Semidefinite Optimization Approach for the Single-Row Layout Problem with Unequal Dimensions

Miguel F. Anjos

Operational Research Group
School of Mathematics
University of Southampton, U.K.

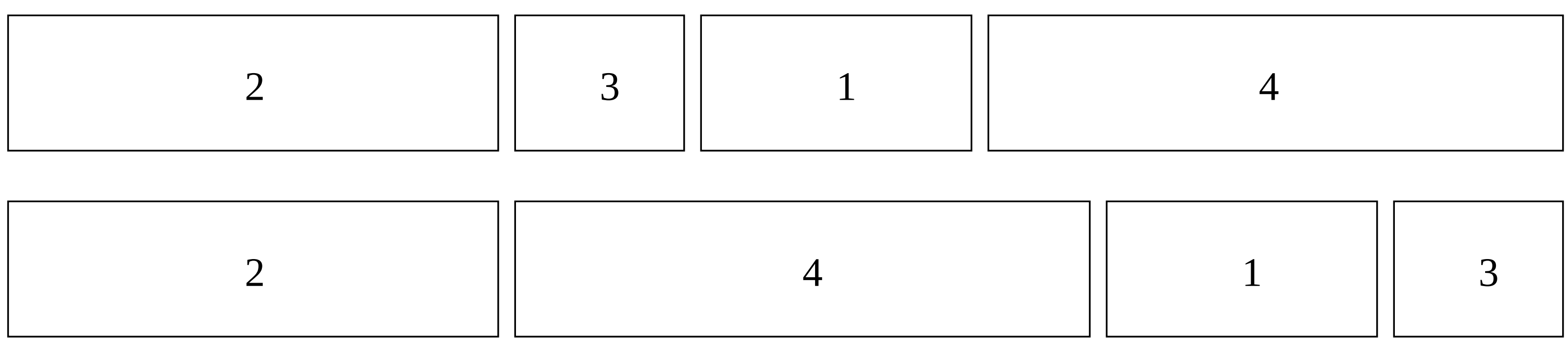
(joint on-going work with Andrew Kennings and Anthony Vannelli, University of Waterloo, Canada)

HPOPT 2004 – CWI – Amsterdam

25 June 2004

The Single-Row Facility Layout Problem

- The single-row facility layout problem, also known as the 1-D space allocation problem, consists in finding an optimal linear placement of facilities with varying dimensions on a straight line.
- Note that the distance between two facilities depends on the facilities placed between them. Consider the distance between 1 and 2 in this example:



The Single-Row Layout Problem (ctd)

An instance of the problem consists of:

- n 1-D facilities $\{r_1, \ldots, r_n\}$
- with positive lengths $\ell_1, \ldots, \ell_n$
- and (usually non-negative) pairwise connectivities c_{ij}.

We seek an arrangement of the facilities so as to minimize the total weighted sum of the center-to-center distances between all pairs of facilities.

We are thus optimizing over all permutations of the facilities.

(If all ℓ_i are equal, this is an instance of the Quadratic Assignment Problem.)

The Single-Row Layout Problem (ctd)

Some applications:

1. Assigning files to the cylinder of a disk
   (minimize average access time)
2. Layout of rooms along a corridor
   (minimize total travelled distance)
3. Standard-cell circuit design
   (minimize wirelength)

Earlier Work on Globally Optimal Algorithms

- Simmons (1969) proposed a branch-and-bound algorithm;
- Picard and Queyranne (1981) developed a DP algorithm, extending an algorithm of Karp and Held (1967) for the special case with all ℓ_i equal.

All these algorithms are guaranteed to find the global optimal solution, but they have very high computational times and memory requirements, and are unlikely to be effective for problems with 20 or more facilities.

Earlier Work on Locally Optimal Algorithms

- Heragu and Kusiak (1991) use non-linear optimization methods;
- Romero and Sánchez-Flores (1990) and Heragu and Alfa (1992) developed simulated annealing algorithms;
- Kumar et al. (1995) proposed a greedy heuristic algorithm.

The last two papers contain the best results in the literature so far, on problems with up to 30 facilities.

Our Contribution

We construct a semidefinite programming (SDP) relaxation providing a global lower bound on the optimal value of the problem.

- To the best of our knowledge, this is the first non-trivial global lower bound in the published literature.
- The structure of the relaxation suggests a simple heuristic procedure which extracts a facility arrangement from an optimal matrix solution to the SDP.

Our Contribution (ctd)

Therefore, the SDP-based approach yields both a feasible solution and guarantee of how far it is from global optimality.

When applied to problems previously considered in the literature, our lower bounds also provide a measure of the distance from optimality of the best layouts obtained using some of the aforementioned algorithms.

We successfully applied the SDP approach to randomly generated instances with up to 80 facilities.

Modelling the Problem

Let $\pi = (\pi_1, \ldots, \pi_n)$ denote a permutation of the indices $[n] := \{1, 2, \ldots, n\}$ of the facilities. Given a permutation π and two distinct facilities r_i and r_j, the center-to-center distance between r_i and r_j with respect to this permutation is

$$\frac{1}{2} l_i + D_\pi(i, j) + \frac{1}{2} l_j$$

where $D_\pi(i, j)$ denotes the sum of the lengths of the facilities between r_i and r_j in the linear arrangement defined by π.

The problem is then

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

where Π denotes the set of all permutations π of $[n]$.

Modelling the Problem (ctd)

Rewriting the objective function as

$$\min_{\pi \in \Pi} \sum_{i<j} c_{ij} D_\pi(i,j) + \sum_{i<j} \frac{1}{2} c_{ij}(l_i + l_j)$$

where the second summation is a constant independent of π, we see that the problem is really to minimize $\sum_{i<j} c_{ij} D_\pi(i,j)$ over all permutations π.

Note also that we can exchange the left and right ends of the layout and obtain the same objective value.

Most algorithms apply explicit symmetry-breaking strategies to reduce the computational cost.

The SDP approach implicitly considers these symmetries.

Construction of the SDP Relaxation

For each pair i, j of facilities, define the ± 1 variable

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

The order of the subscripts matters: $R_{ij} = -R_{ji}$.

To accurately formulate the problem, we must ensure that the variables represent a valid permutation. In particular, we require that

if i is right of j and j is right of k, then i is right of k.

We can formulate this condition as

$$\text{if } R_{ij} = R_{jk}, \text{ then } R_{ik} = R_{ij},$$

or equivalently

$$(R_{ij} + R_{jk})(R_{ik} - R_{ij}) = 0.$$

Construction of the SDP Relaxation (ctd)

Expanding

$$(R_{ij} + R_{jk})(R_{ik} - R_{ij}) = 0,$$

we obtain the quadratic constraint

$$R_{ki}R_{ij} - R_{ij}R_{kj} - R_{ki}R_{kj} = -1.$$

In principle, we have three such constraints for each triple (i, j, k), but it is easy to check that they are all equivalent to this single quadratic equation with $i < j < k$.

These $\binom{n}{3}$ constraints on the R_{ij} variables suffice to obtain precisely all possible permutations of the n facilities.

Construction of the SDP Relaxation (ctd)

Let

$$\mathcal{R} := \left\{\rho \in \{\pm 1\}^{\binom{n}{2}} : R_{ki}R_{ij} - R_{ij}R_{kj} - R_{ki}R_{kj} = -1 \ \forall\, i < j < k\right\}$$

and given $\rho \in \mathcal{R}$, consider

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

Clearly 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. A straightforward mapping of the elements of $\mathcal{P}$ onto $[n]$ is given by

$$p_k = \frac{P_k + n + 1}{2}$$

Construction of the SDP Relaxation (ctd)

Theorem 1 *If $\rho \in \mathcal{R}$ then the values P_k defined in (1) are all distinct.*

This implies that every element of $\mathcal{P}$ is represented by exactly one P_k, and hence that $(p_1, p_2, \ldots, p_n)$ is a permutation of $[n]$ representing ρ.

To express the objective function of the problem in terms of the variables R_{ij}, observe that the sum of the lengths of the facilities between i and j can be expressed as

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

SDP Relaxation

Defining a rank-one matrix variable X with $\binom{n}{2}$ rows and columns such that $X_{ij,kl} = R_{ij}R_{kl}$ for all pairs of facilities, we can formulate the problem as:

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

where $K = \left(\sum_{i<j} \frac{c_{ij}}{2} \right) \left(\sum_{k=1}^{n} \ell_k \right)$.

Omitting the rank constraint yields the SDP relaxation.

SDP-based Heuristic

Let X^* be the optimal solution to the SDP relaxation. If we set $R_{12} = +1$, then we can scan the first row of X^* and assign the value $X_{12,ij}$ to the variable R_{ij}, for every pair $(i, j) \neq (1, 2)$.

Using these values, we then generate the P_k values using equation (1). From these we get the corresponding p_k values, and sorting these we obtain a permutation of the facilities.

Note that if every R_{ij} variable is replaced by its negative, then there is no change to the SDP relaxation.

The choice of $R_{12} = +1$ is arbitrary, and simply breaks this symmetry.

Computational Results

The first set of results was obtained by applying the SDP approach to a set of six test problems from the literature, ranging in dimension from 8 to 30 facilities.

For these problems, we can solve the SDP relaxations to optimality using SDPT3 (version 3.2).

Instance	Number of facilities	SDP Bound	CPU time (seconds)
Lit-1	8	2324.5	0.5
Lit-2	10	2773.9	0.9
Lit-3	11	6846.6	1.0
Lit-4	11	6847.6	1.0
Lit-5	20	15285.9	24.3
Lit-6	30	43963.7	499.0

(Computations performed on a 2.4GHz Pentium IV with 1.5Gb of RAM)

Instance	SDP bound	Best layout from HA (1992)	Best layout from KHL (1995) 1 pass	Best layout from KHL (1995) ≥ 1 passes	Layout by SDP-based heuristic
Lit-1	2324.5	2324.5 (0%)	2324.5 (0%)	2324.5 (0%)	2324.5 (0%)
Lit-2	2773.9	2781.5 (0.27%)	2781.5 (0.27%)	2781.5 (0.27%)	2781.5 (0.27%)
Lit-3	6846.6	6933.5 (1.24%)	6953.5 (1.54%)	6953.5 (1.54%)	7083.5 (3.34%)
Lit-4	6847.6	6933.5 (1.24%)	7265.5 (5.75%)	7265.5 (5.75%)	7083.5 (3.33%)
Lit-5	15285.9	15602.0 (2.03%)	15971.0 (4.29%)	15549.0 (1.69%)	15804.0 (3.28%)
Lit-6	44107.8*	45111.0 (2.22%)	45308.5 (2.65%)	44466.5 (0.81%)	45605.0* (3.28%)

Observations

1. The SDP bounds are remarkably tight for these six instances. In fact, they prove the optimality of the layout for Lit-1 which was (apparently) not yet known.

2. The SDP bound can improve dramatically by using a single branch and solving two more SDPs. This was done for the 30-facility problem, and improved the bound from 43963.7 to 44107.8

3. The SDP-based heuristic is surprisingly competitive (see e.g. Lit-4), considering that its layouts are not (yet) improved by applying a local improvement procedure.

Results on Randomly Generated Instances

To explore the effectiveness of the SDP approach on larger instances, we randomly generated instances with up to 80 facilities, and we used SBmethod for these instances.

Number of facilities	Cutoff for SB method	Gap between SDP global bound and SDP heuristic layout (average over 5 instances)
60	5h	4.47%
70	7h	4.29%
75	10h	4.12%
80	10h	6.85%

Summary and Some Current Research

- This SDP approach provides the first non-trivial global lower bound for single-row layout in the published literature.
- A heuristic procedure extracts a feasible layout from the SDP optimal solution, and empirical evidence shows that it is consistently within a few percentage points of the global optimal solution.
- This construction can be extended to two-dimensional unequal-area facility layout (on-going work with P.L. Takouda and A. Vannelli).