Math. Program., Ser. A (2013) 141:453–477
DOI 10.1007/s10107-012-0533-z

FULL LENGTH PAPER

A polyhedral approach to the single row facility layout problem

André R. S. Amaral · Adam N. Letchford

Received: 20 January 2011 / Accepted: 8 March 2012 / Published online: 21 March 2012
© Springer and Mathematical Optimization Society 2012

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

Mathematics Subject Classification 90C57 (Polyhedral combinatorics, branch-and-bound, 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

A. R. S. Amaral
Departamento de Informática, Universidade Federal do Espírito Santo (UFES),
Vitória, ES 29060-900, Brazil
e-mail: amaral@inf.ufes.br

A. N. Letchford (✉)
Department of Management Science, Lancaster University,
Lancaster LA1 4YX, UK
e-mail: A.N.Letchford@lancaster.ac.uk

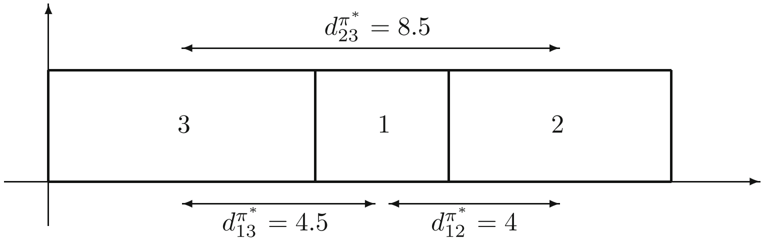

Fig. 1 Optimal layout π^* for an SRFLP instance with $n = 3$

weighted sum of the distances between all facility pairs, i.e., the quantity:

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

where Π denotes the set of all layouts and d_{ij}^{π} 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 Fig. 1, this corresponds to the distances $d_{12}^{\pi^*} = 4$, $d_{13}^{\pi^*} = 4.5$ and $d_{23}^{\pi^*} = 8.5$. The cost of π^* is $(4 \times 4) + (8 \times 4.5) + (9 \times 8.5) = 128.5$.

The SRFLP has many important practical applications [19,27,29]. 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. [15] for a survey of MinLA and other graph layout problems.)

MinLA is *NP*-hard in the strong sense (Garey et al. [17]), and therefore so is the SRFLP. In practice, the SRFLP is similar to the well-known quadratic assignment problem (QAP), in that instances with $n \geq 25$ or so can pose a serious challenge. For this reason, many authors have concentrated on heuristics; see, e.g., [13,16,19, 20,31]. There are, however, also several papers on lower bounding techniques and exact approaches, which we survey in the next section.

At present, one of the most successful approaches to *NP*-hard combinatorial optimisation problems is the so-called *polyhedral* approach, in which a family of polyhedra is associated with the problem, and linear inequalities that are valid for these polyhedra are used as cutting planes (see, e.g., [11,12]). 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 Sect. 2, we briefly review the literature on lower-bounding procedures and exact algorithms for the SRFLP. In Sect. 3, we define our polyhedra and establish some fundamental properties of them. In Sect. 4, we derive five (exponentially large) families of valid inequalities, and provide conditions for them to induce facets. In Sect. 5, we describe a branch-and-cut algorithm for the SRFLP. In Sect. 6, we present extensive computational results. Finally, concluding remarks are made in Sect. 7.

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:*

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

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

2 Literature review

To solve the SRFLP exactly, authors have suggested using combinatorial branch-and-bound algorithms [29,30] and dynamic programming [22,27]. Here, however, we concentrate on lower-bounding procedures and exact algorithms that are based on mathematical programming. From now on, we let LP and SDP stand for linear programming and semidefinite programming, respectively.

A first stream of papers is concerned with LP-based approaches:

- Love & Wong [25] and Heragu & Kusiak [20] presented mixed 0-1 LP formulations with $\mathcal{O}(n^2)$ binary variables, $\mathcal{O}(n^2)$ continuous variables and $\mathcal{O}(n^2)$ constraints. Unfortunately, the lower bound obtained by solving the continuous relaxation of these formulations is easily shown to be 0. This approach was only able to solve instances with $n \leq 11$.
- Amaral [1] presented a different mixed 0–1 LP formulation, with $\mathcal{O}(n^2)$ binary variables, $\mathcal{O}(n^2)$ continuous variables and $\mathcal{O}(n^3)$ constraints. The associated lower bound is easily shown to be:

$$\frac{1}{2} \sum_{1 \leq i < j \leq n} c_{ij} (\ell_i + \ell_j),$$

  which in fact was observed to be a valid lower bound by Simmons [29]. This approach proved to be faster, and instances with $n \leq 15$ could be solved in under an hour.
- Amaral [2] presented a mixed 0–1 LP formulation with $\mathcal{O}(n^2)$ binary variables, $\mathcal{O}(n^3)$ continuous variables, and $\mathcal{O}(n^3)$ constraints. The associated lower bounds are much stronger, making it possible to solve instances with $n \leq 18$ in a few hours.
- Finally, Amaral [3] presents a pure 0–1 LP formulation with $\mathcal{O}(n^3)$ binary variables and $\mathcal{O}(n^4)$ constraints, along with an exponentially large family of additional valid inequalities. With a pure cutting plane algorithm (no branching), he was able to solve instances with $n \leq 35$ in a few hours.

A second stream of papers is concerned with SDP relaxations:

- Anjos et al. [4] used a matrix variable of order $\binom{n}{2}$, together with $\mathcal{O}(n^3)$ linear equations. The lower bounds obtained after letting the SDP solver run for up to 10 h were typically within around 3 % of the optimum, and could be computed for instances with $n \leq 80$.
- Anjos and Vannelli [6] showed that the relaxation in [4] can be strengthened by adding $\mathcal{O}(n^6)$ linear inequalities. Since SDP software cannot cope with so many constraints, they presented results obtained by adding a small subset of them. The resulting bound turned out to be optimal for most instances with $n \leq 30$. The running times, however, were measured in hours or even days.
- Anjos and Yen [7] present an SDP relaxation that is a little weaker than the one in [4], but which has only $\mathcal{O}(n^2)$ linear equations. The lower bounds were typically within around 5 % of the optimum, and could be computed for instances with up to $n \leq 100$, though with running times of several days in some cases.
- Hungerländer & Rendl [21] strengthen the relaxation of [6] further, by using a matrix variable of order $\binom{n}{2}+1$, and adding $\mathcal{O}(n^5)$ more linear inequalities. Instead of incorporating the inequalities as cutting planes, they relax them in Lagrangian fashion and find good multipliers via the bundle method. They solve instances with $n \leq 40$ to proven optimality, and find bounds within 2 % of optimal for instances with n up to 100.

The leading exact algorithms at present appear to be those of Amaral [3] and Hungerländer and Rendl [21]. See also the survey Anjos and Liers [5].

3 Distance polytopes: fundamentals

3.1 Definition

Since there are several ways to formulate the SRFLP, one could define several different families of polyhedra. We have decided to work with what we call 'distance polytopes'. For all $1 \leq i < j \leq n$, let d_{ij} represent the distance between the centroids of facilities i and j in the layout. Then, for any integer $n \geq 2$ and length vector $\ell \in \mathbb{Z}^n_+$, the distance polytope $P(n, \ell)$ is the convex hull of the valid d vectors. That is:

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

Note that $P(n, \ell)$ is the convex hull of $n!/2$ points, since each distance vector corresponds to two layouts (due to symmetry).

We remark that cutting planes involving distance variables have been derived in the past for MinLA, e.g., [9,24].

Note that $P(n, \ell)$ is not an *integral* polytope, since the distances d_{ij} need not be integral (as Fig. 1 illustrates). It is possible to obtain an integral polytope by replacing each variable d_{ij} with $d_{ij} - (\ell_i + \ell_j)/2$. The resulting polytope is just a translation of ours.

3.2 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:

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

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

The following theorem states that the Eq. (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. □

3.3 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 $P(n, \ell)$:*

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

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 $d_{ij} \geq (\ell_i + \ell_j)/2$.

3.4 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 and Laurent [14]).

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 $d^\pi \in \text{CC}_n$. By symmetry, it suffices to prove the result for the identity layout. For all $1 \leq i < j \leq n$, we have:

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

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 Sects. 4.2 and 4.3.

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 Sect. 4.4.

3.5 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.

4 Valid inequalities and facets

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

4.1 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:

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

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 $N\backslash S$ instead, we have:

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

Next, for any $p \in N\backslash S$, 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:

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

Together with (10) this implies:

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

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

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

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

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

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 $N\backslash S$. But it is obvious that the left-hand side of the clique inequality on $N\backslash S$ 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.

4.2 Hypermetric inequalities

In Sect. 3.4, 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 [14] 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 Sect. 4.4) 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:

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

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.

4.3 Strengthened pure negative-type (SPN) inequalities

It is known [14] 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 $T \subset N \backslash S$ with $|S| = |T|$, the following 'strengthened pure negative-type' (SPN) inequality is valid for $P(n, \ell)$ and induces a non-empty face:*

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

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:

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

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 $s_1, \dots, s_{|S|}$ of the facilities in S and a numbering $t_1, \dots, t_{|T|}$ of the facilities in T such that, for $i = 1, \dots, |S|$, 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:

$$l_q \alpha_{ip} = l_p \alpha_{iq} \ (\forall i \in S, \{p, q\} \subset N \backslash (S \cup T))$$
$$l_q \alpha_{ip} = l_p \alpha_{iq} \ (\forall i \in T, \{p, q\} \subset N \backslash (S \cup T)).$$

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:

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

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.

4.4 Rounded positive semidefinite inequalities

At the end of Sect. 3.4, 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 [14]:

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

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., [8,14]).

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.

4.5 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 [23]. 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). □

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:

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

The validity of these inequalities for MinLA was shown in [24].

5 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 Sect. 5.1, branching in Sect. 5.2, the primal heuristic in Sect. 5.3, and other minor considerations in Subsection 5.4.

5.1 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 [18]. 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.

5.1.1 Clique inequalities

We conjecture that the separation problem for the clique inequalities (6) is *NP*-hard. Therefore, we devised a heuristic. The heuristic works by taking a set S corresponding to 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 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 (<0.1).

If this idea is implemented in a naive way, it takes $\mathcal{O}(n^3)$ time per source inequality. One can obtain a much faster implementation using the following observations. For a given S, define for all $j \notin S$ the quantity $R(j) = \sum_{i \in S} \ell_i d^*_{ij}$. Then, the increase in the slack of the clique inequality that would result if we added j to S is equal to:

$$\delta(j) = \ell_j R(j) + \frac{1}{6}\left((\ell(S) + \ell_j)^3 - \ell(S)^3 + \ell_j^3\right).$$

Note that, once $\ell(S)$ and the $R(j)$ have been computed, one can compute $\delta(j)$ (for fixed $j \notin S$) in constant time. Moreover, updating the $R(j)$ after a facility has been inserted into S can be done in constant time (again, for fixed $j \notin S$).

The implementation is as follows:

Let S be given. Compute $\ell(S)$.
Let `slk` denote the slack of the clique inequality corresponding to S.
For each j not in S:
  Compute $R(j)$.
Repeat:
  For each j not in S:
    Compute $\delta(j)$.
    If `slk` $+\delta(j) < 0$, output the violated clique inequality for $S \cup \{j\}$.
  If any clique inequalities have been output, stop.
  Let $k = \arg\min_{j \notin S} \delta(j)$.
  Add k to S and set `slk` := `slk`$+\delta(k)$.
  Add ℓ_k to $\ell(S)$.
  For each j not in S:
    Set $R(j) := R(j) + \ell_k d^*_{jk}$.
Until $|S| = n - 1$.

This algorithm runs in $O(n^2)$ time, and can generate up to n violated inequalities in a single call.

5.1.2 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.

5.1.3 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. Using a similar argument to the one we used for the clique inequalities, 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.

5.1.4 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 $j \in N\backslash\{i\}$, be a 0–1 variable taking the value 1 if and only if $j \in S$. Clearly, finding the set $S \subset N\backslash\{i\}$ that maximises the violation of the weak star inequality amounts to minimising:

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

This is an unconstrained quadratic program in the binary variables y_j, with non-positive quadratic terms. It is well-known (e.g., Picard and Ratliff [28]) 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 $\gamma(S)$, form the corresponding star inequality, and check it for violation. All violated inequalities found (if any) are added to the LP.

5.2 Branching rule

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 [1,2,20,25]. Here, however, we have decided to avoid the use of additional variables, and use a specialised branching rule to achieve feasibility.

After some experimentation, we decided to use a branching scheme in which each node in the branch-and-bound tree represents an ordering of a subset of variables. (A similar scheme was used in [10] to solve the so-called *linear ordering problem*.) 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 tree is represented by a permutation of $\{1, \ldots, p+2\}$. At such a 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.

5.3 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 led us to use a statistical technique called multi-dimensional scaling (MDS). Specifically, we use the classical MDS procedure of Torgerson [32], which is extremely fast in practice. It produces a placement of the centroids in the real line, but the 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.

5.4 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 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.)

6 Computational experiments

The branch-and-cut algorithm was coded in Microsoft Visual C and run on a 1.7 GHz Pentium Dual Core PC, with 2GB of RAM, under Windows XP. We called two simplex routines from the CPLEX 12.1 callable library: primal simplex to solve the initial LP relaxation and dual simplex to re-optimise after cutting planes were added. Due to our specialised branching rule, however, we used our own branch-and-bound shell.

6.1 Seventeen 'classical' instances

We began by testing the branch-and-cut algorithm on 17 'classical' instances from the literature (see Table 1). The seven 'S' instances are due to Simmons [29]. The

Table 1 Cutting-plane results for the seventeen 'classical' instances

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	2,324.5	**2,324.5**	**2,324.5**	0.00	0.00	17	0.101
S9	2,469.5	**2,469.5**	**2,469.5**	0.00	0.00	15	0.126
S9H	4,695.5	4,664.2	**4,695.5**	0.67	0.00	26	0.242
S10	2,781.5	2,778.2	**2,781.5**	0.12	0.00	25	0.327
S11	6,933.5	6,886.8	**6,933.5**	0.67	0.00	34	0.483
H20	15,549.0	15,174.6	**15,549.0**	2.41	0.00	91	8.752
H30	44,965.0	44,136.7	45,158.0	1.84	0.43	127	58.968
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 Branch-and-cut results for the seventeen 'classical' instances

Inst.	Nodes	Time (s)	Inst.	Nodes	Time (s)
S5	1	0.061	C5	1	0.047
S8	13	0.466	C6	1	0.062
S8H	1	0.114	C7	19	0.266
S9	1	0.135	C8	4	0.141
S9H	50	2.376	C12	63	3.978
S10	4	0.414	C15	144	9.594
S11	4	0.674	C20	865	312.45
H20	251	142.13	C30	28,158	64,183
H30	131,885	101,269	–	–	–

Table 3 Cutting-plane results for the instances of Anjos and Vannelli

Inst.	Opt.	LB	UB	%gap1	%gap2	Iter.	Time (s)
N25-1	4,618.0	4,534.4	**4,618.0**	1.81	0.00	112	32.744
N25-2	37,166.5	35,869.6	37,449.5	3.49	0.76	126	29.094
N25-3	24,301.0	23,653.0	24,466.0	2.67	0.68	141	33.010
N25-4	48,291.5	46,681.6	48,537.5	3.33	0.51	137	34.991
N25-5	15,623.0	15,107.4	15,725.0	3.30	0.65	186	47.206
N30-1	8,247.0	8,134.6	8,267.0	1.36	0.24	143	105.020
N30-2	21,582.5	21,226.8	21,754.5	1.65	0.80	139	77.891
N30-3	45,449.0	44,239.8	45,522.0	2.66	0.16	174	86.019
N30-4	56,873.5	56,000.4	56,904.5	1.54	0.05	167	88.936
N30-5	115,268.0	113,039.0	115,304.0	1.93	0.03	230	116.064

two 'H' instances were derived by Heragu and Kusiak [19], by modifying the famous instances of the quadratic assignment problem (QAP) due to Nugent et al. [26]. The remaining eight instances are also due to Heragu and Kusiak [19], but we label them 'C', since they have a so-called 'clearance requirement' (see [19]).

Optimal solutions were first computed for the 'S' instances by Amaral [2], and for the remaining instances by Anjos and Vannelli [6].

Table 1 shows the results obtained at the root node, i.e., using only cutting planes and the multi-dimensional scaling heuristic. The first two columns show the instance name and optimal cost. The next two columns show the lower and upper bounds. 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 up to that point (in seconds).

It can be seen that the gap between the lower bound and the optimum is below 3 % in all cases, and in most cases it is much smaller. Interestingly, both the lower bounds and running times are very similar to the ones obtained in [4,7] using SDP. The lower bounds reported in [3,6,21] are better, but at the expense of much larger running times.

Table 4 Branch-and-cut results for the instances of Anjos and Vannelli

Inst.	Nodes	Time (s)	Inst.	Nodes	Time (s)
N25-1	27,619	26,384	N30-1	61,716	122,451
N25-2	1,640	2,315.4	N30-2	7,397	14,213
N25-3	7,207	5,141.2	N30-3	25,508	47,292
N25-4	2,099	2,373.9	N30-4	2,054	3,500.3
N25-5	2,860	4,689.5	N30-5	18,188	47,031

Table 5 Cutting-plane results for instances with $36 \leq n \leq 56$

Inst.	LB	UB	%gap	Iter.	Time (s)
ste36-1	10,043.1	10,287.0	2.37	306	1,997.2
ste36-2	174,582.4	181,508.0	3.82	278	1,261.2
ste36-3	98,935.9	101,643.5	2.66	1,049	22,692.3
ste36-4	95,406.8	95,805.5	0.42	342	1,361.9
ste36-5	89,397.0	91,659.5	2.47	481	2,129.5
sko42-1	24,955.7	25,531.0	2.25	129	1,152.8
sko42-2	210,472.8	216,154.5	2.63	242	2,132.8
sko42-3	164,929.2	173,267.5	4.81	203	1,993.5
sko42-4	132,766.0	137,626.0	3.53	271	2,687.0
sko42-5	241,865.1	248,238.5	2.57	254	1,991.0
sko49-1	40,249.7	40,981.0	1.78	184	3,433.8
sko49-2	405,881.5	418,824.0	3.09	302	7,724.6
sko49-3	316,238.0	325,224.0	2.76	373	9,274.7
sko49-4	231,199.9	236,791.5	2.36	263	5,014.0
sko49-5	652,924.0	666,456.0	2.03	299	6,029.2
sko56-1	63,164.7	64,063.0	1.40	1,100	72,886.0
sko56-2	479,910.5	496,814.0	3.40	1318	1,31,604.3
sko56-3	164,143.1	170,478.0	3.72	648	30,708.2
sko56-4	297,904.9	313,495.0	4.97	845	47,017.9
sko56-5	574,721.9	592,299.5	2.97	1,340	92,678.8

We also observe that the multi-dimensional scaling heuristic gives remarkably tight upper bounds. Indeed, the maximum gap for these instances is only 0.43 %, and the heuristic solution is optimal in 13 cases out of 17.

Table 2 shows the results obtained with branch-and-cut. For each instance we show the instance name, the number of branch-and-cut nodes and the number of seconds taken to solve the instance to proven optimality. The smaller instances are very easy to solve, but the instances with $n = 30$ are quite challenging.

6.2 Ten instances of Anjos and Vannelli

We then tested the algorithm on 10 newer instances created by Anjos and Vannelli [6], which were again based on the Nugent et al. QAP instances. We call these 'N'

Table 6 Cutting-plane results for $64 \leq n \leq 100$ and 1-day time limit

Inst.	LB	UB	%gap	Iter.
sko64-1	94,954.3	96,930.0	2.04	714
sko64-2	618,415.3	634,332.5	2.51	500
sko64-3	401,179.6	414,356.5	3.18	728
sko64-4	284,273.2	297,358.0	4.40	573
sko64-5	476,921.8	501,922.5	4.98	662
sko72-1	137,249.2	139,174.0	1.38	312
sko72-2	676,154.2	712,261.0	5.07	294
sko72-3	1,015,417.3	1,054,184.5	3.68	425
sko72-4	887,747.2	920,693.5	3.58	415
sko72-5	411,734.0	428,305.5	3.87	347
sko81-1	200,417.2	205,475.0	2.46	147
sko81-2	496,014.0	523,021.5	5.16	163
sko81-3	913,458.3	970,920.0	5.92	141
sko81-4	1,926,464.7	2,032,634.0	5.22	145
sko81-5	1,214,999.8	1,303,756.0	6.81	158
sko100-1	359,142.7	378,584.0	5.14	36
sko100-2	1,948,288.5	2,076,714.5	6.18	37
sko100-3	14,919,729.8	16,177,226.5	7.77	37
sko100-4	3,002,469.7	3,237,111.0	7.25	40
sko100-5	965,656.8	1,034,922.5	6.69	39

instances in Tables 3 and 4. The optimal solutions for these instances were again presented in [6].

The bounds are of slightly poorer quality here, but the heuristic still produces very good upper bounds at the root node. Moreover, the branch-and-cut algorithm is capable of solving all of the instances. Once again, these results are comparable to those in [7], but not as good as those in [3,6,21].

6.3 Forty instances created by Anjos and Yen

Next, we tested the algorithm on 40 larger instances created by Anjos and Yen [7]. These instances have $n \in \{36, 42, 49, 56, 64, 72, 81, 100\}$. For most of them, the optimal solution value is not known. Since solving the LP at the root node is already a challenge for these instances, we report in Tables 5 and 6 only the results obtained at the root node. Moreover, for the instances in Table 6, we imposed a time limit of 1 day to avoid excessive running times.

For the instances in Table 5, the running times and bounds are comparable to those in [7], but not as good as those in [21]. For the instances in Table 6, the bounds are poor, which seems to be due to the time limit.

7 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.

There are several possible avenues for further research. First, one could search for additional facet-inducing inequalities and separation algorithms. Second, one could try to somehow incorporate our cutting planes into the LP-based procedures in [1–3], or the SDP-based procedures in [4,6,7,21]. Third, one could apply our methodology to other 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.

Acknowledgments The first author was supported by CNPq-Brazil Grant `500212/02-3` and in part by the FCT grant 'Investigation of efficient techniques for hard optimization problems in Operations and Logistics". 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 some useful advice.

References

1. Amaral, A.R.S.: On the exact solution of a facility layout problem. Eur. J. Oper. 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. Discret. 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. Discret. Optim. **2**, 113–122 (2005)
5. Anjos, M.F., Liers, F.: Global approaches for facility layout and VLSI floorplanning. To appear in. In: Anjos, M.F., Lasserre, J.B. (eds.) Handbook on Semidefinite, Cone and Polynomial Optimization, Springer, Berlin (2005)
6. 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)
7. Anjos, M.F., Yen, G.: Provably near-optimal solutions for very large single-row facility layout problems. Optim. Methods Softw. **24**, 805–817 (2009)
8. Avis, D., Umemoto, J.: Stronger linear programming relaxations for max-cut. Math. Program. **97**, 451–469 (2003)
9. Caprara, A., Letchford, A.N., Salazar, J.J.: Decorous lower bounds for minimum linear arrangement. INFORMS J. Comput. **23**, 26–40 (2011)
10. de Cani, J.S.: A branch and bound algorithm for maximum likelihood paired comparison ranking. Biometrika **59**, 131–135 (1972)
11. Conforti, M., Cornuéjols, G., Zambelli, G. et al.: Polyhedral approaches to mixed-integer linear programming. In: Jünger, M. (ed.) 50 Years of Integer Programming: 1958–2008, pp. 343–386. Springer, Berlin (2010)
12. Cook, W.J.: Fifty-plus years of combinatorial integer programming. In: Jünger, M. et al. (ed.) 50 Years of Integer Programming: 1958–2008, pp. 387–430. Springer, Berlin (2010)
13. Datta, D., Amaral, A.R.S., Figueira, J.R.: Single row facility layout problem using a permutation-based genetic algorithm. Eur. J. Oper. Res. **213**, 388–394 (2011)
14. Deza, M.M., Laurent, M.: Geometry of Cuts and Metrics. Springer, New York (1997)

15. Díaz, J., Petit, J., Serna, M.: A survey of graph layout problems. ACM Comput Surv. **34**, 313–356 (2002)
16. Djellab, H., Gourgand, M.: A new heuristic procedure for the single-row facility layout problem. Int. J. Comput. Integr. Manuf. **14**, 270–280 (2001)
17. Garey, M.R., Johnson, D.S., Stockmeyer, L.: Some simplified *NP*-complete graph problems. Theor. Comput. Sci. **1**, 237–267 (1976)
18. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Springer, New York (1988)
19. Heragu, S.S., Kusiak, A.: Machine layout problem in flexible manufacturing systems. Oper. Res. **36**, 258–268 (1988)
20. Heragu, S.S., Kusiak, A.: Efficient models for the facility layout problem. Eur. J. Oper. Res. **53**, 1–13 (1991)
21. Hungerländer, P., Rendl, F.: A computational study for the single-row facility layout problem. Technical Report, Alpen-Adria-Universität Klagenfurt (2011)
22. Karp, R.M., Held, M.: Finite-state processes and dynamic programming. SIAM J. Appl. Math. **15**, 693–718 (1967)
23. Laurent, M., Poljak, S.: Gap inequalities for the cut polytope. SIAM J. Matrix Anal. **17**, 530–547 (1996)
24. Liu, W., Vannelli, A.: Generating lower bounds for the linear arrangement problem. Discret. Appl. Math. **59**, 137–151 (1995)
25. Love, R.F., Wong, J.Y.: On solving a one-dimensional allocation problem with integer programming. INFOR. **14**, 139–143 (1976)
26. 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)
27. Picard, J.-C., Queyranne, M.: On the one-dimensional space allocation problem. Oper. Res. **29**, 371–391 (1981)
28. Picard, J.-C., Ratliff, H.: Minimum cuts and related problems. Networks **5**, 357–370 (1975)
29. Simmons, D.M.: One-dimensional space allocation: an ordering algorithm. Oper. Res. **17**, 812–826 (1969)
30. Simmons, D.M.: A further note on one-dimensional space allocation. Oper. Res. **19**, 249 (1971)
31. Solimanpur, M., Vrat, P., Shankar, R.: An ant algorithm for the single row layout problem in flexible manufacturing systems. Comput. Oper. Res. **32**, 583–598 (2005)
32. Torgerson, W.S.: Multidimensional scaling I: theory and method. Psychometrika **17**, 401–409 (1952)