Models and Algorithms for Fair Layout Optimization Problems

Albert E. Fernandes Muritiba[(1)], Manuel Iori[(2)],
Silvano Martello[(1)], Marcos J. Negreiros Gomes[(3)]

(1) DEIS, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy
albert.fernandes2@unibo.it, silvano.martello@unibo.it

(2) DISMI, University of Modena and Reggio Emilia, Via Amendola 2, 42100 Reggio Emilia, Italy
manuel.iori@unimore.it

(3) State University of Ceará, Av. Paranjana, 1700, Fortaleza–Ceará, 60.740-903, Brazil
negreiro@uece.br

Abstract

Given a non-convex two-dimensional area and identical rectangular stands, we consider the problem of placing the maximum number of stands in the area, by satisfying a number of operational constraints. We present linear programming models and show their total unimodularity. We then give computational results obtained by applying the models to the real-world case of the Beira Mar handcraft fair of Fortaleza (Brazil).

1 Introduction

This paper is motivated by a real-world optimization problem, concerning fair layout. Given an exhibition surface having irregular (non-convex) shape such as the one shown by the solid lines in Figure 1 (disregard by the moment the dotted lines), and an unlimited number of identical rectangular stands, we want to find a feasible layout containing the maximum number of such stands. A layout is feasible if it fulfils a number of basic operational constraints (the stands cannot overlap and must completely lie within the exhibition area; the clients must have an easy access to the stands), plus additional constraints coming from specific requests from the organizers.

In this paper we limit our search to a particular case of layout, quite common in practical contexts, in which the stands must all have the same orientation, and must be accommodated into vertical *strips*, with enough space between strips to allow an easy access of the clients. An example is shown in Figure 2.

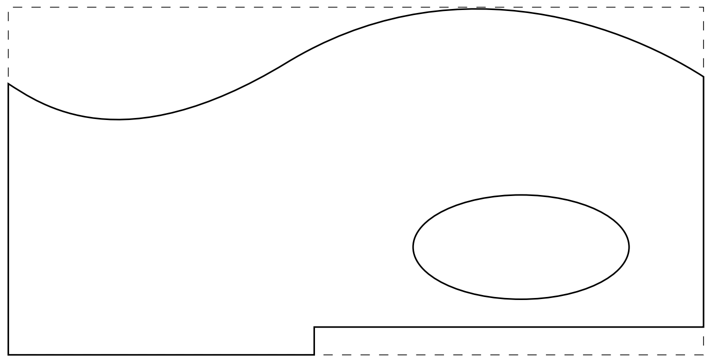


Figure 1: A non-convex exhibition area.

This problem can remind a number of two-dimensional packing problems, such as the two-dimensional bin packing problem and the two-stage two-dimensional cutting stock problem (see, e.g., Lodi, Martello and Vigo [1] and Wäscher, Haußner and Schuman [5], respectively, for recent surveys). It can also remind the facility layout problem, for which we refer the reader to the works by Widmer [6] and Singh and Sharma [4]. Nevertheless, to our knowledge, no study has been devoted to the specific problem we consider.

The only results concerning fair layouts we are aware of are those presented by Schneuwly and Widmer [2] who considered, however, a problem which is quite different from the one addressed in this paper. They referred indeed to a *rectangular* exhibition area, did not impose that the stands be placed into vertical strips, and considered stands of six different shapes. The problem (a real-world case situation arising at the regional fair in Romont, Switzerland) was modeled by discretizing the rectangular exhibition area, and solved through three constructive heuristics, which first place the stands and then construct the space for the aisles. Three possible objective functions were considered: the space utilization, the total attractiveness of the exhibition and the visitor convenience.

In Section 2 we present the problems addressed and describe their main features. Totally unimodular integer linear programming models are presented in Section 3, and then tested in Section 4 on a real-world case study.

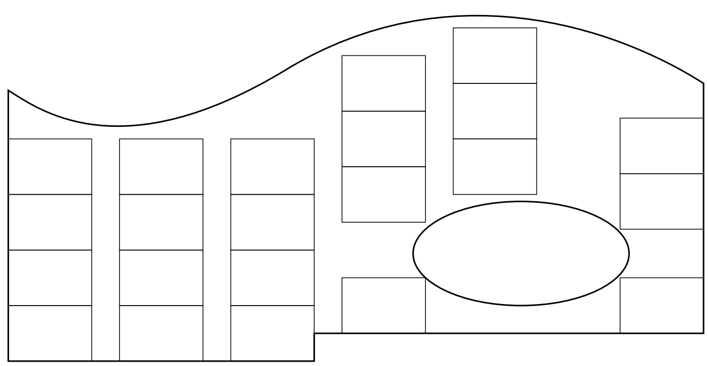


Figure 2: A single strip solution.

2 The Problems

We are given:

(i) a non-convex two-dimensional surface (*exhibition area*);

(ii) a rectangle which encapsulates the exhibition area and touches it on the borders (see the dotted lines in Figure 1);

(iii) an unlimited number of identical rectangular stands;

(iv) a minimum width needed for the aisles.

The *Fair Layout Optimization Problem* we consider consists in orthogonally allocating the maximum number of stands, without rotation, to vertical *strips* parallel to the vertical edges of the rectangle, by ensuring left and/or right (see below) side access to each stand.

Concerning the access constraint, we will consider two variants of the problem, that are frequently encountered in practice:

- FLOP1: it is required that each stand (i.e., each strip) can be accessed from both sides, as in the solution depicted in Figure 2;
- FLOP2: it is allowed to place pairs of strips with no space between them, thus obtaining stands that can be accessed from one side only, as in the solution depicted in Figure 3.

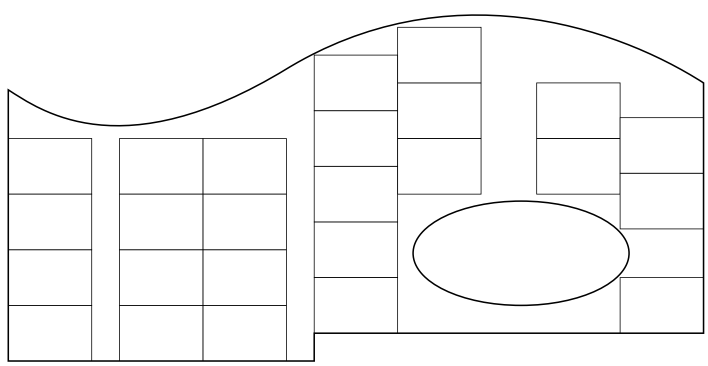


Figure 3: A double strip solution.

We will assume, without loss of generality, that the exhibition area can be accessed from all sides. For the cases where such assumption does not hold, the available area can be conveniently restricted (as it will be clear later).

We will also assume that the width and height (W and H) of the exhibition area, the width and height (w and h) of the stands and the minimum aisle width (a) are integers, i.e., that they are expressed in the minimum unit length, say δ, that is convenient to evaluate (typical δ values in real-world applications are in the range 5–20 cm). This is obtained by rounding down the rectangle sizes, and by rounding up the stands sizes and the aisle width, so as to guarantee a feasible solution.

Let M be an $H \times W$ binary matrix corresponding to the encapsulating rectangle (see Figure 4). For convenience of notation, we will number the rows and columns of M starting from the bottom-left corner, so that the indices correspond to coordinates in the corresponding discretized space. Matrix M is then defined as

$$M_{ij} = \begin{cases} 1 & \text{if the } \delta \times \delta \text{ square located at coordinate } (i,j) \text{ can be used for a stand;} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

for $i = 1, \ldots, H$ and $j = 1, \ldots, W$.

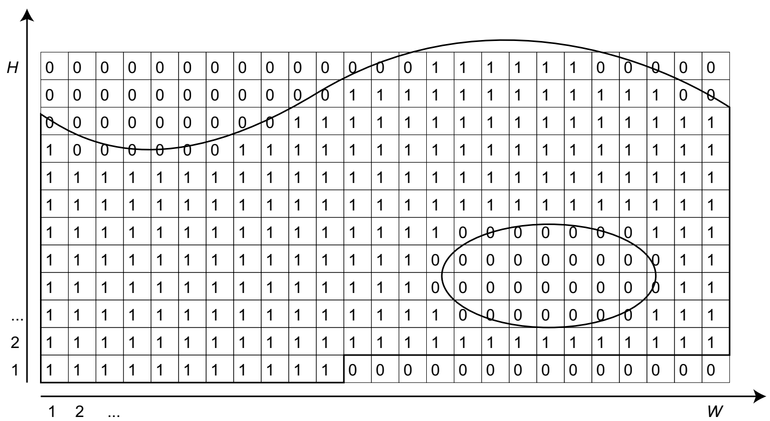


Figure 4: Matrix M associated with the exhibition area of Figure 1.

We say that a column j is selected for the layout if a strip of stands is placed with the left edge of the stands on column j. For example, column 1 is selected for the first strip of Figures 2 and 3 (where $w = 3$, $h = 2$ and $a = 1$)

Consider any column j, and any maximal row interval $[i_a, i_b]$ such that matrix M has all 1's in the rectangle of width w and height $(i_b - i_a + 1)$ having its bottom-left corner in (i_a, j). The maximum number of stands that can be placed within such rectangle is $\lfloor (i_b - i_a + 1)/h \rfloor$. In the example of Figure 4 we have, for $j = W - 3$, two maximal row intervals of width 3: one of height 2 and one of height 5. The corresponding strip can thus accommodate three stands in total.

For any column $j \leq W - w + 1$, the maximum number of stands v_j that can be placed with their left edges on column j can be obtained in a greedy way by iteratively determining the next group of consecutive rows satisfying the above condition. A straightforward implementation of such method would require, for each column, $O(H)$ iterations, each of time complexity $O(w)$, i.e., in total $O(WHw)$ time. We next give a simple procedure that computes the same information v_j $(j = 1, \ldots, W - w + 1)$ in $O(WH)$ time, i.e., in the order of time needed for the input of matrix M.

We define, for each row i such that $M_{i,j} = 1$ (j being the current column), a pointer $p(i)$ to the last column $\hat{\jmath}$ such that $M_{i,j} = M_{i,j+1} = \cdots = M_{i,\hat{\jmath}} = 1$. If instead $M_{i,j} = 0$ for the current column j then $p(i)$ has any value less than j. In this way all maximal row intervals can be determined by examining each element of M a constant number of times. Counter k stores, for the current column j and the current row interval, the number of feasible rows. The detailed procedure is given in Algorithm 1.

Algorithm 1 Computation of the v_j values.

Procedure Alloc_Stands
for $i := 1$ **to** H **do**
  $M_{i,W+1} := 0;$
  if $M_{i,1} = 1$ **then** $p(i) := \min\{k : M_{i,k} = 0\} - 1$ **else** $p(i) = 0$
end for;
for $j := 1$ **to** $W - w + 1$ **do**
  $v_j := k := 0;$
  for $i := 1$ **to** H **do**
    if $p(i) - j + 1 \geq w$ **then** $k := k + 1$
    else
      $v_j := v_j + \lfloor k/h \rfloor;$
      $k := 0;$
      if $(p(i) < j$ **and** $M_{i,j+1} = 1)$ **then** $p(i) := \min\{k > j : M_{i,j} = 0\} - 1$
    end if
  end for;
  $v_j := v_j + \lfloor k/h \rfloor$
end for
end

Our combinatorial optimization problems thus reduces to determining which columns j should be used for packing the strips of stands. (Note that, if a column j is selected, the placement of the v_j stands is straightforward.)

3 Mathematical Models

Let us associate a binary variable x_j,

$$x_j = \begin{cases} 1 & \text{if a single strip of stands has its left edges on column } j, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

to each column j where a strip can be selected, i.e., for $j = 1, \ldots, W - w + 1$. Our first problem can then be modeled as

$$\text{(FLOP1)} \quad \max \sum_{j=1}^{W-w+1} v_j x_j \tag{3}$$

$$\text{s.t.} \quad \sum_{j=k-a-w+1}^{k} x_j \leq 1 \qquad (k = w + a, \ldots, W - w + 1), \tag{4}$$

$$x_j \in \{0, 1\} \qquad (j = 1, \ldots, W - w + 1). \tag{5}$$

The set of constraints (4) imposes that the stands do not overlap and there is enough space left for the aisles: if column k is selected then columns $k - a - w + 1, \ldots, k - 1$ cannot be selected (see Figure 5 (a)). Note that the constraints (4) for $k = 1, \ldots w + a - 1$ are not imposed, as they are dominated by the first constraint ($k = w + a$).

Problem FLOP2 can be modeled in a similar way by introducing a second set of binary variables for each column j, namely

$$\xi_j = \begin{cases} 1 & \text{if a double strip of stands has its leftmost edges on column } j, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

for $j = 1, \dots, W - 2w + 1$. Observe that the maximum number of stands that can be placed in a double strip with the leftmost edges on column j is $v_j + v_{j+w}$. We thus obtain the model

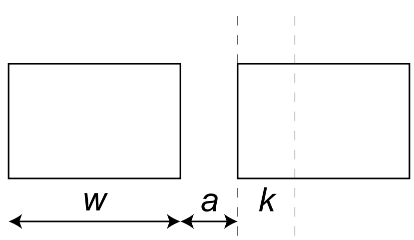
$$\text{(FLOP2)} \quad \max \sum_{j=1}^{W-w+1} v_j x_j + \sum_{j=1}^{W-2w+1} (v_j + v_{j+w}) \xi_j \tag{7}$$

$$\text{s.t.} \quad \sum_{j=k-a-w+1}^{k} x_j + \sum_{j=\max(1,\ k-a-2w+1)}^{\min(W-2w+1,k)} \xi_j \le 1 \qquad (k = w + a, \dots, W - w + 1), \tag{8}$$

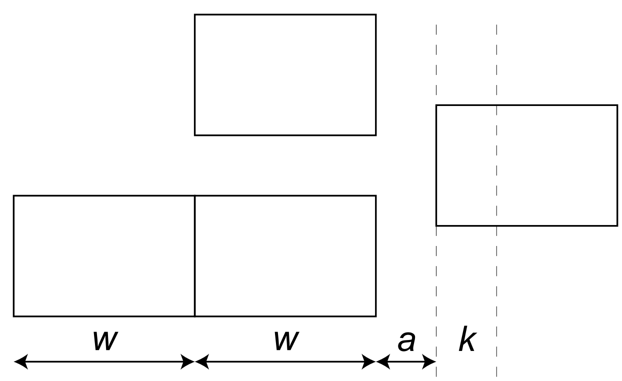
$$x_j \in \{0, 1\} \qquad (j = 1, \dots, W - w + 1), \tag{9}$$

$$\xi_j \in \{0, 1\} \qquad (j = 1, \dots, W - 2w + 1). \tag{10}$$

The set of constraints (8) imposes that the single and double strips of stands do not overlap, and there is enough space left for the aisles: if column k is selected (ether for a single or a double strip) then (see Figure 5 (b)) columns $k - a - w + 1, \dots, k - 1$ cannot be selected for a single strip, and columns $k - a - 2w + 1, \dots, k - 1$ cannot be selected for a double strip. The max and min operators in the second summation exclude indices j that are out of range.



(a) Constraint (4).



(b) Constraint (8).

Figure 5: Constraints visualization.

The structure of the constraint matrix induced by (8) is shown in Figure 6, which provides the constraint matrix corresponding to the instance depicted in Figure 4. (Additionally note that the leftmost portion of such matrix gives the structure of the constraint matrix induced by (4).)

It is not difficult to see that our two models possess the following relevant property.

Property 1 *The constraint matrices of FLOP1 and FLOP2 are totally unimodular.*

1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Figure 6: Constraint matrix for model FLOP2 associated with Figure 4.

Proof. Recall that a $0-1$ matrix in which each column has its ones consecutively is called an *interval matrix*. It is known (see, e.g., Schrijver [3], Chapter 19) that interval matrices are totally unimodular. The coefficients in each row i induced by (4) are all zeroes but a set of $w+a$ consecutive ones starting in column i, hence the resulting constraint matrix is an interval matrix. The coefficients in each row i induced by (8) are all zeroes but two sets of consecutive ones: one starting in column i for the x variables, and one starting in column $\max\{W-w+2, W-w+i-2\}$ for the ξ variables (refer again to Figure 6). Each column of the resulting constraint matrix has thus all zeroes but a set of consecutive ones, from which one has the thesis. $\triangle$

It follows that both problems FLOP1 and FLOP2 can be solved in polynomial time by replacing constraints (5) ($\equiv$ (9)) and (10) with their continuous relaxations

$$0 \leq x_j \leq 1 \qquad (j = 1, \dots, W-w+1), \tag{11}$$

$$0 \leq \xi_j \leq 1 \qquad (j = 1, \dots, W-2w+1) \tag{12}$$

and solving the resulting linear programming problem.

4 A real-word case

The models introduced in the previous section were implemented in Java and used for studying optimal layouts for the permanent Beira Mar handcraft fair of Fortaleza (Brazil), whose exhibition area is shown in Figure 7. The linear programs were solved using the freeware LP solver `lp_solve` Version 5.5. (see `http://sourceforge.net/projects/lpsolve/`.)

The algorithm was embedded in a graphical interface, so as to facilitate the inclusion of additional requests on the exhibition area. All tests were run on a dual-core Pentium IV with

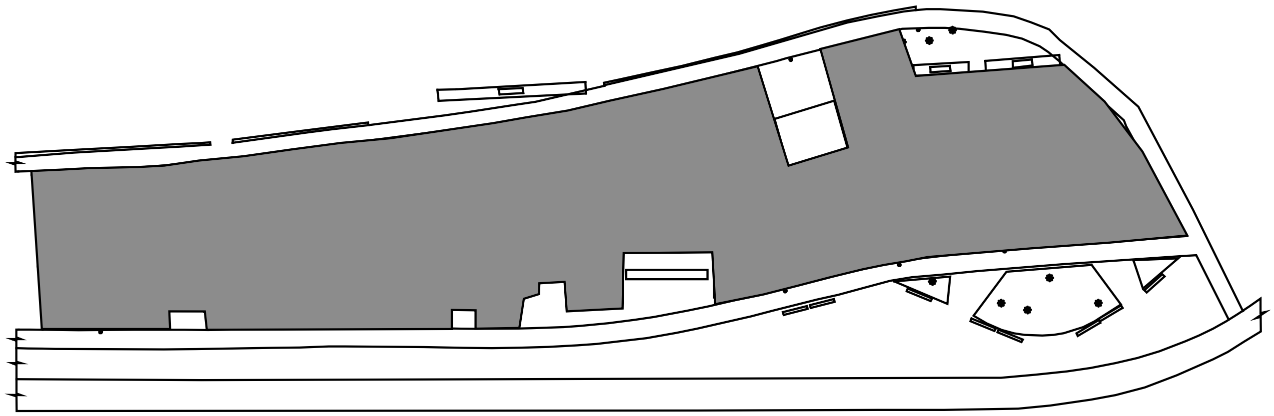


Figure 7: The Beira Mar handcraft fair exhibition area.

1.8 GHz and 2 GB RAM.

The fair is located in an area approximately 200 meters wide and 55 meters high. Its current configuration consists of 617 stands, placed along 26 double strips roughly similar to those produced by FLOP2. The width and height of each stand are 2 meters.

The available area for placing the stands is depicted in grey in Figure 7. The area is non-convex due to the presence of palm trees, small shopping facilities, light poles, a monument and some sub-areas which cannot be used for the stands.

We solved the problem by setting $\delta = 5$ cm, thus obtaining $w = h = 40$, $W = 4090$ and $H = 1084$. The aisle width was set to the same value currently adopted, i.e., $a = 70$ (3.5 meters).

In order to test the algorithm we solved both FLOP1 (very different from the current configuration) and FLOP2. The input consisted of an AutoCAD map, from which an SHP file was produced and georeferenced to the the actual coordinates of the exhibition area. A Borland Delphi Object Pascal program was then executed on the SHP file to obtain the 1084×4090 binary matrix M (common to the two models). This was the most time consuming phase, requiring 979 CPU seconds.

Model FLOP1 was constructed and solved in 24 CPU seconds. The resulting solution, shown in Figure 8, consists of 505 stands placed along 36 single strips, i.e., with less than 20% decrease with respect to the current double strip configuration.

Model FLOP2 was constructed and solved in 62 CPU seconds. The solution, shown in Figure 8, includes 742 stands placed along 26 double strips, increasing by 20% the current number of stands. The decision makers wanted to also test the possibility of allowing the use of the roughly triangular area located in the upper right corner of the exhibition area. The outcome was an increase to 759 stands placed along 26 double strips. None of the solutions produced by FLOP2 combined double and single strips, probably due to the fact that the aisles are relatively quite large, thus making their use not convenient.

We also tested different values of δ, obtaining similar results for what concerns the numbers of stands, but considerable differences in the CPU times. For $\delta = 10$ cm the algorithm needed 246 CPU seconds to build matrix M, 4 seconds to solve FLOP1 and 10 seconds to solve FLOP2. The solution of FLOP1 included 503 stands instead of 505, while the solution of

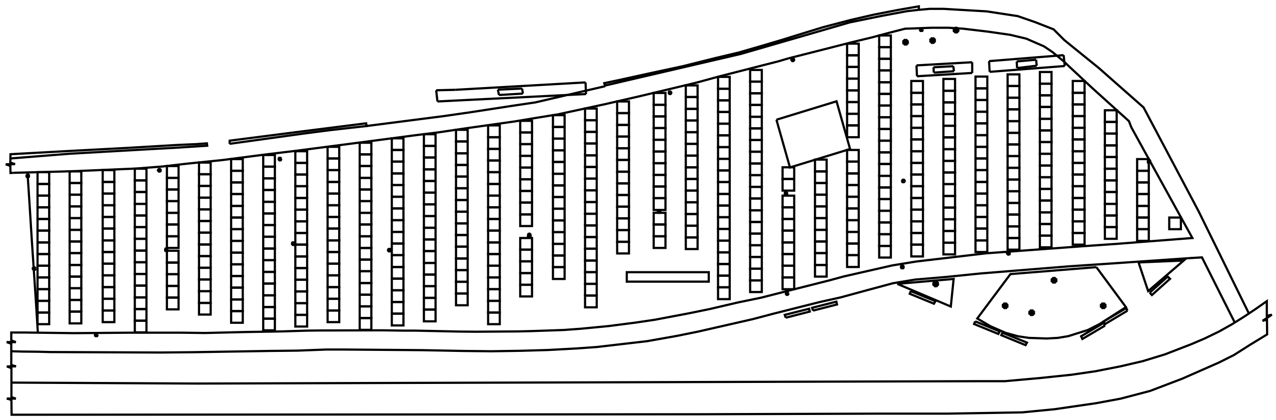


Figure 8: Solution produced by FLOP1.

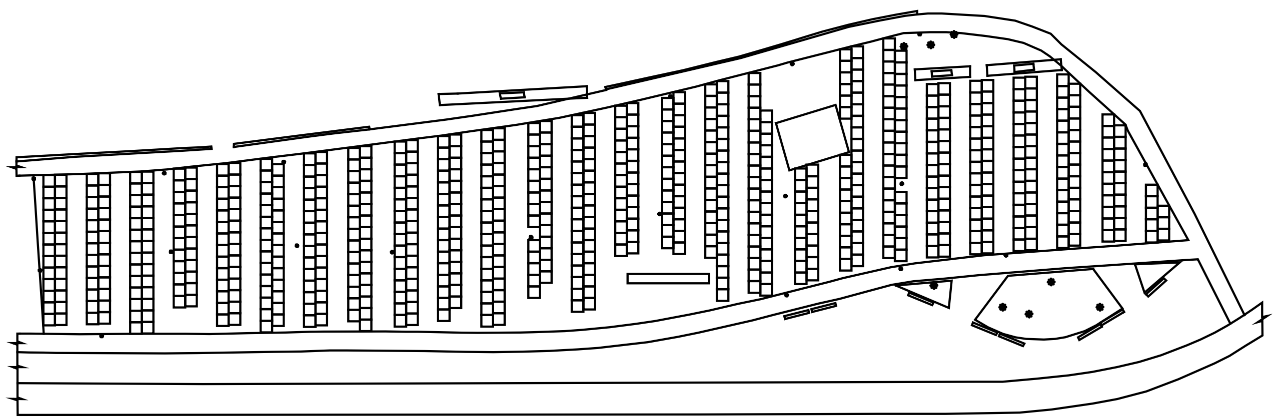


Figure 9: Solution produced by FLOP2.

FLOP2 was still composed by 742 stands. For $\delta = 20$ cm, the CPU times were 64 seconds to create the matrix, and 2 and 1 seconds to solve the two models, with a further decrease to 499 and 737 stands. Values of δ lower than 5 cm produced very high CPU times with no improvement.

5 Conclusions

We addressed a particular layout problem derived from a real-world situation. We proposed two LP models and showed that their constraint matrices are totally unimodular. The solutions obtained have been given to the local authorities, that are currently discussing the new fair layout configuration to be implemented.

Several additional matters of interest arise in fair layout optimization. In the case we considered, the decision makers imposed that the orientation of the strips had to be perpendicular to the large alley that can be seen on the bottom of the maps. Allowing different orientations could produce better solutions. It would also be interesting to consider other configurations that are sometimes adopted in fair layouts such as, e.g., the use of blocks of four stands each, with both vertical and horizontal aisles.

Acknowledgements

We thank the ALFA PROJECT, under grant II-0457-FA-FCD-F1-FC. We also thank the GRAPHVS Ltda and the Universidade Estadual do Ceará, Fortaleza (Brazil), and the Ministero dell'Università e della Ricerca (Italy). Thanks are also due to Ana Mércia (Municipality Environmental Manager), Sérgio Pinheiro (Fortaleza Tourism Municipality Secretary) and Francisco Carlos Simões (President of the Beira Mar Fair Association, ASFABEM) for their kind support during this research.

References

[1] A. Lodi, S. Martello, and D. Vigo. Recent advances on two-dimensional bin packing problems. *Discrete Applied Mathematics*, 123:379–396, 2002.

[2] P. Schneuwly and M. Widmer. Layout modeling and construction procedure for the arrangement of exhibition spaces in a fair. *International Transactions in Operational Research*, 10:311–338, 2003.

[3] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Chichester, 1986.

[4] S.P. Singh and R.R.K. Sharma. A review of different approaches to the facility layout problems. *The International Journal of Advanced Manufacturing Technology*, 30:425–433, 2006.

[5] G. Wäscher, H. Haußner, and H. Schumann. An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183:1109–1130, 2007.

[6] M. Widmer. *Modèles mathématiques pour une gestion efficace des ateliers flexibles*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Switzerland, 1990.