

Drawing Constrained Rectangles in the Plane without Intersections

M. A. Garrido, A. Márquez, J. R. Portillo and P. Reyes
Dept. de Matemática Aplicada I, Universidad de Sevilla
Sevilla, E-41012, Spain
{vizuete, almar, josera, preyes}@cica.es

ABSTRACT

This paper is devoted to study a particular case of the general problem known as Labeling. We want to draw rectangles in the plane in such a way that between any pair of them either one rectangle is contained in the other or both are disjoint. For each one of these rectangles a set of three points is given, such that k of them are corners of the rectangle and the other $3 - k$ points are in its sides. We prove that if $k = 2$ or 3 then the problem is solvable in polynomial time and it is NP-complete for $k = 1$.

Keywords: Orthogonal Layouts, Rectangles Intersection, Graph Drawing, Labeling, Geographic Information Systems.

1. INTRODUCTION

Two are the main ingredients of this work, namely, orthogonal drawing and geometry of rectangles. Orthogonal layouts are used in many applications of Computer Science. Although firstly they were considered mainly as a tool for VLSI-design, nowadays, Information Systems is perhaps the main area of application, spreading its influence to data flow diagrams, database design or entity relationships diagrams [1, 2, 11]. This fact has attracted the attention of many authors and numerous

results have been obtained about orthogonal drawings [3, 7, 8, 12, 13, 16, 17]. Equally, it could be said that the geometry of rectangles has applications to the same areas (see, for example [15]), but another specific application is as a subfield of G.I.S. (Geographic Information Systems) known as Labeling [5, 10].

Among the applications we have mentioned above, the problem we face in this work has clearly applications to database design, entity relationships diagrams and to labeling. We want to draw some rectangles in the plane in such a way that between any pair of them either one rectangle is contained in the other or both are disjoint. In general, many variations of this problem are NP-complete problems, and it is usually assumed that some constraints are known for each rectangle. A very usual constraint is to know at least one of the vertices of each rectangle (one-corner elastic labeling problem) but this problem is NP-complete and it remains so even if other points of its border are known, unless all those other points are placed on two axes (see [9]). But, in general, this restriction is too stringent for many practical purposes. In this work we relax these restrictions in order to cover more applications. Thus we know one of the corners of each rectangle and two additional points in different sides such that at least one of them must be a corner as well. We also will prove that if this last point is not a corner, the problem remains NP-complete.

¹Partially supported by dgicyt project PB96-1374 and PAI project FQM-0164

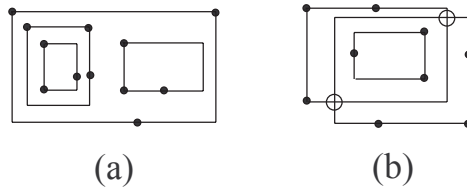


Figure 1: Two inputs of OR(2) problem, (a) with solution, (b) without solution.

In other words, we face the problem in the following way

Orthogonal Rectangles with Corners

OR(k)

INSTANCE: Set of triples $\{T_1, T_2, \dots, T_n\}$, where each T_i is a set of three points in the plane and $k = 1, 2$ or 3 .

QUESTION: Can be traced simultaneously n rectangles $\{R_1, R_2, \dots, R_n\}$, such that for each set T_i , at least k of its points are corners of R_i and the other $3 - k$ points are in its sides, and such that either $R_i \subseteq R_j$ or $R_i \cap R_j = \emptyset$?

Figure 1 shows an example of OR(2) problem. For each rectangle its three points are pictured and we draw the configuration of rectangles if there is solution.

In fact, keeping in mind the kind of applications of these problems, the drawings of the different labels (rectangles in the problem) have to admit a minimum separation distance. So, we will suppose that if $R_i \cap R_j = \emptyset$, then the distance between R_i and R_j is greater than a given $\varepsilon > 0$.

Thus, in the following section we will prove that OR(3) problem is solvable in optimal $O(n \log n)$ time and that OR(2) can be solved in $O(n^2)$ time. For $k = 1$, the computational character of the problem is very different, proving that OR(1) is an NP-complete problem.

2. POLYNOMIAL SOLVABLE CASES

In this section we deal with the cases in which two or three of the given points for each rectangle are corners of it.

Theorem 1 OR(3) can be solved in optimal $O(n \log n)$ time.

Proof: If three corners are fixed for each rectangle (OR(3)), we can only wire the triples whose points are L-shaped (two arbitrary points have to be in the same vertical or horizontal line), that is, the three points are in the vertices of a rectangle. Of course, this fact can be checked in linear time, thus, the problem is reduced to intersection of rectangles and there is an $O(n \log n)$ algorithm that solve the problem in optimal time (see [4, 14]). \square

If we know two corners (that is, the problem that we have denoted as OR(2)) then there are infinitely many possibilities for each rectangle, depending on the position of the points (see Figure 2). In spite of this fact, a polynomial (actually, a quadratic) algorithm exists in this case. This algorithm is based again in a reduction to a problem in Computational Geometry;

Theorem 2 OR(2) can be solved in $O(n^2)$ time.

Proof: First of all, it can be checked in linear time that there exists no triple of points as Figure 2 (a).

For the triples with solution, each rectangle must have two of its points p_1, p_2 in the same horizontal or vertical line, say in the same horizontal. Now, depending on the position of the

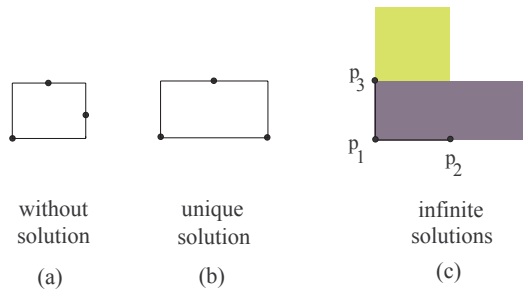


Figure 2: Different cases for OR(2) problem.

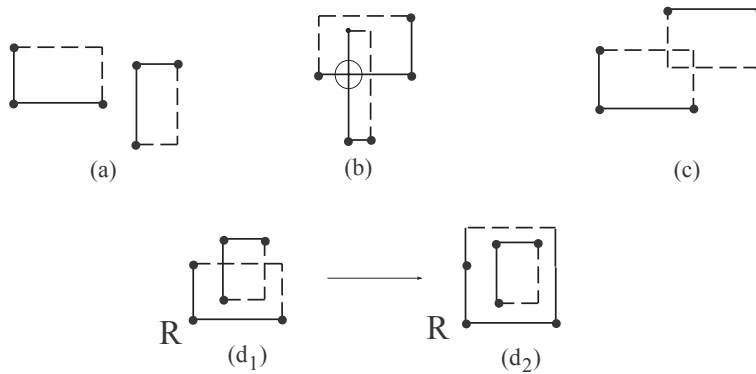


Figure 3: Different intersections between rectangles.

third point p_3 , two possible cases can occur. If p_3 is not in the same vertical of p_1 or p_2 , there exists only a possible solution for the triple (see Figure 2 (b)). Otherwise, if p_3 is in the same vertical of p_1 (or p_2), there are infinitely many possible solutions for the triple, all of them sharing the segments p_1p_3 and p_1p_2 , and defining, in this way two orthogonal bands for each triple (see Figure 2 (c)). In addition, one of those segment must be a side of the rectangle solution. Now, we try to find non-intersecting rectangles with the constrains given above, and we can prove that this problem can be solved in quadratic time using an incremental algorithm in the following way.

Each rectangle is identified by the four coordinates of its sides (top, bottom, left and right)

and a Boolean variable is associated to each coordinate, pointing out if this one is changeable (value 1) or if it is fixed (value 0). Of these four variables, initially two are zero and only one coordinate can be changed by the algorithm.

The intersection of two rectangles corresponds to one of the following cases. We have represented the fixed coordinates by continue lines and the variable ones by broken lines (Figure 3).

1. Without intersection (Figure 3 (a)).
2. Without solution (Figure 3 (b) and (c)).
3. Only one non-fixed side of a rectangle R intersects to another rectangle (Figure 3 (d₁)), so its coordinate changes and the Boolean variables associated to the

other sides of R change to zero (Figure 3 (d_2)).

We maintain a tree associated to each partial solution, each node n_i in the tree has associated with it a rectangle R_i and its subtree is constituted by all the nodes associated to rectangles that are contained in R_i . The insertion of a new triple can modify some previous rectangles, depending on its intersections (see Figure 3). This information is obtained from the four coordinates and Boolean variables associated to each rectangle.

The maintaining of this tree can be done in linear time for each insertion, so the global cost of the algorithm is $O(n^2)$ \square

Figure shows an input of OR(2) problem, with the associated bands to the triples and a possible solution.

3. NP-COMplete CASES

We have seen, in previous section, that, for $k = 2$, our problem can be solved in quadratic time although there exist infinitely many solutions for some triples. But, if we impose that only one point of the triples is corner of the rectangles ($k = 1$), then the NP-completeness of the problem is obtained.

Theorem 3 *OR(1) is an NP-complete problem.*

Proof: OR(1) is readily seen to be solvable in non-deterministic polynomial time. So, in order to show the problem is NP-complete, it suffices to transform an NP-complete problem to ours. In order to do this we consider the well-known NP-complete problem PLANAR-3SAT [6]. Let $U = \{u_1, u_2, \dots, u_p\}$ be a set of variables and let $C = \{c_1, c_2, \dots, c_q\}$ be a set of clauses making up an arbitrary instance \mathcal{S} of PLANAR-3SAT. Let $\mathcal{G}_{\mathcal{S}} = (V, E)$ denote

the planar bipartite graph where $V = U \cup C$ and E contains exactly those pairs $\{u, c\}$ such that either u or \bar{u} belongs to the clause c . Corresponding to \mathcal{S} , we construct, in polynomial time, an instance \mathcal{W} of OR(1) such that \mathcal{S} is satisfied if and only if \mathcal{W} can be laid out.

For each triple of \mathcal{W} that we will construct, a finite number of rectangle-solutions (at most four) will be allowed. The other possible rectangles will be “blocked” by a triple with unique solution, showed in Figure 5.

The first step consists in constructing an orthogonal representation $\Gamma_{\mathcal{G}_{\mathcal{S}}}$ of $\mathcal{G}_{\mathcal{S}}$ in the plane, where the vertices (variables and clauses) are depicted by boxes. Now, each clause of \mathcal{S} is represented by a triple, whose points are placed on the sides of the associated box in $\Gamma_{\mathcal{G}_{\mathcal{S}}}$. Considering blocker-triples drawn on suitable places, this triple can determine only three possible rectangles as Figure 6 shows.

Each variable is represented by a rectangle formed by an even number of triples (for each one of them, the blocker-triples only allow two possible rectangles as Figure 7 shows). Observe that, once we fix one of the allowed rectangles of a triple, in all the other triples, the compatible rectangle is fixed, so we can associate one rectangle with the “true” literal and the other with the “false” literal. So far, in some sense, we have represented the vertex set of $\mathcal{G}_{\mathcal{S}}$; now, we represent the edges joining each clause with the corresponding variables (actually with the corresponding rectangle associated to literal presented in each clause) as Figure 8 shows.

With these elements we have finished the construction of \mathcal{W} , and it is not difficult to see that the positive answer to PLANAR-3SAT with \mathcal{S} as an input is equivalent to the positive answer of OR(1) with \mathcal{W} as an input. \square

4. CONCLUSIONS AND OPEN PROBLEMS

In this work, we have dealt with orthogonal lay-

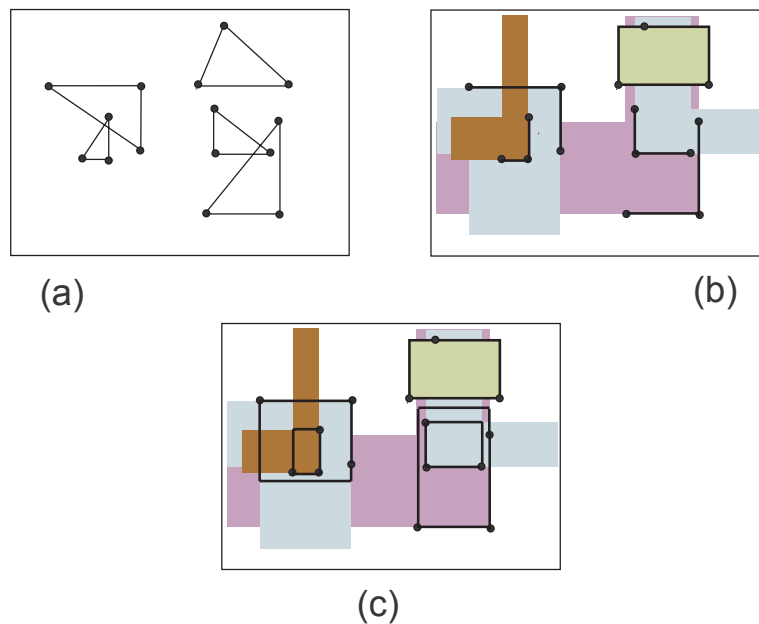


Figure 4: (a) An input of OR(2) problem. (b) The bands defined by this input. (c) A solution of this example.

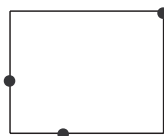


Figure 5: Blocker-triple.

outs and geometry of rectangles. These topics appear in many applications of Computer Science, particularly in Information Systems as Problems of Labeling. Summarizing the previous results, we obtain the following theorem

Theorem 4 *OR(k) is solvable in polynomial time if and only if $k \geq 2$. Unless P=NP.*

We can interpret our general problem as connections of triple of points by a orthogonal triangle with fixed restrictions. This interpretation lead us to consider our problem as an ex-

tension of Single Bend Wiring problem [16], that connects pairs of points by orthogonal layouts using only one bend. In this sense, many open problems arise in this area, considering different sets of points and using orthogonal embeddings with several bends.

References

- [1] C. Batini, E. Nardelli and R. Tamassia, "A Layout Algorithm for Data-Flow Di-

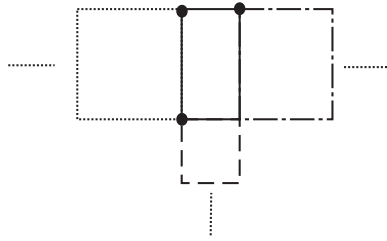


Figure 6: Clause represented by a triple with three possible solutions.

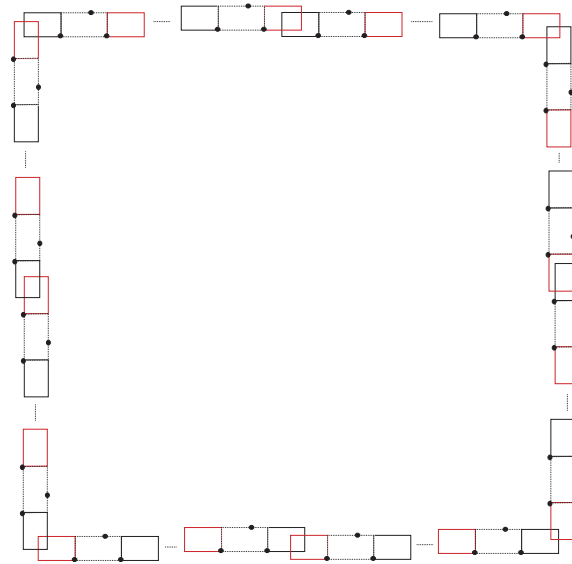


Figure 7: Variable represented by a rectangle of triples.

- agrams”, *IEEE Trans. on Software Engineering*, SE-12.4, 1986, pp. 538–546.
- [2] C. Batini, M. Talamo and R. Tamassia, “Computer Aided Layout of Entity–Relationship Diagrams”, *The Journal of Systems and Software*, Vol. 4, 1984, pp. 163–173.
- [3] T.C. Biedl, “Optimal orthogonal drawings of triconnected plane graphs”, *Proc. of SWAT’96*, LNCS Vol. 1097, Springer–Verlag, 1996, pp. 333–344.
- [4] H. Edelsbrunner, “Dynamic data structures for orthogonal intersection queries”, Rep. F59, Tech. Univ. Graz. Institute für Informationsverarbeitung, 1980.
- [5] M. Formann and F. Wagner, “A packing problem with applications in lettering of maps”, *Proceedings of the 7th ACM Symposium on Computational Geometry*, 1991, pp. 281–288.
- [6] M.R. Garey and D.S. Johnson, *Computers and Intractability: a guide to the theory of NP–completeness*, New York: Freeman, 1979.
- [7] M.A. Garrido and A. Márquez, “Embedding a graph in the grid of a surface with the minimum number of bends is NP–hard”, In G. DiBattista, editor, *Graph Drawing (Proc. of GD’97)*, LNCS, Vol. 1353, Springer–Verlag, 1998, pp. 124–133.
- [8] M.A. Garrido, A. Márquez, A. Morgana and J.R. Portillo. “Single bend wiring on

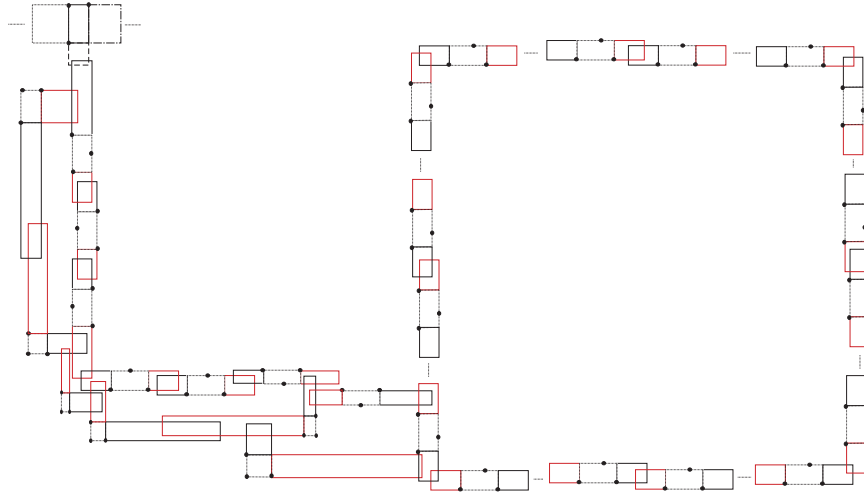


Figure 8: Clause and variable joined by an edge.

surfaces”, submitted, 1998, 12 pages.

- [9] C. Iturriaga and A. Lubiw “Elastic Labels: the two-axis case”, In G. DiBattista, editor, *Graph Drawing (Proc. of GD’97)*, LNCS, Vol. 1353, Springer-Verlag, 1998, pp. 181–192.
- [10] D. Knuth and A. Raghunathan, “The problem of compatible representatives”, *SIAM Disc. Math.*, Vol. 5, No. 3, 1992, pp. 442–427.
- [11] Th. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout*, Teubner/Wiley & Sons, 1990.
- [12] Y. Liu, A. Morgana and B. Simeone, “General theoretical results on rectilinear embeddability of graphs”, *Acta Math. Appl. Sinica*, Vol. 7, 1991, pp. 187–192.
- [13] Y. Liu, A. Morgana and B. Simeone, “A linear algorithm for 2-bend embeddings of planar graphs in the two-dimensional grid”, *Discrete Applied Mathematics*, Vol. 8, 1998, pp. 69–91.
- [14] E. M. MacCreight, “Priority search trees”, Tech. Rep. CSL-81-5, Xerox PARC, 1981.
- [15] F.P. Preparata and M.I. Shamos, *Computational Geometry. An Introduction*, New York: Springer-Verlag, 1985.
- [16] R. Raghavan, J. Cohoon and S. Sahni, “Single Bend Wiring”, *Journal of Algorithms*, Vol. 7, 1986, pp. 232–257.
- [17] R. Tamassia, “On embedding a graph in the grid with the minimum number of bends”, *SIAM J. Comput.*, Vol. 16, No. 3, 1987, pp. 421–444.