The following lemma introduces a fundamental class of valid inequalities:

Lemma 2 *For all $S \subset N$ such that $2 \leq |S| < n$, the following 'clique' inequality is valid for $P(n, \ell)$:*

$$\sum_{\{i,j\} \subset S} \ell_i \ell_j d_{ij} \geq \frac{1}{6} \left(\ell(N)^3 - \sum_{i \in N} \ell_i^3 \right). \tag{6}$$

Proof. From the Lemma 1, the inequality (6) is satisfied at equality if the facilities in S appear consecutively in the layout. If they do not appear consecutively, the left-hand side of (6) will exceed the right-hand side, since inserting extra facilities between the existing ones can only increase the left-hand side. □

We remark that, when $|S| = 2$, the clique inequality takes the form $\ell_i \ell_j d_{ij} \geq \ell_i \ell_j (\ell_i + \ell_j)/2$, which is equivalent to the lower bound $d_{ij} \geq (\ell_i + \ell_j)/2$.

2.4 A connection with the cut cone

Next, we show a connection between $P(n, \ell)$ and a well-known polyhedron in combinatorial optimisation: the so-called *cut cone* (see Deza & Laurent [9]).

A vector $\bar{d} \in \{0,1\}^{\binom{n}{2}}$ is called a *cut vector* if there is a set $S \subset N$ such that $\bar{d}_{ij} = 1$ if and only if $i \in S$ and $j \notin S$. The cut cone of order n, which we shall denote by CC_n, is the polyhedral cone in $\mathbb{R}^{\binom{n}{2}}$ consisting of all non-negative linear combinations of cut vectors.

Proposition 2 *$P(n, \ell)$ is contained in CC_n.*

Proof. Let $\pi \in \Pi$ be a layout and let d^π be the corresponding distance vector. We will show that $d^\pi \in \mathrm{CC}_n$. By symmetry, it suffices to prove the result for the identity layout. For all $1 \leq i < j \leq n$, we have:

$$\begin{aligned} d_{ij}^\pi \quad &= (\ell_i + \ell_j)/2 + \sum_{k=i+1}^{j-1} \ell_k \\ &= \sum_{k=i}^{j-1} \frac{(\ell_k + l_{k+1})}{2}. \end{aligned}$$

Now let $\bar{d}(k)$ for $k = 1, \ldots, n-1$ be the cut vector obtained by setting $S = \{1, \ldots, k\}$. (That is, $\bar{d}(k)_{ij} = 1$ if and only if $i \leq k < j$.) Then

$$d^\pi = \sum_{k=1}^{n-1} \frac{(\ell_k + l_{k+1})}{2} \bar{d}(k),$$

showing that d^π is a non-negative linear combination of cut vectors. □

This proposition has the following useful corollary:

Corollary 1 *If the inequality $\alpha^T d \leq 0$ is valid for CC_n, then it is valid for $P(n, \ell)$.*

We will use this result in Subsections 3.2 and 3.3.

Note that the proof of Proposition 2 actually tells us a little more: if d is an extreme point of $P(n, \ell)$, then it is a non-negative linear combination of precisely $n - 1$ distinct cut vectors. We will exploit this fact in Subsection 3.4.

2.5 Zero-lifting

Next, we define an operation that we call *zero-lifting*:

Definition 1 *Let $n' > n \geq 2$, $\ell \in \mathbb{Z}_+^n$ and $\ell' \in \mathbb{Z}_+^{n'}$ be given, and define $N' = \{1, \ldots, n'\}$. Suppose that the inequality $\alpha^T d \geq \beta$ is valid for $P(n, \ell)$. Moreover, suppose that there exists a set $S = \{s(1), \ldots, s(n)\} \subset N'$ such that $\ell'_{s(i)} = \ell_i$ for all $i \in N$. Then the inequality*

$$\sum_{\{s(i), s(j)\} \subset S} \alpha_{ij} d_{s(i), s(j)} \geq \beta \tag{7}$$

is said to be obtained from the inequality $\alpha^T d \geq \beta$ by 'zero-lifting'.

We will call a valid inequality for $P(n, \ell)$ *zero-liftable* if all inequalities obtained from it by zero-lifting are valid for all suitable polytopes $P(n', \ell')$. The following lemma gives a necessary and sufficient condition for a valid inequality to be zero-liftable:

Lemma 3 *A valid inequality $\alpha^T d \geq \beta$ is zero-liftable if and only if*

$$\sum_{i \in T, j \in N \setminus T} \alpha_{ij} \geq 0 \qquad (\forall T \subset N). \tag{8}$$

Proof. Assume without loss of generality that $S = N$. Suppose the condition (8) does not hold for some T. Then the left-hand side of (7) can be made less than β by choosing n' sufficiently large, putting the facilities in T in the first $|T|$ positions, and putting the facilities in $N \setminus T$ in the last $n - |T|$ positions. Thus, the inequality is not zero-liftable.

Now suppose the condition (8) holds. Since the original inequality is valid for $P(n, \ell)$, any zero-lifted inequality will be satisfied by all layouts in which the facilities in N appear consecutively. Moreover, inserting extra facilities between the facilities in N cannot decrease the slack of the zero-lifted inequality. Thus, the inequality is zero-liftable. □

Now we give a necessary condition for zero-lifting to preserve the property of being facet-inducing:

Theorem 2 *Suppose that an inequality $\alpha^T d \geq \beta$ is zero-liftable and induces a facet of $P(n, \ell)$. Suppose moreover that all zero-liftings of it induce facets of $P(n', \ell')$ for all $n' > n$ and all suitable $\ell' \in \mathbb{Z}_+^{n'}$. Then*

$$\min_{\emptyset \neq T \subset N} \sum_{i \in T, j \in N \setminus T} \alpha_{ij} = 0. \tag{9}$$

Proof. From Lemma 3, the left hand side of (9) is non-negative. If it is positive, we can subtract a suitable positive multiple of the implicit equation (4) from the inequality $\alpha^T d \geq \beta$ so that (9) holds. The resulting inequality induces the same facet of $P(n, \ell)$ as the original inequality, and is zero-liftable by Lemma 3. The zero-liftings of the original inequality are weaker than the zero-liftings of the new inequality, since they can be obtained from the zero-liftings of the new inequality by adding a positive multiple of the clique inequality on S. This contradicts the assumption that all zero-liftings of the original inequality were facet-inducing. □

We do not know if the condition given in Theorem 2 is sufficient as well as necessary.

3 Valid Inequalities and Facets

In this section, we present various valid inequalities and show that they induce facets under mild conditions.

3.1 Clique inequalities

First, we consider the clique inequalities (6):

Theorem 3 *The clique inequalities (6) induce facets of $P(n, \ell)$.*

Proof. First, suppose that $3 \leq |S| \leq n-3$. Suppose the equation $\alpha^T d = \beta$ is satisfied by all layouts in which the clique inequality holds at equality. The exchange argument used to prove Theorem 1 shows that:

$$\begin{aligned} \ell_j \alpha_{ik} &= \ell_i \alpha_{jk} \quad (\forall \{i, j, k\} \subset S) \\ l_p \alpha_{qr} &= l_q \alpha_{pr} \quad (\forall \{p, q, r\} \subset N \setminus S). \end{aligned}$$

Now let π be any layout such that the facilities in S occupy the first $|S|$ positions. By exchanging the positions of pairs of adjacent facilities in S, we have:

$$\ell_i \sum_{q \in N \setminus S} \alpha_{jq} = \ell_j \sum_{q \in N \setminus S} \alpha_{iq} \qquad (\forall \{i, j\} \subset S). \tag{10}$$

By exchanging pairs of facilities in $N \setminus S$ instead, we have:

$$l_q \sum_{i \in S} \alpha_{ip} = l_p \sum_{i \in S} \alpha_{iq} \qquad (\forall \{p, q\} \subset N \setminus S). \tag{11}$$

Next, for any $p \in N \setminus S$, let π be any layout such that p occupies the first position and the facilities in S occupy the next $|S|$ positions. By exchanging the positions of pairs of adjacent facilities in S, we have:

$$\ell_j \alpha_{ip} + \ell_i \sum_{q \in N \setminus (S \cup \{p\})} \alpha_{jq} = \ell_i \alpha_{jp} + \ell_j \sum_{q \in N \setminus (S \cup \{p\})} \alpha_{iq} \qquad (\forall \{i,j\} \subset S, \forall p \in N \setminus S).$$

Together with (10) this implies:

$$\ell_i \alpha_{jp} = \ell_j \alpha_{ip} \qquad (\forall \{i,j\} \subset S, p \in N \setminus S). \tag{12}$$

Putting (11) and (12) together, we have:

$$\ell_i l_p \alpha_{ip} = \ell_j l_q \alpha_{jq} \qquad (\forall \{i,j\} \subset S, \{p,q\} \subset N \setminus S).$$

By adding a suitable multiple of the implicit equation (4) to the equation $\alpha^T d = \beta$, we can assume that:

$$\alpha_{ip} \quad = 0 \quad (\forall i \in S, p \in N \setminus S).$$

The left-hand side of the equation $\alpha^T d = \beta$ is now a non-negative linear combination of the left-hand side of the clique inequality on S and the left-hand side of the clique inequality on $N \setminus S$. But it is obvious that the left-hand side of the clique inequality on $N \setminus S$ can vary when the clique inequality on S holds at equality. Thus, the weight of the former in the linear combination must be zero.

The cases in which $S \in \{2, n-2, n-1\}$ are similar, but easier. □

An immediate consequence of Theorem 3 is:

Corollary 2 *The lower bounds $d_{ij} \geq (\ell_i + \ell_j)/2$ induce facets of $P(n, \ell)$.*

Note that the clique inequalities meet the condition (9) given in Theorem 2.

3.2 Hypermetric inequalities

In Subsection 2.4, we showed that valid inequalities for the cut cone CC_n lead to valid inequalities for $P(n, \ell)$. In this subsection, we consider the well-known *hypermetric* inequalities for CC_n. They take the form:

$$\sum_{\{i,j\} \subset N} b_i b_j d_{ij} \leq 0 \qquad (\forall b \in \mathbb{Z}^n : \sigma(b) = 1), \tag{13}$$

where $\sigma(b)$ denotes $\sum_{i \in N} b_i$. See [9] for a survey of the literature on hypermetric inequalities. We recall that the hypermetric inequalities with

$b \in \{0, \pm 1\}^n$ are called *pure*. The pure hypermetric inequalities include the following well-known *triangle* inequalities as a special case:

$$d_{ij} - d_{ik} - d_{jk} \leq 0 \qquad (\forall \{i,j\} \subset N, k \in N \setminus \{i,j\}). \tag{14}$$

Corollary 1 implies that the hypermetric inequalities are valid for $P(n,\ell)$. The following proposition states that only the pure ones are of interest:

Proposition 3 *A hypermetric inequality (13) induces a non-empty face of $P(n,\ell)$ if and only if it is pure.*

Proof. Suppose that we are given a vector $b \in \{0, \pm 1\}^n$ that defines a pure hypermetric inequality. By symmetry, we can assume that there exists an odd integer $1 \leq p \leq n$ such that:

- $b_i = 1$ if $1 \leq i \leq p$ and i is odd
- $b_i = -1$ if $1 < i < p$ and i is even
- $b_i = 0$ if $p < i \leq n$.

The identity layout then satisfies the pure hypermetric inequality at equality. Therefore, the inequality induces a non-empty face of $P(n,\ell)$.

We will show later (Proposition 5 in Subsection 3.4) that non-pure hypermetric inequalities can be strengthened by decreasing their right-hand side. Therefore, non-pure hypermetric inequalities do not define a non-empty face of $P(n,\ell)$. □

For our next result, we will find it helpful to define $S = \{i \in N : b_i = 1\}$ and $T = \{i \in N : b_i = -1\}$. Note that, in the case of a pure hypermetric inequality, we have $|T| = |S| - 1$.

Theorem 4 *Pure hypermetric inequalities induce facets of $P(n,\ell)$ if and only if $|S| + |T| \leq n - 2$.*

Proof. For the sake of brevity, we only sketch the proof. First, one shows that a layout π satisfies the hypermetric inequality at equality if and only if the facilities in S 'alternate' with facilities in T; that is, if and only if there exists a numbering $s_1, \ldots, s_{|S|}$ of the facilities in S and a numbering $t_1, \ldots, t_{|T|}$ of the facilities in T such that $\pi(s_i) < \pi(t_i) < \pi(s_{i+1})$ for $i = 1, \ldots, |S|$.

Next, one shows that, if $n = |S| + |T|$, then every layout satisfying the hypermetric inequality at equality also satisfies the equations

$$\sum_{j \in S \setminus \{i\}} d_{ij} - \sum_{j \in T} d_{ij} = \ell(N \setminus \{i\})/2 \qquad (\forall i \in S).$$

Then, one shows that, if $n = |S| + |T| + 1$, then every layout satisfying the hypermetric inequality at equality also satisfies the equation

$$\sum_{j \in S} d_{ij} - \sum_{j \in T} d_{ij} = \ell(N)/2,$$

where $\{i\} = N \setminus (S \cup T)$.

So suppose that $n \geq |S|+|T|+2$ and let the equation $\alpha^T d = \beta$ be satisfied by all layouts in which the hypermetric inequality holds at equality. Let π be a layout in which a facility in S occupies the first position, a facility in T occupies the fourth position, and facilities in $N \setminus (S \cup T)$ occupy the second and third positions. Just as in previous proofs, exchanges of facilities in the first three positions imply:

$$\ell_q \alpha_{ip} = \ell_p \alpha_{iq} \qquad (\forall i \in S, \{p, q\} \subset N \setminus (S \cup T)).$$

Similarly, exchanges of facilities in the second to fourth positions imply:

$$\ell_q \alpha_{ip} = \ell_p \alpha_{iq} \qquad (\forall i \in T, \{p, q\} \subset N \setminus (S \cup T)).$$

These equations fix the ratios between all pairs of α coefficients apart from those involving only facilities in $S \cup T$. By adding or subtracting a suitable multiple of the implicit equation (4), we can assume that all of the α coefficients are zero apart from those involving facilities in $S \cup T$.

Finally, a series of further exchange arguments shows that:

$$\begin{aligned} \alpha_{ij} &= -\alpha_{ik} \quad (\forall \{i, j\} \subset S, k \in T) \\ \alpha_{ij} &= -\alpha_{ik} \quad (\forall \{i, j\} \subset T, k \in S). \end{aligned}$$

Thus, the equation $\alpha^T d = \beta$ is equivalent to the pure hypermetric inequality (in equation form). □

Corollary 3 *The triangle inequalities (14) induce facets of $P(n, \ell)$ if and only if $n \geq 5$.*

Note that the pure hypermetric inequalities also meet the condition (9) given in Theorem 2.

3.3 Strengthened pure negative-type (SPN) inequalities

It is known [9] that the inequalities (13) remain valid for the cut cone when $\sigma(b) = 0$, in which case they are called *negative-type* inequalities. Negative-type inequalities do not define facets of the cut cone, and therefore do not induce facets of $P(n, \ell)$ either. Interestingly, however, we can obtain facets of $P(n, \ell)$ by taking *pure* negative-type inequalities and adjusting the right-hand side.

As before, we will find it helpful to define $S = \{i \in N : b_i = 1\}$ and $T = \{i \in N : b_i = -1\}$. Then, a pure negative-type inequality can be written in the form:

$$\sum_{i \in S, j \in T} d_{ij} - \sum_{\{i,j\} \subset S} d_{ij} - \sum_{\{i,j\} \subset T} d_{ij} \geq 0.$$

Moreover, we have $|T| = |S|$ in the pure case.

We are now ready to present a strengthened version of the pure negative-type inequalities:

Proposition 4 *For all $S \subset N$ and all $T \subset N \setminus S$ with $|S| = |T|$, the following 'strengthened pure negative-type' (SPN) inequality is valid for $P(n, \ell)$ and induces a non-empty face:*

$$\sum_{i \in S, j \in T} d_{ij} - \sum_{\{i,j\} \subset S} d_{ij} - \sum_{\{i,j\} \subset T} d_{ij} \geq (\ell(S) + \ell(T))/2. \qquad (15)$$

Proof. Since the inequalities (15) satisfy the condition (9) given in Theorem 2, we can assume that $S \cup T = N$. Let S and T be given, and let $q = |S| = |T| = n/2$. Moreover, let π be a given layout, and let d^* be the corresponding distance vector. For a given $i \in N$, let $s(i)$ and $t(i)$ be the number of facilities in S and T, respectively, that lie to the left of facility i in the layout π. When $i \in S$, the number of facilities in S and T lying to the right of facility i is $q - s(i) - 1$ and $q - t(i)$, respectively. The contribution of ℓ_i to the left-hand side of (15), computed with respect to d^*, can then be shown to equal:

$$\frac{1}{2} + (s(i) - t(i))(s(i) - t(i) + 1).$$

Similarly, when $i \in T$, the contribution of ℓ_i to the left-hand side can be shown to equal:

$$\frac{1}{2} + (s(i) - t(i))(s(i) - t(i) - 1).$$

Thus, the left-hand side of (15) is equal to:

$$\frac{1}{2}(\ell(S)+\ell(T))+\sum_{i \in S}(s(i)-t(i))(s(i)-t(i)+1)\ell_i+\sum_{i \in T}(s(i)-t(i))(s(i)-t(i)-1)\ell_i. \qquad (16)$$

Since the $s(i)$ and $t(i)$ are integers, the two summation terms in (16) are non-negative. This proves validity. Moreover, the two summation terms are equal to zero when the facilities in S occupy the odd positions and the facilities in T occupy the even positions (or vice-versa). This shows that the SPN inequality induces a non-empty face of $P(n, \ell)$. □

It turns out that all SPN inequalities induce facets.

Theorem 5 *The SPN inequalities always induce facets of $P(n, \ell)$.*

Proof. For the sake of brevity, we only sketch the proof. First, one shows that a layout satisfies the SPN inequality at equality if and only if there exists a numbering $s_1, \ldots, s_{|S|}$ of the facilities in S and a numbering $t_1, \ldots, t_{|T|}$ of the facilities in T such that, for $i = 1, \ldots, |S|$, facility s_i is adjacent to facility t_i in the layout. Then, as usual, suppose the equation $\alpha^T d = \beta$ is satisfied by all layouts in which the SPN inequality holds at equality. Similar exchange arguments to those used in previous proofs show the following:

$$\begin{aligned} l_q \alpha_{ip} &= l_p \alpha_{iq} \quad (\forall i \in S, \{p, q\} \subset N \setminus (S \cup T)) \\ l_q \alpha_{ip} &= l_p \alpha_{iq} \quad (\forall i \in T, \{p, q\} \subset N \setminus (S \cup T)). \end{aligned}$$

Just as for the pure hypermetric inequalities, we can then assume that all of the α coefficients are zero apart from those involving facilities in $S \cup T$.

Finally, a series of further exchange arguments shows that:

$$\begin{aligned} \alpha_{ij} &= -\alpha_{ik} \quad (\forall \{i, j\} \subset S, k \in T) \\ \alpha_{ij} &= -\alpha_{ik} \quad (\forall \{i, j\} \subset T, k \in S). \end{aligned}$$

and therefore the equation $\alpha^T d = \beta$ is equivalent to the SPN inequality (in equation form). □

Note that the SPN inequalities reduce to lower bounds of the form $d_{ij} \geq (\ell_i + \ell_j)/2$ when $|S| = |T| = 1$. Thus, the lower bounds are a special case of both clique and SPN inequalities.

3.4 Rounded positive semidefinite inequalities

At the end of Subsection 2.4, it was noted that feasible d vectors are a non-negative linear combination of $n - 1$ distinct cut vectors. This fact is now exploited to derive a class of inequalities that includes the pure hypermetric and SPN inequalities as a special case.

Our starting point is the well-known fact that every cut vector satisfies the following valid inequalities [9]:

$$\sum_{\{i,j\} \subset N} b_i b_j d_{ij} \leq \sigma(b)^2/4 \qquad (\forall b \in \mathbb{R}^n).$$

These are sometimes called *positive semidefinite* (psd) inequalities, because they define the feasible region of the well-known semidefinite programming relaxation of the max-cut problem. Moreover, when b is integral and $\sigma(b)$ is odd, the right-hand side of the psd inequality is fractional, and can therefore be rounded down to an integer while maintaining validity (see, e.g., [7, 9]).

The following proposition shows that there exists an analogous class of rounded psd inequalities for the SRFLP:

Proposition 5 *The following 'rounded psd' inequalities are valid for $P(n,\ell)$:*

$$\sum_{\{i,j\}\subset N} b_i b_j d_{ij} \leq \frac{1}{2}\sum_{i\in N}\left\lfloor \frac{\sigma(b)^2 - b_i^2}{2}\right\rfloor \ell_i \qquad (\forall b \in \mathbb{Z}^n). \tag{17}$$

Proof. Let π be a given layout, and let d^* be the corresponding distance vector. For a given $i \in N$, let $B(i)$ be the sum of the b coefficients over all facilities to the left of i in the layout. Note that the sum of the b coefficients over the facilities to the right of i in the layout must be $\sigma(b) - B(i) - b_i$. The contribution of ℓ_i to the left-hand side of (17), computed with respect to d^*, is therefore:

$$B(i)(\sigma(b) - B(i) - b_i) + \frac{1}{2} b_i(\sigma(b) - b_i).$$

This quantity is maximised when $B(i)$ is equal to $\lfloor(\sigma(b) - b_i)/2\rfloor$, in which case the contribution of ℓ_i to the left-hand side becomes

$$\lfloor(\sigma(b) - b_i)/2\rfloor\lceil(\sigma(b) - b_i)/2\rceil + \frac{1}{2} b_i(\sigma(b) - b_i).$$

With a little work, this can be re-written as:

$$\frac{1}{2}\left\lfloor \frac{\sigma(b)^2 - b_i^2}{2}\right\rfloor.$$

Multiplying this quantity by ℓ_i, and summing over all $i \in N$, yields the desired right-hand side. □

Notice that the rounded psd inequalities (17) reduce to pure hypermetric inequalities when $b \in \{0, \pm 1\}^n$ and $\sigma(b) = 1$, and to SPN inequalities when $b \in \{0, \pm 1\}^n$ and $\sigma(b) = 0$. They therefore induce facets under certain conditions. On the other hand, the rounded psd inequalities do not in general meet the condition (9) of Theorem 2, and therefore they do not always induce facets. Nevertheless, we have found that they make useful cutting planes.

3.5 Star inequalities

Before introducing our last class of inequalities, we will need some additional notation. For any $S \subset N$, we define the following quantity:

$$SSP(S) = \max\left\{\sum_{i\in S}\ell_i x_i : \sum_{i\in S}\ell_i x_i \leq l(S)/2,\ x \in \{0,1\}^{|S|}\right\}.$$

(We denote it by '$SSP(S)$', because it is obtained by solving a subset-sum problem.) We also write $\gamma(S) := \ell(S) - 2\ SSP(S)$. For example, if $S =$

$\{1,2,3\}$ and $(\ell_1,\ell_2,\ell_3) = (3,5,6)$, we have $\ell(S) = 14$, $SSP(S) = 6$ and $\gamma(S) = 2$.

The quantity $\gamma(S)$ is related to the notion of the *gap* of an integer sequence, defined in Laurent & Poljak [18]. Computing $SSP(S)$, and therefore $\gamma(S)$, is NP-hard in the weak sense, but can be performed in pseudo-polynomial time by dynamic programming.

We have the following result:

Proposition 6 *For any $i \in N$ and any $S \subseteq N \setminus \{i\}$, the following 'star' inequality is valid for $P(n,\ell)$:*

$$\sum_{j \in S} \ell_j d_{ij} \geq \frac{1}{4}\left(\ell(S)^2 + \gamma(S)^2\right) + \frac{1}{2}\ell_i \ell(S). \tag{18}$$

Proof. Let S_L (respectively, S_R) be the set of facilities to the left (right) of facility i in a layout. One can show (e.g., by induction) that the contribution of the facilities in S_L to the left hand side of (18) is at least $\ell(S_L)\ell(S_L \cup \{i\}/2$. An analogous result holds for the facilities in S_R. The left-hand side of (18) is therefore at least

$$\ell_i \ell(S) + \frac{1}{2}\left(\ell(S_L)^2 + \ell(S_R)^2\right).$$

This quantity is minimised when $\ell(S_L) = SSP(S)$ and $\ell(S_R) = l(S) - SSP(S)$ (or vice-versa), in which case it reduces to the right-hand side of (18). $\square$

Note that, when $|S| = 1$, the star inequalities reduce to (facet-inducing) lower-bounds of the form $d_{ij} \geq (\ell_i + \ell_j)/2$. In general, however, the star inequalities do not induce facets, since they do not meet the condition (9) of Theorem 2. Nevertheless, they can be shown (with a little work) to induce faces of dimension at least $\binom{n-1}{2} - |S|^2/2$. In any case, we found them to be useful in our branch-and-cut algorithm. We leave to future research the problem of strengthening the star inequalities in order to make them facet-defining.

Observe that, in the special case of MinLA, the star inequalities reduce to:

$$\sum_{j \in S} d_{ij} \geq \left\lfloor (|S|-1)^2/4 \right\rfloor \qquad (i \in N, S \subseteq N \setminus \{i\}). \tag{19}$$

These validity of these inequalities for MinLA was observed by Liu & Vannelli [19].

4 A Branch-and-Cut Algorithm

In this section, we describe a branch-and-cut algorithm that uses the inequalities that we described in the previous section. We discuss separation in Subsection 4.1, branching in Subsection 4.2, the primal heuristic in Subsection 4.3, and other minor considerations in Subsection 4.4.

4.1 Separation

The *separation problem* for a given class of valid inequalities is this: given n, ℓ, and a vector $d^* \in \mathbb{R}^{\binom{n}{2}}$, either find an inequality in that class violated by d^*, or prove that none exists [13]. Separation algorithms, either exact or heuristic, are an essential component of branch-and-cut algorithms. We now briefly describe our separation algorithms for various classes of inequalities.

Clique inequalities

We conjecture that the separation problem for the clique inequalities (6) is NP-hard. Therefore, we devised a simple greedy heuristic. The heuristic works by taking a previously-generated clique inequality (which could be a 'mere' lower bound of the form $d_{ij} \geq (\ell_i + \ell_j)/2$), and iteratively inserting facilities into the set S in a greedy manner, until either a violated inequality is found or $|S| = n - 1$. This procedure is applied to every clique inequality in the LP whose slack is small (less than 0.1).

If this heuristic is implemented in a naive way, it takes $\mathcal{O}(n^3)$ time per source inequality. With appropriate data structures, however, it can be implemented so that it takes only $\mathcal{O}(n^2)$ time per source inequality. We omit details, for the sake of brevity.

Triangle inequalities

The separation problem for the triangle inequalities (14) can be solved in $O(n^3)$ time by brute-force enumeration. If this is done in a naive way, however, a huge number of violated inequalities can be generated, which can lead to memory problems in the LP solver. For this reason, we used the following routine:

For each pair $\{i, j\}$ of facilities
    Find the facility $k \in N \setminus \{i, j\}$ that minimises $d^*_{ik} + d^*_{jk}$.
    If the inequality (14) is violated, output it.

This routine outputs at most $\binom{n}{2}$ inequalities, which turned out to be much more manageable.

A similar procedure can be used to detect violated SPN inequalities with $|S| = |T| = 2$, in $O(n^4)$ time.

Rounded psd inequalities

Recall that the rounded psd inequalities (17) include the pure hypermetric and SPN inequalities as special cases. We therefore devised a separation heuristic for the rounded psd inequalities. The heuristic simply takes a previously-generated rounded psd inequality (which could be a 'mere' triangle inequality or an SPN inequality with $|S| = |T| = 2$), and checks whether the associated b vector can be adjusted in order to obtain a violated rounded psd inequality. The adjustments considered are:

- incrementing b_i for some $i \in N$,
- decrementing b_i for some $i \in N$,
- simultaneously incrementing b_i for some $i \in N$ and decrementing b_j for some $j \in N \setminus \{i\}$.

If this heuristic is implemented in a naive way, it takes $\mathcal{O}(n^4)$ time per source inequality. With appropriate data structures, however, it can be implemented so that it takes only $\mathcal{O}(n^2)$ time per source inequality. As before, we omit details for the sake of brevity.

Star inequalities

Finally, we consider the star inequalities (18). Since computing the right-hand side of these inequalities is already NP-hard, it is certain that the separation problem for them is also NP-hard. Consider, however, the following 'weak star' inequalities:

$$\sum_{j \in S \setminus \{i\}} \ell_j d_{ij} \geq \frac{1}{4} l(S)^2 - \frac{1}{2} \sum_{j \in S} l_j^2 \qquad (i \in N, S \subseteq N \setminus \{i\}) \tag{20}$$

It turns out that the separation problem for these weak star inequalities can be solved exactly in polynomial time. We write the inequalities in the following alternative form:

$$\sum_{j \in S} \left(\ell_j d_{ij} + \frac{\ell_j^2}{4} \right) - \sum_{\{j,k\} \subset S} \left(\frac{\ell_j \ell_k}{2} \right) \geq 0 \qquad (i \in N,\ S \subseteq N \setminus \{i\}).$$

Now, let i be fixed, and let y_j, for $j \in N \setminus \{i\}$, be a 0-1 variable taking the value 1 if and only if $j \in S$. Clearly, finding the set $S \subset N \setminus \{i\}$ that maximises the violation of the weak star inequality amounts to minimising:

$$\sum_{j \in N \setminus \{i\}} \left(\ell_j d_{ij}^* + \frac{\ell_j^2}{4} \right) y_j - \sum_{\{j,k\} \subset N \setminus \{i\}} \left(\frac{\ell_j \ell_k}{2} \right) y_j y_k.$$

This is an unconstrained quadratic program in the binary variables y_j, with non-positive quadratic terms. It is well-known (e.g., Picard & Ratliff [24]) that such problems can be solved in $O(n^3)$ time via a max-flow computation.

Our separation heuristic for star inequalities is therefore as follows: for each $i \in N$ in turn, run the exact separation algorithm for weak star inequalities. If a violated weak star inequality is found, solve a subset-sum problem to compute $\gamma(S)$, convert the weak star inequality into a star inequality, and check for violation. All violated inequalities found (if any) are added to the LP.

4.2 Branching rule

At this stage, it is worth pointing out that there is no known formulation of the SRFLP that involves only the d_{ij} variables. If one introduces additional 0-1 variables, and constraints linking them to the d_{ij} variables, one can easily formulate the SRFLP as a mixed 0-1 LP [1, 2, 16, 20]. Here, however, we have decided to avoid the use of additional variables. This means that we have to use a specialised branching rule.

After some experimentation, we settled on the following branching rule. We first sort the facilities in decreasing order of length, and impose that facility 1 is to the left of facility 2 in the layout. The root node is then represented by the permutation $\{1, 2\}$. A node at depth p in the branch-and-bound tree is represented by a permutation of $\{1, \ldots, p + 2\}$. At that node, we require the first $p + 2$ facilities to appear in the given order in the layout. To ensure this, we add equations to the LP.

For example, suppose a node at depth 1 is represented by the permutation $1 - 2 - 3$. This means that facility 1 must be to the left of facility 2, which in turn must be to the left of facility 3. Therefore, the triangle inequality $d_{12} + d_{23} - d_{13} \geq 0$ must hold at equality. Thus, at that node, we add the equation $d_{12} + d_{23} - d_{13} = 0$ to the LP.

Now suppose that a child node at depth 2 is represented by the permutation $1 - 4 - 2 - 3$. This means that facility 1 must be to the left of facility 4, and facility 4 must be to the left of facility 2. To ensure this, we change an additional *two* triangle inequalities to equations:

$$d_{14} + d_{24} - d_{12} = 0$$
$$d_{23} + d_{24} - d_{34} = 0.$$

There is no need to also impose that facility 4 lies between facilities 1 and 3, since this is implied by the other equations at that node.

In general, we impose $(p - 1)(p - 2)/2$ equations to fix the order of p facilities.

4.3 Primal heuristic

In this subsection, we describe a *primal heuristic* for the SRFLP, which takes an LP solution vector d^* and produces a feasible layout.

The heuristic is based on the following observations:

- For any $\{i, j\} \subset N$, the value d^*_{ij} can be interpreted as an estimate of the optimal distance between the centroids of facilities i and j.
- In a feasible solution to the SRFLP, the centroids of the facilities are points in the real line $\mathbb{R}$.
- We can assume that d^* satisfies all triangle inequalities, and therefore d^* defines a metric on the set N.

This is exactly the kind of situation in which statisticians use *Multi-Dimensional Scaling* (MDS). So, we decided to feed d^* into the classical MDS procedure of Torgerson [29]. This procedure involves the construction of a certain $n \times n$ matrix, followed by a single eigenvalue computation. It is extremely fast in practice.

The MDS procedure produces a placement of the centroids in the real line. This placement need not correspond to a feasible layout, since the facilities themselves may overlap. To fix this, it suffices simply to use the ordering of the centroids, rather than their absolute positions.

In our experience, the layouts obtained using MDS are rather good. Nevertheless, in many cases they can be improved further by applying local search. We therefore use a simple 2-opt procedure, based on iteratively swapping pairs of facilities. The bounds obtained turn out to be remarkably tight, as shown in the next section.

4.4 Other ingredients

We include the following constraints in the initial LP relaxation:

- the implicit equation (4),
- the lower bounds $d_{ij} \geq (\ell_i + \ell_j)/2$ (which are handled implicitly with the bounded version of the simplex method),
- for each $i \in N$, the clique inequality that has $S = N \setminus \{i\}$
- for each $i \in N$, the star inequality that has $S = N \setminus \{i\}$ and, if it is different, the star inequality that has $S = \{j \in N \setminus \{i\} : c_{ij} > 0\}$.

We remark that the clique inequalities with $S = N \setminus \{i\}$ can be re-written, using the implicit equation (4), to take the following simple form:

$$\sum_{j \in S} \ell_j d_{ij} \leq \ell(N)\ell(S)/2.$$

As a result, the initial LP contains only $O(n^2)$ non-zero constraint coefficients. It can therefore can be solved very quickly by primal simplex.

The separation routines are called in the following order:

1. exact separation for triangle inequalities
2. heuristic separation for clique inequalities
3. heuristic separation for rounded psd inequalities
4. heuristic separation for star inequalities.
5. exact separation for SPN inequalities with $|S| = |T| = 2$.

So, for example, clique separation is called only if no violated triangle inequalities can be found. (We leave star and SPN separation to the end because they are rather time-consuming, taking $\mathcal{O}(n^4)$ time each.)

The separation routines and the primal heuristic are called at every node of the branch-and-cut tree. A node is fathomed if its lower bound exceeds the best upper bound, or if the LP solution represents a feasible layout. (One can easily check in $\mathcal{O}(n^2)$ time if this is the case.)

5 Computational Experiments

In this section, we report the results of some computational experiments.

5.1 Cutting planes only

First, we report results obtained using cutting planes only, without any branching. For these experiments, the LPs were solved with the CPLEX 12.1 callable library on a 2.5 GHz Pentium Dual Core PC, with 2GB of RAM, under Windows Vista.

We began by testing the cutting-plane algorithm on 9 'classical' instances from the literature (see Table 1). The 'S' instances are due to Simmons [26]. The two 'H' instances were derived by Heragu & Kusiak [15], by modifying the famous instances of the Quadratic Assignment Problem (QAP) due to Nugent *et al.* [21].

The results are shown in Table 1. The first two columns show the instance name and the optimal solution value. (The optimal solutions for the 'S' instances were computed by Amaral [2], whereas those for the two 'H' instances were found by Anjos & Vannelli [5]). The next two columns show the lower bound when the cutting-plane algorithm terminates, and the upper bound obtained with our multi-dimensional scaling heuristic. Bounds that are optimal are shown in bold font. The next two columns show the percentage gap between the lower bound and the optimum (`gap1`), and between the upper bound and the optimum (`gap2`). The final two columns show the number of cutting-plane iterations and the time taken by the cutting-plane algorithm (in seconds).

Next, we ran the algorithm on 8 other instances of Heragu & Kusiak [15], which have a so-called 'clearance requirement' of 0.01 length units between each pair of consecutive facilities. We call these 'C' instances in Table 2. We also tested the algorithm on 10 newer instances created by Anjos & Vannelli [5], which were again based on the Nugent *et al.* QAP instances. We call these 'N' instances in Table 3. The optimal solutions for the 'C' and 'N' instances were also presented in [5].

It can be seen that the gap between the lower bound and the optimum is below 3.5% in all cases, and in most cases it is much smaller. In fact, our

Inst.	Opt.	LB	UB	%gap1	%gap2	Iter.	Time (s)
S5	151.0	**151.0**	**151.0**	0.00	0.00	8	0.051
S8	801.0	797.6	**801.0**	0.43	0.00	20	0.205
S8H	2324.5	**2324.5**	**2324.5**	0.00	0.00	17	0.101
S9	2469.5	**2469.5**	**2469.5**	0.00	0.00	15	0.126
S9H	4695.5	4664.2	**4695.5**	0.67	0.00	26	0.242
S10	2781.5	2778.2	**2781.5**	0.12	0.00	25	0.327
S11	6933.5	6886.8	**6933.5**	0.67	0.00	34	0.483
H20	15549.0	15174.6	**15549.0**	2.41	0.00	91	8.752
H30	44965.0	44136.7	45158.0	1.84	0.43	127	58.968

Table 1: Cutting-plane results for 9 ‘classical’ SRFLP instances.

Inst.	Opt.	LB	UB	%gap1	%gap2	Iter.	Time (s)
C5	1.100	**1.100**	**1.100**	0.00	0.00	5	0.031
C6	1.990	**1.990**	**1.990**	0.00	0.00	6	0.047
C7	4.730	4.678	**4.730**	1.09	0.00	10	0.063
C8	6.295	6.245	**6.295**	0.79	0.00	12	0.078
C12	23.365	22.670	23.395	2.98	0.13	37	0.796
C15	44.600	43.981	**44.600**	1.39	0.00	52	2.325
C20	119.710	117.239	119.990	2.06	0.23	99	10.125
C30	334.870	326.663	336.080	2.45	0.36	138	69.124

Table 2: Cutting-plane results for 8 SRFLP instances with clearance requirement.

Inst.	Opt.	LB	UB	%gap1	%gap2	Iter.	Time (s)
N25-1	4618.0	4534.4	**4618.0**	1.81	0.00	112	32.744
N25-2	37166.5	35869.6	37449.5	3.49	0.76	126	29.094
N25-3	24301.0	23653.0	24466.0	2.67	0.68	141	33.010
N25-4	48291.5	46681.6	48537.5	3.33	0.51	137	34.991
N25-5	15623.0	15107.4	15725.0	3.30	0.65	186	47.206
N30-1	8247.0	8134.6	8267.0	1.36	0.24	143	105.020
N30-2	21582.5	21226.8	21754.5	1.65	0.80	139	77.891
N30-3	45449.0	44239.8	45522.0	2.66	0.16	174	86.019
N30-4	56873.5	56000.4	56904.5	1.54	0.05	167	88.936
N30-5	115268.0	113039.0	115304.0	1.93	0.03	230	116.064

Table 3: Cutting-plane results for 10 newer SRFLP instances.

Inst.	Nodes	Time (s)	Inst.	Nodes	Time (s)	Inst.	Nodes	Time (s)
S5	1	0.061	C5	1	0.047	N25-1	27619	26384
S8	13	0.466	C6	1	0.062	N25-2	1640	2315.4
S8H	1	0.114	C7	19	0.266	N25-3	7207	5141.2
S9	1	0.135	C8	4	0.141	N25-4	2099	2373.9
S9H	50	2.376	C12	63	3.978	N25-5	2860	4689.5
S10	4	0.414	C15	144	9.594	N30-1	61716	122451
S11	4	0.674	C20	865	312.45	N30-2	7397	14213
H20	251	142.13	C30	28158	64183	N30-3	25508	47292
H30	131885	101269	—	—	—	N30-4	2054	3500.3
—	—	—	—	—	—	N30-5	18188	47031

Table 4: Branch-and-cut results for the SRFLP instances.

lower bounds are very similar in quality to the ones obtained using the SDP-based approach of Anjos *et al.* [4]. The only approaches in the literature that give stronger bounds are the ones of [3, 5]; but for those, the running times are much higher.

It is also interesting to observe that the multi-dimensional scaling heuristic gives remarkably tight upper bounds. Indeed, the average gap between the upper bound and the optimum is only 0.19%, the maximum gap is only 0.80%, and the heuristic solution is optimal in 14 cases out of 27.

5.2 Branch-and-cut

Now, we move on to the results obtained with the full branch-and-cut algorithm. Table 4 shows, for each instance, the instance name, the number of branch-and-cut nodes and the total time taken to solve the instance to proven optimality (in seconds).

As can be seen, the algorithm is capable of solving all of the instances using reasonable computational resources. The running times are very good when $n \leq 20$, but rather excessive for larger values of n. We remark that the optimal solution was typically found very early on. The remainder of the time (over 99%) was spent proving optimality. Moreover, the bottleneck in all cases was the time taken by the LP solver, rather than the separation time. This is probably due to the fact that some our inequalities (namely, the rounded psd and clique inequalities) can be rather dense.

5.3 Experiments with large instances

Finally, we report some results obtained when applying our method to some much larger instances. For these experiments, we used a slightly more powerful Intel Core 2 Duo 3.33GHz computer with 3Gb RAM. Moreover, since solving the LP at the root node is already a challenge for the instances considered, we report results only for the root node.

Inst.	nodes	edges	UB	LB1	LB2	LB3	Iter.	Time (h)
gd95c	62	144	506	292	443	471.5	448	9.96
gd96c	65	125	519	241	402	463.5	298	4.60

Table 5: Cutting-plane results for 2 MinLA instances.

Inst.	ℓ-range	c-range	LB	UB	%gap	Iter.
1	$[7, 23]$	$[0, 100]$	139045719	144331884.5	3.66	96
2	$[5, 14]$	$[0, 100]$	82676930	86065390.0	3.94	102
3	$[3, 7]$	$[0, 5]$	2155583	2234803.5	3.54	97

Table 6: Cutting-plane results for 3 SRFLP instances with $n = 110$, with time limit of 2.5 days.

First, we experimented with two MinLA instances, taken from Petit [22]. In Table 5 we report the following for each instance: the name, the number of nodes and edges, the best known upper bound (taken from Rodriguez-Tello *et al.* [25]), the best known combinatorial lower bound (from Petit [22]), the lower bound from the cutting-plane algorithm of Caprara *et al.* [8], the lower bound from our cutting-plane algorithm, the number of iterations, and the total time in hours.

It is apparent that our cutting planes close around half of the gap between the previously best-known lower bounds and the best-known upper bounds. On the other hand, the running times are relatively large.

Finally, to test the limits of our algorithm, we created three large-scale SRFLP instances with $n = 110$. These were created by taking lengths ℓ_i randomly from a uniform distribution, and taking costs c_{ij} randomly from another uniform distribution. Table 6 displays the following for each instance: the instance number, the range of the ℓ_i distribution, the range of the c_{ij} distribution, the lower bound at the end of the cutting-plane phase, the upper bound obtained from our primal heuristic, the percentage gap between the lower and upper bounds, and the number of iterations. A time limit of 2.5 days was imposed.

We see that the percentage gaps were between 3 and 4 percent. We remark that similar percentage gaps were obtained in [6] for $n = 100$, but only by running their SDP-based algorithm for over a week per instance on a comparable machine.

6 Concluding Remarks

We have performed the first ever polyhedral study of the SRFLP, deriving several huge classes of valid inequalities, and giving conditions for them to induce facets. Our cutting planes yield excellent lower and upper bounds very quickly for instances with $n \leq 30$ or so, but computing times can be quite long for larger instances. The full branch-and-cut algorithm is capable of solving instances with $n \leq 30$ to proven optimality, but suffers from excessive time and memory requirements for larger values of n.

In our view, these results are rather promising and merit further research. One obvious line of research would simply be the derivation of additional facet-inducing inequalities and separation algorithms. A more interesting possibility would be to apply our methodology to other similar facility layout problems, such as the problems of locating facilities on a circle or on a rectangular grid. Finally, we believe that a more detailed study of MinLA, which is an important special case of the SRFLP, would also be very worthwhile.

Acknowledgement: The research of the first author was supported by CNPq-Brazil Grant 500212/02-3. The research of the second author was supported by the Engineering and Physical Sciences Research Council under grant EP/D072662/1. The authors are grateful to an anonymous referee for helpful comments on an earlier version of the paper.

References

[1] Amaral, A.R.S.: On the exact solution of a facility layout problem. Eur. J. Opl Res. **173**, 508–518 (2006)

[2] Amaral, A.R.S.: An exact approach for the one-dimensional facility layout problem. Oper. Res. **56**, 1026–1033 (2008)

[3] Amaral, A.R.S.: A new lower bound for the single row facility layout problem. Discr. Appl. Math. **157**, 183–190 (2009)

[4] Anjos, M.F., Kennings, A., Vannelli, A.: A semidefinite optimization approach for the single-row layout problem with unequal dimensions. Discr. Opt. **2**, 113–122 (2005)

[5] Anjos, M.F., Vannelli, A.: Computing globally optimal solutions for single-row layout problems using semidefinite programming and cutting planes. INFORMS J. Comput. **20**, 611–617 (2008)

[6] Anjos, M.F., Yen, G.: Provably near-optimal solutions for very large single-row facility layout problems. Optim. Meth. Soft. **24**, 805–817 (2009)

[7] Avis, D., Umemoto, J.: Stronger linear programming relaxations for max-cut. Math. Program. **97**, 451–469 (2003)

[8] Caprara, A., Letchford, A.N., Salazar, J.J.: Decorous lower bounds for minimum linear arrangement. INFORMS J. Comput., to appear.

[9] Deza, M.M., Laurent, M.: Geometry of Cuts and Metrics. Springer, New York (1997)

[10] Díaz, J., Petit, J., Serna, M.: A survey of graph layout problems. ACM Computing Surveys **34**, 313–356 (2002)

[11] Djellab, H., Gourgand, M.: A new heuristic procedure for the single-row facility layout problem. Int. J. Comp. Integrated Manufacturing **14**, 270–280 (2001)

[12] Garey, M.R., Johnson, D.S., Stockmeyer, L.: Some simplified NP-complete graph problems. Theor. Comp. Sci. **1**, 237–267 (1976)

[13] Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Springer, New York (1988)

[14] Helmberg, C., Rendl, F.: Solving quadratic (0,1)-problems by semidefinite programming and cutting planes. Math. Program. **82**, 291–315 (1998)

[15] Heragu, S.S., Kusiak, A.: Machine layout problem in flexible manufacturing systems. Oper. Res. **36**, 258–268 (1988)

[16] Heragu, S.S., Kusiak, A.: Efficient models for the facility layout problem. Eur. J. Opl Res. **53**, 1–13 (1991)

[17] Karp, R.M., Held, M.: Finite-state processes and dynamic programming. SIAM J. Appl. Math. **15**, 693–718 (1967)

[18] Laurent, M., Poljak, S.: Gap inequalities for the cut polytope. SIAM J. Matrix Anal. **17**, 530-547 (1996)

[19] W. Liu, A. Vannelli: Generating lower bounds for the linear arrangement problem. Discr. Appl. Math. **59**: 137–151 (1995)

[20] Love, R.F., Wong, J.Y.: On solving a one-dimensional allocation problem with integer programming. INFOR **14**, 139–143 (1976)

[21] Nugent, C.E., Vollmann, T.E., Ruml, J.: An experimental comparison of techniques for the assignment of facilities to locations. Oper. Res. **16**, 150–173 (1968)

[22] Petit, J.: Experiments on the minimum linear arrangement problem. J. Experimental Algorithmics **8**, article 2.3 (2003)

[23] Picard, J.-C., Queyranne, M.: On the one-dimensional space allocation problem. Oper. Res. **29**, 371–391 (1981)

[24] Picard, J.-C., Ratliff, H.: Minimum cuts and related problems. Networks **5**, 357-370 (1975)

[25] Rodriguez-Tello, E., Hao, J.-K., Torres-Jimenez, J.: An effective two-stage simulated annealing algorithm for the minimum linear arrangement problem. Comp. & Oper. Res. **35**, 3331–3346 (2008)

[26] Simmons, D.M.: One-dimensional space allocation: an ordering algorithm. Oper. Res. **17**, 812–826 (1969)

[27] Simmons, D.M.: A further note on one-dimensional space allocation. Oper. Res. **19**, 249 (1971)

[28] Solimanpur, M., Vrat, P., Shankar, R.: An ant algorithm for the single row layout problem in flexible manufacturing systems. Comp. & Oper. Res. **32**, 583–598 (2005)

[29] Torgerson, W.S.: Multidimensional scaling I: theory and method. Psychometrika **17**, 401–409 (1952)