Provably Near-Optimal Solutions for Very Large Single-Row Facility Layout Problems

Miguel F. Anjos* Ginger Yen†

May 13, 2008

(Revised January 4, 2009; March 14, 2009)

Abstract

The facility layout problem is a global optimization problem that seeks to arrange a given number of rectangular facilities so as to minimize the total cost associated with the (known or projected) interactions between them. This paper is concerned with the single-row facility layout problem (SRFLP), the one-dimensional version of facility layout that is also known as the one-dimensional space allocation problem. It was recently shown that the combination of a semidefinite programming (SDP) relaxation with cutting planes is able to compute globally optimal layouts for SRFLPs with up to 30 facilities. This paper further explores the application of SDP to this problem. First, we revisit the recently proposed quadratic formulation of this problem that underlies the SDP relaxation and provide an independent proof that the feasible set of the formulation is a precise representation of the set of all permutations on n objects. This fact follows from earlier work of Murata et al., but a proof in terms of the variables and structure of the SDP construction provides interesting insights on our approach. Second, we propose a new matrix-based formulation that yields a new SDP relaxation with fewer linear constraints but still yielding high-quality global lower bounds. Using this new relaxation, we are able to compute nearly-optimal solutions for instances with up to 100 facilities.

Key words: Single-row Facility Layout, Space Allocation, Combinatorial Optimization, Semidefinite Optimization, Global Optimization.

1 Introduction

The single-row facility layout problem (SRFLP) is concerned with the arrangement of a given number of rectangular facilities next to each other along a line so as to minimize the total weighted sum of the center-to-center distances between all pairs of facilities. This problem is a special case of the unequal-area facility layout problem, and is also known in the literature as the one-dimensional space allocation problem; see, e.g., [29]. An instance of the SRFLP consists of n one-dimensional

*Department of Management Sciences, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada. Research partially supported by grants 312125 and 314668 from the Natural Sciences and Engineering Research Council of Canada. Email anjos@stanfordalumni.org

†Department of Management Sciences, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada. Research partially supported by an Ontario Graduate Scholarship in Science and Technology, and by a TD Bank Graduate Scholarship in the Environment. Email tyen@engmail.uwaterloo.ca

facilities with given positive lengths $\ell_1, \ldots, \ell_n$, and pairwise weights c_{ij}. The objective is to arrange the facilities so as to minimize the total weighted sum of the center-to-center distances between all pairs of facilities. If all the facilities have the same length, the SRFLP becomes an instance of the linear arrangement problem, see, e.g., [25], which is itself a special case of the quadratic assignment problem; see, e.g., [10]. Several practical applications of the SRFLP have been identified in the literature, such as the arrangement of departments on one side of a corridor in supermarkets, hospitals, or offices [31], the assignment of disk cylinders to files [29], the assignment of airplanes to gates in an airport terminal [33], and the arrangement of machines in flexible manufacturing systems, where machines within manufacturing cells are often placed along a straight path travelled by an automated guided vehicle [20]. We refer the reader to the book of Heragu [18] for more information.

Several heuristic algorithms for the SRFLP have also been proposed. We point out the early work of Hall [14], the application of non-linear optimization methods by Heragu and Kusiak [21], the simulated annealing algorithms proposed independently by Romero and Sánchez-Flores [30] and Heragu and Alfa [19], a greedy heuristic algorithm proposed by Kumar et al. [23], and the use of both simulated annealing and tabu search by de Alvarenga et al. [11]. However, these heuristic algorithms do not provide a guarantee of global optimality, or an estimate of the distance from optimality.

Simmons [31] was the first to state and study the SRFLP, and proposed a branch-and-bound algorithm. Simmons [32] mentioned the possibility of extending the dynamic programming algorithm of Karp and Held [22] to the SRFLP, which was done by Picard and Queyranne [29]. Different mixed integer linear programming models were also proposed [1, 3, 21, 26], however they have very high computational time and memory requirements, and are unlikely to be effective for problems with more than about 20 facilities. Significant progress was achieved recently through the first ever polyhedral study of the SRFLP carried out by Amaral and Letchford [4]. Anjos and Vannelli [7] considered the semidefinite programming relaxation in [6] and tightened it using triangle inequalities as cutting planes. Their method routinely found optimal solutions for SRFLP instances with up to 30 facilities (although requiring several dozen hours on a 2.0GHz Dual Opteron with 16Gb of RAM). In this study they were able to provide optimal solutions for SRFLPs that had remained unsolved since 1988. At the time of writing, the current state-of-the-art in terms of globally optimal methods is the cutting-plane algorithm of Amaral [2] which found optimal solutions for all SRFLP instances tested in [7] but which performed much faster than the method in [7]. Optimal solutions for new instances with up to 35 facilities were also computed by Amaral [2].

Semidefinite programming (SDP) refers to the class of optimization problems where a linear function of a symmetric matrix variable X is optimized subject to linear constraints on the elements of X and the additional constraint that X must be positive semidefinite. This includes linear programming problems as the special case of a diagonal matrix variable. A variety of algorithms for solving SDP problems, including polynomial-time interior-point algorithms, have been implemented and benchmarked, and several excellent solvers for SDP are now available. We refer the reader to the SDP webpage [15] and the books [12, 35] for a thorough coverage of the theory and algorithms in this area, as well as of several application areas where SDP researchers have made significant contributions. In particular, SDP has been very successfully applied to problems which, like the SRFLP, have a strong combinatorial flavour. Recent survey papers on the application of SDP to combinatorial optimization include [5, 8, 24].

In this paper, we put the SDP-based formulation of the SRFLP on firmer theoretical ground, and we extend the size of the instances that can be handled by the SDP approach. First, we provide an independent proof that the feasible set of the quadratic formulation underlying the SDP relaxations is a precise representation of the set of all permutations on n objects. Specifically, we prove that the mapping used in the heuristic to extract a feasible solution to the SRFLP from the optimal solution of the SDP relaxation is a bijection between the feasible set of the quadratic formulation and the set of all permutations on n objects. This fact follows from the earlier work of Murata et al. [28], but a proof in terms of the variables and structure used here still provides interesting insights on the SDP approach.

Second, we demonstrate the ability of the SDP-based approach to tackle very large instances with up to 100 facilities. This is motivated by the fact that the results in [6] for instances with up to 80 facilities were obtained using the spectral bundle solver SB [16, 17], and while this solver is able to handle very large SDP problems, a major drawback is that its convergence slows down significantly after several hours of computation. As a consequence, instances with 80 facilities were the largest for which bounds could be obtained in reasonable time in [6], and some of the optimality gaps were greater than 10%. Our contribution is the use of a new matrix-based formulation of the SRFLP that yields an SDP relaxation that can be solved using the interior-point method solver CSDP. This simultaneously provides the ability to tackle problems with up to 100 facilities that were previously unattainable, and significantly improves the quality of the global lower bounds and of the layouts.

The paper is structured as follows. In Section 2, we prove the validity of the bijection between the feasible set of the quadratic formulation and the set of all permutations. In Section 3, we introduce the new matrix-based formulation of the SRFLP, and prove its validity by showing it is equivalent to the SDP-based formulation in [6]. In Section 4, we report preliminary computational results showing that using the SDP relaxation obtained from the new formulation, a primal-dual interior-point method and an acceptable amount of computational effort, we consistently obtain global lower bounds and feasible layouts with a global optimality gap of 5% or less for instances with up to 100 departments. Finally, some conclusions and current research efforts are summarized in Section 5.

2 Properties of a Quadratic ± 1 Formulation

Let $\pi = (\pi_1, \ldots, \pi_n)$ denote a permutation of the indices $[n] := \{1, 2, \ldots, n\}$ of the facilities, so that the leftmost facility is π_1, the facility to the right of it is π_2, and so on, with π_n being the last facility in the arrangement. Given a permutation π and two distinct facilities i and j, the center-to-center distance between i and j with respect to this permutation is $\frac{1}{2}\ell_i + D_\pi(i,j) + \frac{1}{2}\ell_j$, where $D_\pi(i,j)$ denotes the sum of the lengths of the facilities between i and j in the ordering defined by π. Solving the SRFLP consists of finding a permutation of the facilities that minimizes the weighted sum of the distances between all pairs of facilities. In other words, the problem is:

$$\min_{\pi \in \Pi_n} \sum_{i<j} c_{ij} \left[\frac{1}{2}\ell_i + D_\pi(i,j) + \frac{1}{2}\ell_j \right]$$

where Π_n denotes the set of all permutations of $[n]$. Since the lengths of the facilities are constant, it is clear that the crux of the problem is to minimize $\sum\limits_{i<j} c_{ij} D_\pi(i,j)$ over all permutations $\pi \in \Pi_n$. It is also clear that one can exchange the left and right ends of the layout and obtain the same

objective value. Hence, it is possible to simplify the problem by considering only the permutations for which, say, facility 1 is on the left half of the arrangement. This type of symmetry-breaking strategy is important for reducing the computational requirements of most algorithms, including those based on linear programming or dynamic programming. One noteworthy aspect of the SDP-based approach is that it implicitly accounts for these symmetries, and thus does not require the use of additional explicit symmetry-breaking constraints.

The quadratic formulation of the SRFLP proposed in [6] is obtained as follows. For a given permutation π of $[n]$, for each pair of integers ij with $1 \leq i < j \leq n$, define a binary ± 1 variable such that

$$R_{ij} := \begin{cases} 1, & \text{if } i \text{ is to the right of } j \text{ in } \pi, \\ -1, & \text{if } i \text{ is to the left of } j \text{ in } \pi. \end{cases}$$

In this definition, the order of the subscripts matters, and $R_{ij} = -R_{ji}$.

Given a particular assignment of ± 1 values to the R_{ij} variables, if this assignment represents a permutation of $[n]$, then it must hold that if $R_{ij} = R_{jk}$ then $R_{ik} = R_{ij}$, a necessary transitivity condition that can be formulated as a set of quadratic constraints:

$$R_{ij}R_{jk} - R_{ij}R_{ik} - R_{ik}R_{jk} = -1 \text{ for all triples } 1 \leq i < j < k \leq n.$$

The objective function of the SRFLP can be expressed as

$$\sum_{k \neq i,j} \ell_k \left(\frac{1 - R_{ki}R_{kj}}{2} \right).$$

upon observing that $R_{ki}R_{kj} = -1$ if and only if facility k is between i and j. This resulting formulation of the SRFLP is:

$$\begin{array}{ll} \min & K - \sum_{i<j} \frac{c_{ij}}{2} \left[\sum_{k<i} \ell_k R_{ki} R_{kj} - \sum_{i<k<j} \ell_k R_{ik} R_{kj} + \sum_{k>j} \ell_k R_{ik} R_{jk} \right] \\ \text{s.t.} & \\ & R_{ij}R_{jk} - R_{ij}R_{ik} - R_{ik}R_{jk} = -1 \text{ for all triples } i < j < k \\ & R_{ij}^2 = 1 \text{ for all } i < j \end{array} \qquad (1)$$

where $K := \left(\sum_{i<j} \frac{c_{ij}}{2} \right) \left(\sum_{k=1}^{n} \ell_k \right)$. Note that if every R_{ij} variable is replaced by its negative, then there is no change whatsoever to the formulation. This is how our formulation, and the subsequent matrix-based formulations and corresponding SDP relaxations, implicitly account for the symmetry in the SRFLP.

Let $\rho \in \{\pm 1\}^{\binom{n}{2}}$ denote a particular assignment of values to the R_{ij} variables, and hence define the set:

$$\mathcal{R}_n := \left\{ \rho \in \{\pm 1\}^{\binom{n}{2}} | R_{ij}R_{jk} - R_{ij}R_{ik} - R_{ik}R_{jk} = -1 \text{ for all triples } 1 \leq i < j < k \leq n \right\}.$$

Our objective in this section is to explore in more detail than was done in [6] the relationship between $\mathcal{R}_n$ and Π_n. More specifically, we prove that the function $f : \mathcal{R}_n \to \Pi_n$ defined by

$$f(\rho) = (\pi_1, \ldots, \pi_n), \text{ where } \pi_k := \frac{P_k + n + 1}{2}$$

and

$$P_k := \sum_{j\neq k} R_{kj} = \sum_{j<k} -R_{jk} + \sum_{j>k} R_{kj} \qquad \text{for } k = 1, 2, \ldots, n \tag{2}$$

is a bijection. This fact follows from the earlier work of Murata et al. [28] who demonstrated that for every layout in two dimensions, there exists a finite representation via so-called sequence-pairs. This technique has been applied with great success in VLSI floorplanning, the original application in [28], and also in two-dimensional facility layout [36, 27]. Nonetheless, a proof in terms of the variables and structure used here still provides interesting insights on the SDP approach.

First we recall that it was shown in [6] that every element of $\mathcal{R}_n$ corresponds to a permutation of $[n]$.

Lemma 1 *[6] If $\rho \in \mathcal{R}_n$ then the values P_k defined in (2) are all distinct.*

Corollary 1 *[6] The function f is onto.*

Proof: If $\rho \in \mathcal{R}_n$, then all the P_k values are integer and belong to the set

$$\mathcal{P} := \{-(n-1), -(n-3), \ldots, n-3, n-1\}$$

which has exactly n elements. By Lemma 1, every element of $\mathcal{P}$ equals exactly one P_k, and hence $f(\rho)$ is a permutation of $[n]$. ■

We now want to prove that f is one-to-one. To this end, consider the system of n equations in $\binom{n}{2}$ ± 1 unknowns:

$$\sum_{j<k} -R_{jk} + \sum_{j>k} R_{kj} = \beta_k, k = 1, \ldots, n, \tag{3}$$

for given right-hand sides β_k such that $\{\beta_k : k = 1, \ldots, n\} = \mathcal{P}$.

Lemma 2 *The system (3) has a unique solution in R_{ij}.*

Proof: The proof is by induction. There are two base cases to prove, $n = 2$ and $n = 3$.

For $n = 2$, the system has the form

$$\begin{aligned} R_{1,2} &= \beta_1, \\ -R_{1,2} &= \beta_2 \end{aligned}$$

and since $\mathcal{P} = \{-1, 1\}$, $\beta_1 = -\beta_2$, and the solution is clearly unique.

For $n = 3$, the system has the form

$$\begin{aligned} R_{1,2} + R_{1,3} &= \beta_1, \\ -R_{1,2} + R_{2,3} &= \beta_2, \\ -R_{1,3} - R_{2,3} &= \beta_3, \end{aligned}$$

where $\mathcal{P} = \{-2, 0, 2\}$. Suppose first that $\beta_1 = -2$, $\beta_2 = 0$, and $\beta_3 = 2$. Then $R_{1,2} = -1$, $R_{1,3} = -1$, and $R_{2,3} = -1$ is the unique solution. The other cases are solved similarly.

Now consider the system (3) for $n \geq 4$, and let k_1 and k_2 be such that $\beta_{k_1} = -(n-1)$ and $\beta_{k_2} = n-1$. (Such k_1 and k_2 always exist.) Then for any solution of (3), it holds that

$$- R_{j,k_1} = -1 \text{ for } j < k_1, R_{k_1,j} = -1 \text{ for } j > k_1, \text{ and } - R_{j,k_2} = 1 \text{ for } j < k_2, R_{k_2,j} = 1 \text{ for } j > k_2. \quad (4)$$

Now consider the equations for every other $\beta_k \neq k_1, k_2$. If $k > k_1$ and $k > k_2$ then we can write it as

$$-R_{k_1,k} - R_{k_2,k} + \sum_{j<k, j\neq k_1,k_2} -R_{jk} + \sum_{j>k, j\neq k_1,k_2} R_{kj} = \beta_k$$

and since $R_{k_1,k} = -1$ and $R_{k_2,k} = 1$, the first two terms cancel each other.

Similarly, if $k_1 < k < k_2$ then we can write it as

$$-R_{k_1,k} + \sum_{j<k} -R_{jk} + R_{k,k_2} + \sum_{j>k} R_{kj} = \beta_k$$

and since $R_{k_1,j} = -1$ and $R_{k,k_2} = -1$, the two terms again cancel each other.

If $k_2 < k < k_1$ then we can write it as

$$-R_{k_2,k} + \sum_{j<k} -R_{jk} + R_{k,k_1} + \sum_{j>k} R_{kj} = \beta_k$$

and since $R_{k_2,k} = 1$ and $R_{k,k_1} = 1$, again we have cancellation.

Finally, if $k < k_1$ and $k < k_2$ then

$$\sum_{j<k} -R_{jk} + \sum_{j>k} R_{kj} + R_{k,k_1} + R_{k,k_2} = \beta_k$$

and since $R_{k,k_1} = 1$ and $R_{k,k_2} = -1$, we cancel again.

In summary, we have reduced to a system on $n-2$ equations in $\binom{n-2}{2}$ ± 1 unknowns of the same form.

By the inductive hypothesis, this reduced system has a unique solution. Using the values from (4), this solution can be extended to a solution to the original system, and this extension is uniquely defined. Hence, the original system has a unique solution in R_{ij}. ■

Corollary 2 *The function f is one-to-one.*

Proof: Applying Lemma 2, if $\rho^1, \rho^2 \in \mathcal{R}_n$ and $f(\rho^1) = f(\rho^2)$, then $\rho^1 = \rho^2$. ■

Corollaries 1 and 2 yield the desired result.

Theorem 1 *The function $f : \mathcal{R}_n \to \Pi_n$ is a bijection.*

We conclude by recalling that an explicit expression of $f^{-1} : \Pi_n \to \mathcal{R}_n$ was provided in [6]. Given $\pi = (\pi_1, \pi_2, \ldots, \pi_n) \in \Pi_n$, let

$$R_{\pi_p,\pi_q} = -1 \text{ for all } p < q.$$

(Note that $\pi_p > \pi_q$ may hold even if $p < q$, and if that is the case, then $R_{\pi_q,\pi_p} = 1$.) The corresponding interpretation of π is that π_1 is the leftmost element in the linear arrangement, with π_2 on its right, and so on, up to π_n being the rightmost element.

We have thus shown directly that the mapping (2) is a bijection between $\mathcal{R}_n$ and the set of all permutations on n objects. In the next section, we obtain matrix-based formulations and SDP relaxations using the quadratic formulation (1).

3 Matrix-Based Formulations and Semidefinite Programming Relaxations

Following [6], we obtain a formulation of the SRFLP in the space of real symmetric matrices by fixing an ordering of all pairs ij such that $i < j$, and defining the vector

$$v := (R_{p_1}, \ldots, R_{p_{\binom{n}{2}}})^T,$$

where p_k denotes the k^{th} pair ij in the ordering. Using v, we construct the rank-one matrix $X := vv^T$ whose rows and columns are indexed by pairs ij. By construction, $X_{p_i,p_j} = R_{p_i} R_{p_j}$ for any two pairs p_i, p_j, and therefore we can formulate the SRFLP as:

$$\begin{array}{ll} \min & K - \sum\limits_{i<j} \frac{c_{ij}}{2} \left[\sum\limits_{k<i} \ell_k X_{ki,kj} - \sum\limits_{i<k<j} \ell_k X_{ik,kj} + \sum\limits_{k>j} \ell_k X_{ik,jk} \right] \\ \text{s.t.} & \\ & X_{ij,jk} - X_{ij,ik} - X_{ik,jk} = -1 \text{ for all triples } i < j < k \\ & \operatorname{diag}(X) = e \\ & \operatorname{rank}(X) = 1 \\ & X \succeq 0 \end{array} \tag{5}$$

where $\operatorname{diag}(X)$ represents a vector containing the diagonal elements of X, e denotes the vector of all ones, and $X \succeq 0$ denotes that X is symmetric positive semidefinite.

Removing the rank constraint yields an SDP relaxation:

$$\begin{array}{ll} \min & K - \sum\limits_{i<j} \frac{c_{ij}}{2} \left[\sum\limits_{k<i} \ell_k X_{ki,kj} - \sum\limits_{i<k<j} \ell_k X_{ik,kj} + \sum\limits_{k>j} \ell_k X_{ik,jk} \right] \\ \text{s.t.} & \\ & X_{ij,jk} - X_{ij,ik} - X_{ik,jk} = -1 \text{ for all triples } i < j < k \\ & \operatorname{diag}(X) = e \\ & X \succeq 0 \end{array} \tag{6}$$

Note that in general the SDP problem only provides a lower bound on the optimal value of the SRFLP, and not a feasible solution, unless the optimal matrix X^* happens to have rank equal to one. A standard way to tighten linear or semidefinite relaxations of integer optimization problems is to add inequalities (such as the triangle inequalities) that are valid for the integer feasible points. There are several classes of such inequalities that can be considered, see e.g. [13]. This SDP relaxation, used with a simple scheme to add violated triangle inequalities as cutting planes, was used in [7] to solve SRFLPs with up to 30 facilities to global optimality. Its main limitation from a computational perspective is that it has $O(n^3)$ linear constraints; this limits the size of instances that can be tackled with it.

This limitation leads us to consider an alternative formulation, obtained by reducing the number of the linear constraints in the following way:

$$
\begin{array}{ll}
\min & K - \sum_{i<j} \frac{c_{ij}}{2} \left[\sum_{k<i} \ell_k X_{ki,kj} - \sum_{i<k<j} \ell_k X_{ik,kj} + \sum_{k>j} \ell_k X_{ik,jk} \right] \\
\text{s.t.} & \\
& \sum_{k \neq i,j,k=1}^{n} X_{ij,jk} - \sum_{k \neq i,j,k=1}^{n} X_{ij,ik} - \sum_{k \neq i,j,k=1}^{n} X_{ik,jk} = -(n-2) \text{ for all pairs } i<j \\
& \operatorname{diag}(X) = e \\
& \operatorname{rank}(X) = 1 \\
& X \succeq 0
\end{array} \tag{7}
$$

It is straightforward to prove that the two formulations are equivalent.

Theorem 2 *The feasible sets of (5) and (7) are equal.*

Proof: Rewrite the first constraint of (7) as

$$
\sum_{k \neq i,j,k=1}^{n} (X_{ij,jk} - X_{ij,ik} - X_{ik,jk}) = -(n-2) \text{ for all pairs } i<j
$$

Suppose X is feasible for (7). Then the constraints $\operatorname{diag}(X) = e$ and $\operatorname{rank}(X) = 1$ together imply that $X_{ij,k\ell} = \pm 1$ for all entries of X. Furthermore, $X \succeq 0$ implies that $X_{ij,jk} - X_{ij,ik} - X_{ik,jk} \geq -1$ for all distinct i, j, k. Hence,

$$
\sum_{k \neq i,j,k=1}^{n} (X_{ij,jk} - X_{ij,ik} - X_{ik,jk}) \geq -(n-2).
$$

Therefore, it is clear that each term $X_{ij,jk} - X_{ij,ik} - X_{ik,jk}$ must equal -1. This means X is feasible for (5).

It is then straightforward to show that X feasible for (5) is also feasible for (7). By summing up the first constraint in (5) over k from 1 to n for each pair $i<j$, we obtain the first constraint in (7). ■

Removing the rank-one constraint from (7) again yields an SDP relaxation:

$$
\begin{array}{ll}
\min & K - \sum_{i<j} \frac{c_{ij}}{2} \left[\sum_{k<i} \ell_k X_{ki,kj} - \sum_{i<k<j} \ell_k X_{ik,kj} + \sum_{k>j} \ell_k X_{ik,jk} \right] \\
\text{s.t.} & \\
& \sum_{k \neq i,j,k=1}^{n} X_{ij,jk} - \sum_{k \neq i,j,k=1}^{n} X_{ij,ik} - \sum_{k \neq i,j,k=1}^{n} X_{ik,jk} = -(n-2) \text{ for all pairs } i<j \\
& \operatorname{diag}(X) = e \\
& X \succeq 0
\end{array} \tag{8}
$$

Although the number of linear constraints in this relaxation is now $O(n^2)$, with a corresponding favourable impact on the computational time and memory requirements of a primal-dual interior-point algorithm, it turns out that the quality of the solution appears to deteriorate only slightly, as illustrated by the results in Section 4.1.

4 Computational Results

To perform these tests, we generated a number of new instances of the SRFLP by starting with the connectivity data from some of the well-known Nugent QAP Problems, and adding to them facility lengths. All the instances ending in "1" have all facility lengths equal to unity, while the lengths for the other instances were generated randomly.

These computational results were obtained on a 2.4GHz Quad Opteron with 16Gb of RAM. The SDP problems were solved using the interior-point solver CSDP (version 5.0) of [9] in conjunction with the ATLAS library of routines of [34].

4.1 Comparison of the Relaxations

First, we compare the two SDP relaxations for problems with 25 to 56 facilities. This is to provide a sense of the extent to which the global bounds are weakened by the reduction in the number of constraints. The results are reported in Table 1. We see that while the CPU times are significantly smaller for the new SDP, the resulting gaps between each SDP bound and the corresponding best layout obtained are still small, mostly between 3% to 7% (with only 1 exception out of 20 test instances). In particular, for the instances of size 56, the CPU time for the original SDP relaxation is about 3 times greater, while the average gap only decreases to 2.76% from 4.13%. Furthermore, if we compare the two lower bounds directly, the average relative gap between them is only 1.19%.

The CPU times also illustrate one of the major advantages of using a primal-dual interior-point method, namely the predictability of the computational effort required to solve the relaxations for a given instance size using a given SDP relaxation.

It is also interesting to compare the bounds from both (6) and (8) with the bounds from the LP-based approach in [4]. This is done in Table 2. The SDP bound (6) is of course always better than the SDP bound (8). Observe that the SDP bound (6) is always within $\pm$ 2% of the value of the LP bound, and is often slightly superior. The weaker SDP bound (8) is also always within $\pm$ 2% of the LP bound, but is more often inferior to it. Therefore, the LP bound is tightly sandwiched between the two SDP bounds. Note however that the LP bound is obtained by separating several classes of cutting planes and re-solving the LP relaxations, while the SDP bounds reported here make no use of cutting planes.

4.2 Results for Very Large Instances

Second, we show that using the new relaxation, and several hours of computing power, we are able to consistently obtain nearly-optimal solutions for instances with up to 100 facilities.

First we consider the 20 instances originally constructed in [6]. We report the results of solving the SDP bound (8) for these instances in Table 3. Note that the CPU times here are greater than those allowed for the SB solver in [6], but because we are using a primal-dual interior-point method (CSDP) as opposed to a bundle method (SB), we can actually solve the SDP problems to optimality in a fairly predictable amount of time, instead of choosing a cutoff time after some experimentation, as had to be done for SB in [6]. (If the time cutoff for SB is set too low, then the bounds obtained are useless. On the other hand, very limited benefit is obtained from a high cutoff because the convergence of SB slows down significantly after about 10 hours.)

Second, in contrast with the results in [6], the results in Table 4 demonstrate that the combination of the new SDP relaxation (8) and the CSDP solver makes it possible to simultaneously:

Instance	# of fac.	SDP Relaxation (6)				SDP Relaxation (8)				Gap between lower bounds
		Lower bound	CPU time (sec)	Best layout by SDP-based heuristic	Gap	Lower bound	CPU time (sec)	Best layout by SDP-based heuristic	Gap	
SRFLP-nug25-1	25	4515.0	42	4622.0	2.37%	4463.5	40	4626.0	3.64%	1.15%
SRFLP-nug25-2	25	36355.5	42	37641.5	3.54%	35960.5	42	37346.5	3.86%	1.10%
SRFLP-nug25-3	25	23691.0	42	24537.0	3.57%	23398.0	40	24609.0	5.18%	1.25%
SRFLP-nug25-4	25	47330.0	43	48887.5	3.29%	46798.5	41	48811.5	4.30%	1.14%
SRFLP-nug25-5	25	15304.5	42	15767.0	3.02%	15148.0	42	15783.0	4.19%	1.03%
SRFLP-nug30-1	30	8061.0	252	8305.0	3.03%	7975.5	130	8310.0	4.19%	1.07%
SRFLP-nug30-2	30	21188.5	170	21663.5	2.24%	20921.5	129	21672.5	3.59%	1.28%
SRFLP-nug30-3	30	44518.5	175	45712.0	2.68%	43986.0	135	45703.0	3.90%	1.21%
SRFLP-nug30-4	30	55947.5	174	56922.5	1.74%	55181.0	137	57060.5	3.41%	1.39%
SRFLP-nug30-5	30	113072.0	170	115776.0	2.39%	111828.5	129	115986.0	3.72%	1.11%
SRFLP-ste36-1	36	10087.5	897	10301.0	2.12%	9851.0	435	10328.0	4.84%	2.40%
SRFLP-ste36-2	36	175387.0	857	181910.0	3.72%	170759.5	432	182649.0	6.96%	2.71%
SRFLP-ste36-3	36	98739.0	822	102179.5	3.48%	96090.0	433	104041.5	8.28%	2.76%
SRFLP-ste36-4	36	94650.5	863	96080.5	1.51%	91103.0	436	96854.5	6.31%	3.89%
SRFLP-ste36-5	36	89533.0	865	91893.5	2.64%	87688.0	439	92563.5	5.56%	2.10%
SRFLP-sko42-1	42	24807.0	2721	25724.0	3.70%	24517.0	1207	25779.0	5.15%	1.18%
SRFLP-sko42-2	42	210785.0	2714	217296.5	3.09%	207357.0	1221	218117.5	5.19%	1.65%
SRFLP-sko42-3	42	169944.5	2852	173854.5	2.30%	167783.5	1210	174694.5	4.12%	1.29%
SRFLP-sko42-4	42	133429.5	2898	138829.0	4.05%	131536.0	1161	139630.0	6.15%	1.44%
SRFLP-sko42-5	42	242925.5	2741	249327.5	2.64%	238669.5	1218	250501.5	4.96%	1.78%
SRFLP-sko49-1	49	39794.5	10137	41308.0	3.80%	39333.5	3595	41379.0	5.20%	1.17%
SRFLP-sko49-2	49	407741.5	10232	418288.0	2.59%	403024.5	3810	418370.0	3.81%	1.17%
SRFLP-sko49-3	49	317628.0	10305	325747.0	2.56%	313923.5	3733	326004.0	3.85%	1.18%
SRFLP-sko49-4	49	232368.0	10240	237894.5	2.38%	229809.5	3934	238380.5	3.73%	1.11%
SRFLP-sko49-5	49	652638.0	10305	671508.0	2.89%	645406.5	3853	673303.0	4.32%	1.12%
SRFLP-sko56-1	56	62496.5	31240	64396.0	3.04%	61789.5	11119	64454.0	4.31%	1.14%
SRFLP-sko56-2	56	486426.5	32830	498836.0	2.55%	480473.5	11375	499700.0	4.00%	1.24%
SRFLP-sko56-3	56	166441.5	32270	171860.0	3.26%	164609.5	11296	171963.0	4.47%	1.11%
SRFLP-sko56-4	56	306550.5	32452	315175.0	2.81%	302572.5	10551	315803.0	4.37%	1.31%
SRFLP-sko56-5	56	582117.5	32273	594477.5	2.12%	575501.5	10580	595593.5	3.49%	1.15%

Table 1: Comparison of the Two SDP Relaxations

Instance	# of fac.	SDP Relaxation (6)		SDP Relaxation (8)		Lower bound from LP in [4]
		Lower bound	Gap from the LP bound (±)	Lower bound	Gap from the LP bound (±)	
SRFLP-nug25-1	25	4515.0	-0.36%	4463.5	-1.50%	4531.3
SRFLP-nug25-2	25	36355.5	+1.59%	35960.5	+0.49%	35785.8
SRFLP-nug25-3	25	23691.0	+0.48%	23398.0	-0.77%	23578.6
SRFLP-nug25-4	25	47330.0	+1.66%	46798.5	+0.52%	46555.8
SRFLP-nug25-5	25	15304.5	+1.48%	15148.0	+0.44%	15081.2
SRFLP-nug30-1	30	8061.0	-0.94%	7975.5	-1.99%	8137.3
SRFLP-nug30-2	30	21188.5	-0.01%	20921.5	-1.27%	21191.6
SRFLP-nug30-3	30	44518.5	+0.69%	43986.0	-0.51%	44212.4
SRFLP-nug30-4	30	55947.5	+0.01%	55181.0	-1.36%	55939.5
SRFLP-nug30-5	30	113072.0	+0.31%	111828.5	-0.80%	112727.8

Table 2: Comparison of the SDP and LP Bounds

1. increase the dimension of the largest instances considered to 100 facilities, and

2. decrease the percentage gap between the global lower bound and the best layout found.

5 Conclusions and Current Research

This paper extends the previous work in the literature on the application of semidefinite programming to the SRFLP. We proved the existence of an explicit bijection between the feasible set of the recently proposed quadratic formulation, and the set of all permutations. We also proposed a new matrix-based formulation that yields a new semidefinite programming relaxation with $O(n^2)$ linear constraints (instead of $O(n^3)$ for the original SDP relaxation), and used it to obtain nearly-optimal solutions for instances with up to 100 facilities with optimality gaps of consistently 5% or less.

Current research is looking at the potential of combining the new SDP relaxation (8) in combination with the cutting planes used in [7] to obtain global optimal solutions for instances larger than those solved in [7]. Promising preliminary results in this direction are presented in [37]. Another promising direction of research is the application to the SRFLP of the novel approach of Xie and Sahinidis [36] for the continuous layout problem.

Acknowledgments

The authors thank Brian Borchers for his assistance in installing the CSDP and ATLAS software used to obtain the computational results. Parts of this research were completed while the first author was a Visiting Fellow at Princeton University. He thanks Robert J. Vanderbei and all the members of the ORFE department for their hospitality. The authors also thank two anonymous referees for their comments that significantly improved this paper.

Instance	# of fac.	SDP lower bound	CPU time (sec)	Best layout by SDP-based heuristic	Gap
AKV-60-01	60	1473338.5	20353	1478464.0	0.35%
AKV-60-02	60	829965.5	18490	844695.0	1.78%
AKV-60-03	60	641723.0	17448	650533.5	1.38%
AKV-60-04	60	389733.0	17719	400669.0	2.81%
AKV-60-05	60	316284.5	18328	319103.0	0.89%
AKV-70-01	70	1513741.5	87930	1533075.0	1.28%
AKV-70-02	70	1424673.5	87639	1444720.0	1.41%
AKV-70-03	70	1503311.5	83507	1526830.5	1.56%
AKV-70-04	70	951725.0	82611	972389.0	2.17%
AKV-70-05	70	4207969.5	85367	4218730.5	0.26%
AKV-75-01	75	2377176.0	144912	2394812.5	0.74%
AKV-75-02	75	4294138.0	152600	4322967.0	0.67%
AKV-75-03	75	1230123.5	138459	1255634.0	2.07%
AKV-75-04	75	3911919.0	149269	3950444.5	0.99%
AKV-75-05	75	1763890.5	155398	1797676.0	1.92%
AKV-80-01	80	2045170.5	176849	2073453.5	1.38%
AKV-80-02	80	1903788.0	174708	1923506.0	1.04%
AKV-80-03	80	3237288.5	177751	3256577.0	0.60%
AKV-80-04	80	3730569.0	188203	3747950.0	0.47%
AKV-80-05	80	1555271.5	169384	1594228.0	2.51%

Table 3: Bounds for Very Large Instances from [6] Using the SDP Relaxation (8)

Instance	# of fac.	SDP lower bound	CPU time (sec)	Best layout by SDP-based heuristic	Gap
SRFLP-sko64-1	64	93388.0	29768	97842.0	4.77%
SRFLP-sko64-2	64	619258.0	30966	636602.5	2.80%
SRFLP-sko64-3	64	402165.5	31641	418083.5	3.96%
SRFLP-sko64-4	64	285762.5	31081	300469.0	5.15%
SRFLP-sko64-5	64	488035.0	31669	505185.5	3.51%
SRFLP-sko72-1	72	135280.5	73595	140209.0	3.64%
SRFLP-sko72-2	72	690377.0	71909	716873.0	3.84%
SRFLP-sko72-3	72	1026164.0	80365	1063314.5	3.62%
SRFLP-sko72-4	72	898586.5	73237	924542.5	2.89%
SRFLP-sko72-5	72	415320.5	73275	432062.5	4.03%
SRFLP-sko81-1	81	197416.5	171757	207229.0	4.97%
SRFLP-sko81-2	81	507726.0	176564	527239.5	3.84%
SRFLP-sko81-3	81	942850.5	171913	979816.0	3.92%
SRFLP-sko81-4	81	1971210.5	168481	2042462.0	3.62%
SRFLP-sko81-5	81	1267977.0	182549	1311605.0	3.44%
SRFLP-sko100-1	100	367048.5	773345	380981.0	3.80%
SRFLP-sko100-2	100	2024668.0	864788	2089757.5	3.21%
SRFLP-sko100-3	100	15750362.0	849831	16251391.5	3.18%
SRFLP-sko100-4	100	3148661.0	921191	3266569.0	3.74%
SRFLP-sko100-5	100	1002763.5	790405	1040987.5	3.81%

Table 4: Bounds for New Very Large Instances Using the SDP Relaxation (8)

References

[1] A.R.S. Amaral. On the exact solution of a facility layout problem. *Eur. J. Oper. Res.*, 173(2):508–518, 2006.

[2] A.R.S. Amaral. On the exact solution of a facility layout problem. *Discr. Appl. Math.*, 157(1):183–190, 2009.

[3] A.R.S. Amaral. An exact approach for the one-dimensional facility layout problem. *Oper. Res.*, to appear.

[4] A.R.S. Amaral and A.N. Letchford. A polyhedral approach to the single-row facility layout problem. Technical report, Department of Management Science, Lancaster University, March 2008.

[5] M.F. Anjos. Semidefinite optimization approaches for satisfiability and maximum-satisfiability problems. *J. on Satisfiability, Boolean Modeling and Computation*, 1(1):1–47, 2005.

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

[7] M.F. Anjos and A. Vannelli. Computing globally optimal solutions for single-row layout problems using semidefinite programming and cutting planes. *INFORMS J. Comp.*, 20(4):611–617, 2008.

[8] M.F. Anjos and H. Wolkowicz. Semidefinite programming for discrete optimization and matrix completion problems. *Discrete Appl. Math.*, 123(1–2):513–577, 2002.

[9] B. Borchers. CSDP, a C library for semidefinite programming. *Optim. Methods Softw.*, 11/12(1-4):613–623, 1999.

[10] E. Çela. *The Quadratic Assignment Problem*, volume 1 of *Combinatorial Optimization*. Kluwer Academic Publishers, Dordrecht, 1998.

[11] A.G. de Alvarenga, F.J. Negreiros-Gomes, and M. Mestria. Metaheuristic methods for a class of the facility layout problem. *J. Intell. Manuf.*, 11:421–430, 2000.

[12] E. de Klerk. *Aspects of Semidefinite Programming*, volume 65 of *Applied Optimization*. Kluwer Academic Publishers, Dordrecht, 2002.

[13] M.M. Deza and M. Laurent. *Geometry of Cuts and Metrics*, volume 15 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 1997.

[14] K.M. Hall. An r-dimensional quadratic placement algorithm. *Management Sciences*, 17:219–229, 1970.

[15] C. Helmberg. `http://www-user.tu-chemnitz.de/~helmberg/semidef.html`.

[16] C. Helmberg and K.C. Kiwiel. A spectral bundle method with bounds. *Math. Program.*, 93(2, Ser. A):173–194, 2002.

[17] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM J. Optim.*, 10(3):673–696 (electronic), 2000.

[18] S. S. Heragu. *Facilities Design.* iUniverse, second edition, 2006.

[19] S.S. Heragu and A.S. Alfa. Experimental analysis of simulated annealing based algorithms for the layout problem. *European J. Oper. Res.*, 57(2):190–202, 1992.

[20] S.S. Heragu and A. Kusiak. Machine layout problem in flexible manufacturing systems. *Oper. Res.*, 36(2):258–268, 1988.

[21] S.S. Heragu and A. Kusiak. Efficient models for the facility layout problem. *European J. Oper. Res.*, 53:1–13, 1991.

[22] Richard M. Karp and Michael Held. Finite-state processes and dynamic programming. *SIAM J. Appl. Math.*, 15:693–718, 1967.

[23] K.R. Kumar, G.C. Hadjinicola, and T. Lin. A heuristic procedure for the single-row facility layout problem. *European J. Oper. Res.*, 87(1):65–73, 1995.

[24] M. Laurent and F. Rendl. Semidefinite programming and integer programming. In K. Aardal, G. Nemhauser, and R. Weismantel, editors, *Handbook on Discrete Optimization*, pages 393–514. Elsevier, 2005.

[25] W. Liu and A. Vannelli. Generating lower bounds for the linear arrangement problem. *Discrete Appl. Math.*, 59(2):137–151, 1995.

[26] R.F. Love and J.Y. Wong. On solving a one-dimensional space allocation problem with integer programming. *INFOR*, 14(2):139–143, 1976.

[27] R.D. Meller, W. Chen, and H.D. Sherali. Applying the sequence-pair representation to optimal facility layout designs. *Oper. Res. Lett.*, 35(5):651–659, 2007.

[28] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani. VLSI module placement based on rectangle-packing by the sequence-pair. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 15(12):1518–1524, 1996.

[29] J.-C. Picard and M. Queyranne. On the one-dimensional space allocation problem. *Oper. Res.*, 29(2):371–391, 1981.

[30] D. Romero and A. Sánchez-Flores. Methods for the one-dimensional space allocation problem. *Comput. Oper. Res.*, 17(5):465–473, 1990.

[31] D.M. Simmons. One-dimensional space allocation: An ordering algorithm. *Oper. Res.*, 17:812–826, 1969.

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

[33] J.K. Suryanarayanan, B.L. Golden, and Q. Wang. A new heuristic for the linear placement problem. *Computers & Operations Research*, 18(3):255–262, 1991.

[34] R. Whaley, A. Petitet, and J. Dongarra. Automated empirical optimizations of software and the ATLAS project. *Parallel Computing*, 27(1-2):3–35, 2001.

[35] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming*. Kluwer Academic Publishers, Boston, MA, 2000.

[36] W. Xie and N.V. Sahinidis. A branch-and-bound algorithm for the continuous facility layout problem. *Comput. & Chem. Eng.*, 32:1016–1028, 2008.

[37] G. Yen. Cutting-plane separation strategies for semidefinite programming models to solve single-row facility layout problems. Master's thesis, University of Waterloo, 2008.