Elastic Labels: The Two-axis Case

Claudia Iturriaga*
Anna Lubiw **

Dept. of Computer Science, University of Waterloo

Abstract. One of the most challenging tasks of cartographic map lettering is the optimal placement of region information on a map. We propose as an approach to this task the *elastic labeling problem*, in which we are given a set of *elastic rectangles* as labels, each associated with a point in the plane. An elastic rectangle has a specified area but its width and height may vary. The problem then is to choose the height and width of each label, and the corner of the label to place at the associated point, so that no two labels overlap.
This problem is known to be NP-hard even when there is no elasticity (just because of the choice of the corners). We show that the problem remains NP-hard when we have elasticity but no choice about which corner of the label to use—we call this the *one-corner elastic labeling problem.* We give a polynomial time algorithm for the special case of the one-corner elastic labeling problem in which the points lie on the positive x and y axes and the labels lie in the first quadrant. We call this the *two-axis labeling problem.*

1 Introduction

An essential part of communication in a map or a diagram is the written information included on it ([DFr92]). In the case of cartography, work has been done to determine a good set of rules for how a region should be labeled to agree with its geographical characteristics ([Imh75]). In this paper we focus on the actual placement of information on a map.

A special case is when we must label points. A problem arises when the map contains thousands of points that are too close together. In this case, it is difficult to place the labels so that they do not overlap and are easy to read without magnification. Algorithms have been developed to automate the process of placing labels on a map ([Yoe72], [CMS95], [KRa92], [FWa91]). One formulation of the label placement problem is the *point-feature-label placement problem* of [FWa91], [KIm88], [KRa92], and [MSh91], where we are given a set of points in the plane and with each point an associated axis-parallel rectangular label. The problem is to place the labels so that they do not overlap, and so that each label has one of its corners at its associated point. This problem is known to be NP-complete [FWa91], [KIm88], [KRa92], and [MSh91] and we can find heuristics in [CMS95],

* email: cciturri@barrow.uwaterloo.ca
** email: alubiw@daisy.uwaterloo.ca. Research partially supported by NSERC

[FWa91] and [WWo97]. For the special case where only two corners of each label can be placed at the point, there is an $O(n \log^2 n)$ algorithm [FWa91].

In this paper we introduce the *elastic labeling problem* which generalizes the above point-feature-label placement problem in that, in addition to choosing the corner of the label to place at the point, we are free to choose the height and width as long as we achieve the specified area.

This problem is useful when the goal of placing a label at a given point is to associate some text (more than one word) with the point. In this case we are able to write the specified text inside the label by using one, two, or more rows, as long as the label is placed at the specified point.

It is clear that the elastic labeling problem is NP-hard since it generalizes the point-feature-label placement problem. We show that even the *one-corner elastic labeling problem*—in which we fix the corner of each label that must be placed at the point—is NP-hard.

Our main result is a polynomial time algorithm for the *two-axis labeling problem*. This is the special case of the one-corner elastic labeling problem in which the points lie on the positive x and y axes, and the labels lie in the first quadrant. The two-axis labeling problem is the main step in solving the more general and more natural problem when the labels lie on four sides of an enclosing rectangle. This "rectangle perimeter labeling problem" arises when the perimeter of a map is labeled with information about things that lie beyond the boundary of the map, e.g. where the roads lead to, etc. This problem is likely to be relevant in GIS as maps are displayed dynamically on a computer screen using clipping, panning, and zooming.

The paper is structured as follows. In section 2, we define elastic rectangles, the elastic labeling problem and the two-axis labeling problem. Section 3 contains an algorithm for the two-axis labeling problem. In section 4 we present the NP-hardness proof for the one-corner elastic labeling problem.

2 Definitions

2.1 Elastic Rectangles

We define an orthogonal rectangle R by a quadruple (x, y, x', y') where the pair (x, y) sets the coordinates of the bottom left corner and the pair (x', y') sets the coordinates of the upper right corner. For a rectangle R we define $\mathit{left}(R) = x$, $\mathit{bottom}(R) = y$, $\mathit{right}(R) = x'$, $\mathit{top}(R) = y'$, $\mathit{height}(R) = y' - y$, and $\mathit{width}(R) = x' - x$. We consider rectangles to be topologically open.

Definition 1. An *elastic rectangle* $\mathcal{E}$ is a family of rectangles specified by a quintuplet (p, α, H, W, Q) where p is a point that is a corner of any rectangle in $\mathcal{E}$, α is the area of any rectangle in $\mathcal{E}$, H is the range of the height of the rectangles $[h^{\min}, h^{\max}]$, W is the range of the width $[w^{\min}, w^{\max}]$, and $Q \subseteq \{1, 2, 3, 4\}$ is a set of possible positions of p allowed in the family. The value of the position is 1 when p is a bottom left corner, 2 when p is a top left corner, 3 when p is a top right corner, and 4 when p is a bottom right corner.

We use the notation $p(\mathcal{E})$, $\alpha(\mathcal{E})$, $H(\mathcal{E})$, $W(\mathcal{E})$, and $Q(\mathcal{E})$ for the parameters of an elastic rectangle $\mathcal{E}$. The point $p(\mathcal{E})$ will be called the *anchor* of $\mathcal{E}$.

When Q is a singleton, the family of rectangles $\mathcal{E}$ is described by a hyperbola segment tracing out the locus of the corner of the elastic rectangle opposite p. Figure 1 shows an elastic rectangle with $Q(\mathcal{E}) = \{1\}$, with the hyperbola as a dashed curve.

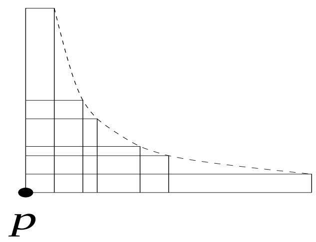

Fig. 1. An elastic rectangle $\mathcal{E}$.

An elastic rectangle $\mathcal{E}$ is *x-base* if $Q(\mathcal{E}) = \{1\}$ or $Q(\mathcal{E}) = \{4\}$ and the bottom side of its rectangles lies on the positive x-axis. Similarly $\mathcal{E}$ is *y-base* if $Q(\mathcal{E}) = \{1\}$ or $Q(\mathcal{E}) = \{2\}$ and the left side of its rectangles lies on the positive y-axis.

2.2 Elastic Labeling Problems

In this subsection we define the elastic labeling problem and the special case of the one-corner labeling problem.

A *realization* of an elastic rectangle $\mathcal{E}$, denoted E, is a single rectangle from the family—i.e. we must choose a valid height, width, and corner to place at p. A realization of a set of elastic rectangles is called a *good realization* if the rectangles do not intersect pairwise (Figure 2).

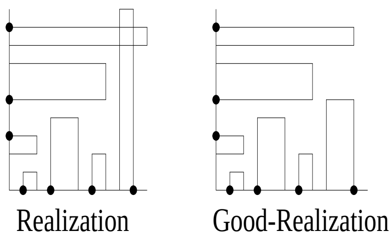

Fig. 2. Realizations.

The *elastic labeling problem* is: Given a set of elastic rectangles, find a good realization.

The special case when $|Q| = 1$ for each point is called the *one-corner elastic labeling problem.* This problem is shown to be NP-hard in section 4. In the following subsection we describe a subproblem that we are able to solve in polynomial time.

2.3 The Two-axis Labeling Problem

This section contains the definition and preliminary discussion of the *two-axis labeling problem*: given a set of x-base elastic rectangles, and a set of y-base elastic rectangles, find a good realization for the union of the two sets. See Figure 2.

Let $B = \{\mathcal{B}_1 \ldots \mathcal{B}_n\}$ be a set of x-base elastic rectangles with $p_i = p(\mathcal{B}_i)$ and $x(p_i) \leq x(p_{i+1})$ for $i = 1, \ldots, n-1$. Let $L = \{\mathcal{L}_1 \ldots \mathcal{L}_m\}$ be a set of y-base elastic rectangles with $q_i = p(\mathcal{L}_i)$ and $y(q_i) \leq y(q_{i+1})$ for $i = 1, \ldots, m-1$.

We need one more definition: the *height* [*width*] of a realization is the height [width] of the smallest axis-parallel rectangle containing all the rectangles of the realization.

Our algorithm for the two-axis labeling problem will add points one by one. To gain some intuition about where the difficulty of the problem lies, observe that when two consecutive points along the x-axis have their rectangles facing each other (as in Figure 3), we can trade-off the height of one against the height of the other, and which solution is best depends on the particular y-base rectangles we have.

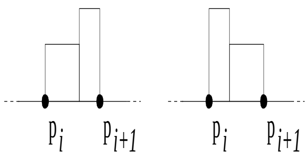

Fig. 3. Elastic rectangles facing each other.

3 Algorithm for the Two-axis Labeling Problem

Our algorithm for the two-axis labeling problem is a dynamic programming algorithm in which we will add points on by one along the x and y axes. We will begin in section 3.1 by describing the global structure of a solution. This allows us to identify the subproblems to be solved in the dynamic programming algorithm. Section 3.2 contains the complete algorithm.

3.1 Structure of a Good Realization

Our dynamic programming algorithm hinges on the following.

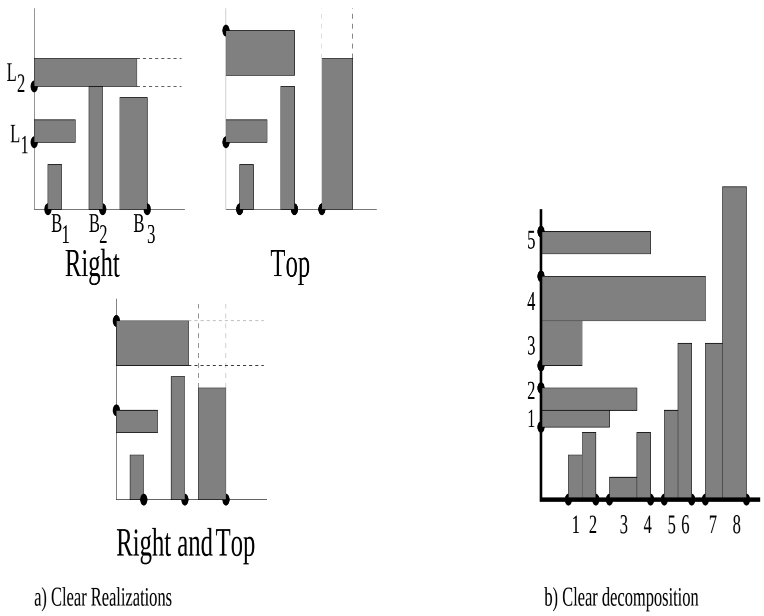

Fig. 4. Good clear realizations.

Observation 1. In a good realization of $\mathcal{B}_1, \ldots, \mathcal{B}_n$ and $\mathcal{L}_1, \ldots, \mathcal{L}_m$, using rectangles $B_1, \ldots B_n$ and $L_1, \ldots, L_m$, respectively, either B_n is *top-clear*—i.e. all the other rectangles lie to the left of the x-coordinate $\mathit{left}(B_n)$—or L_m is *right-clear*—i.e. all the other rectangles lie below the y-coordinate $\mathit{bottom}(L_m)$. See Figure 4.a.

We will call a realization *top-clear* if its last x-base rectangle is top-clear; we will call a realization *right-clear* if its last y-base rectangle is right-clear. Note that a realization can be both top-clear and right-clear.

The above observation tells us that we can always "peel away" the rectangles of a good realization, always removing a top-clear or right-clear one. Such an ordering (actually, its reverse) will drive our dynamic algorithm. More formally, let $\mathcal{S}_{i,j}$ be the subproblem of finding a good realization for $\mathcal{B}_1, \ldots, \mathcal{B}_i, \mathcal{L}_1, \ldots, \mathcal{L}_j$. Let $\mathcal{S}_{0,0}$ be the null subproblem.

Theorem 2. *Suppose $B_1, \ldots, B_n$, $L_1, \ldots, L_m$ is a good realization of an instance of the two-axis labeling problem, and let $S_{i,j}$ be the good realization induced for the subproblem $\mathcal{S}_{i,j}$. Then there is a sequence $(0,0), \ldots, (n,m)$ such that, if (i,j) appears in the sequence then the next element of the sequence is either $(i+1,j)$ and the good realization $S_{i+1,j}$ is top-clear, or $(i,j+1)$ and the good realization $S_{i,j+1}$ is right-clear.*

The proof of this theorem follows from the above observation. We call such a sequence a *clear decomposition.* For example, in Figure 4.b a clear decomposition is (0,0), (1,0), (2,0), (2,1), (3,1), (3,2), (4,2) (4,3), (5,3), (6,3), (6,4), (7,4), (7,5), (8,5). Note that the clear decomposition is not unique, we can replace (6,4) by (7,3) in this sequence and obtain another clear decomposition of the same realization.

3.2 Algorithm

The subproblems solved by our dynamic programming algorithm are as follows. For each $i, j, 1 \leq i \leq n, 1 \leq j \leq m$ we will solve two subproblems $\mathcal{S}^T_{i,j}$ and $\mathcal{S}^R_{i,j}$. For $\mathcal{S}^T_{i,j}$ we will find the "best" good realizations of $\mathcal{B}_1, \ldots, \mathcal{B}_i, \mathcal{L}_1, \ldots, \mathcal{L}_j$ with B_i top-clear. For $\mathcal{S}^R_{i,j}$ we will find all the "best" good realizations for $\mathcal{B}_1, \ldots, \mathcal{B}_i, \mathcal{L}_1, \ldots, \mathcal{L}_j$ with L_j right-clear. What do we mean by "best"? Since we will only ever add top-clear x-base rectangles and right-clear y-base rectangles, the only things that matter about a realization are its height and width. Recall that the height [width] of a realization is the maximum height [width] of a rectangle in the realization. The height of the realization matters for adding a right-clear rectangle, and the width matters for adding a top-clear rectangle.

In case there is a unique minimum height and width for a good realization, we want a good realization of that height and width. The good realization itself will in general not be unique, but that does not matter. The left pane of Figure 5 shows a case where there is a unique minimum height and width top-clear good realization.

Often there will be no unique minimum height and width—height and width may be traded off as they are in a single elastic rectangle. For example in the right pane of Figure 5, if we take the highest rectangle in $\mathcal{B}_2$ we have a top-clear good realization with minimum width, but if we take the widest rectangle then we have a top-clear good realization with minimum height. In this case we will preserve all possible height-width combinations. As we shall see, we can fix the realizations of all rectangles except the last one (the clear one), which must remain elastic.

To be more specific, in solving $\mathcal{S}^T_{i+1,j}$ we are adding $\mathcal{B}_{i+1}$ as a top-clear rectangle. If $\mathcal{B}_{i+1}$'s anchor is to its right, we will find one good realization of $\mathcal{B}_1, \ldots, \mathcal{B}_{i+1}, \mathcal{L}_1, \ldots, \mathcal{L}_j$ of minimum height and width, and with $\mathcal{B}_{i+1}$ top-clear. To justify this, note that since the width of any good realization is fixed (namely at $p(\mathcal{B}_{i+1})$), we can simply minimize height, and fix one good realization. On the other hand, if $\mathcal{B}_{i+1}$'s anchor is to its left then we will find one good realization of $\mathcal{B}_1, \ldots, \mathcal{B}_i, \mathcal{L}_1, \ldots, \mathcal{L}_j$ of width no more than $p(\mathcal{B}_{i+1})$ and then of minimum height, together with a restricted range for the elastic rectangle $\mathcal{B}_{i+1}$. It makes sense to leave $\mathcal{B}_{i+1}$ elastic, since, as noted before, the realization we choose for $\mathcal{B}_{i+1}$ will affect the realizations we can choose for the next x-base or y-base rectangle we add. And on the other hand, it makes sense to fix realizations of all prior rectangles, since we now have a bound on the width, and can simply minimize height.

It is easy to make these justifications more formal, and prove by induction as the algorithm progresses:

Lemma 3. *The solution to subproblem $\mathcal{S}^T_{i+1,j}$ captures all minimal height-width combinations for good realizations of $\mathcal{B}_1, \ldots, \mathcal{B}_{i+1}, \mathcal{L}_1, \ldots, \mathcal{L}_j$ with $\mathcal{B}_{i+1}$ top-clear. (By minimal, we mean minimal in the ordering $(h, w) \leq (h', w')$ if $h \leq h'$ and $w \leq w'$.)*

Algorithm. Assume we have solved subproblems $\mathcal{S}^T_{i,j}$ and $\mathcal{S}^R_{i,j}$. This is how we solve subproblem $\mathcal{S}^T_{i+1,j}$. (We use a symmetric solution for $\mathcal{S}^R_{i,j+1}$.)

We need to add $\mathcal{B}_{i+1}$, top-clear. We will divide into two cases: either $p(\mathcal{B}_{i+1})$ is on the left of the elastic rectangle or on its right. In the first case the new rectangle will remain elastic in our solution; in the second case, because the width of the solution is fixed, we can simply minimize the height, and will fix B_{i+1}. See Figure 5.

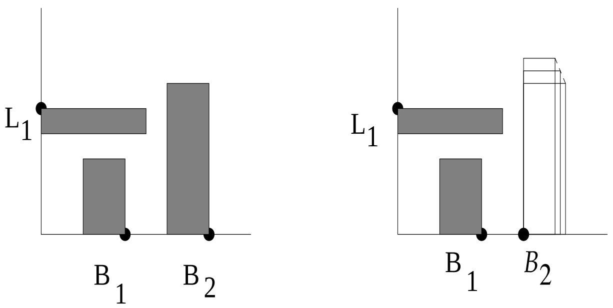

Fig. 5. Adding elastic rectangle $\mathcal{B}_2$ to a right-clear realization.

Case I. $p = p(\mathcal{B}_{i+1})$ is to the left. We may build on $\mathcal{S}^T_{i,j}$ or on $\mathcal{S}^R_{i,j}$. We will compare to see which is best. We need a solution to $\mathcal{S}_{i,j}$ with width no more than $x(p)$. From the stored solutions to each of $\mathcal{S}^T_{i,j}$ and $\mathcal{S}^R_{i,j}$, take a good realization with width less than or equal to $x(p)$ and with minimum height. Of these two solutions, keep the one of minimum height h. We now add $\mathcal{B}_{i+1}$, keeping it as an elastic rectangle, though with a height and width range that may be restricted. Since the height of any good realization must be at least h, we want the minimum width solution of height h, and all the possible trade-offs for solutions of height greater than h. Thus, if the maximum height of $\mathcal{B}_{i+1}$ is less than h, we fix B_{i+1} to the maximum possible height, and otherwise, we restrict the height range of B_{i+1} from h on up and take the corresponding width range.

Case II. $p(\mathcal{B}_{i+1})$ is to the right. The width of our solution is then fixed (namely at $p(\mathcal{B}_{i+1})$) and we are looking for the solution of minimum height. We will fix B_{i+1}. We may build on $\mathcal{S}^T_{i,j}$ or on $\mathcal{S}^R_{i,j}$. We will compare to see which provides us with the best—i.e. minimum height—solution.

Case 1. Building on $\mathcal{S}^T_{i,j}$. If we have a unique realization stored for this subproblem, then we simply add the minimum height realization for $\mathcal{B}_{i+1}$ that fits, i.e. whose left side is greater than or equal to the width of the subproblem's realization (Figure 6.a). Otherwise, $\mathcal{B}_i$ is still elastic in the solution to the subproblem. In this case we must find the minimum height good realization for two x-base elastic rectangles $\mathcal{B}_i$ and $\mathcal{B}_{i+1}$ with their anchors on the left and right respectively. See Figure 6.b. This involves intersecting two hyperbolas. The details of the computation can be found in the following section.

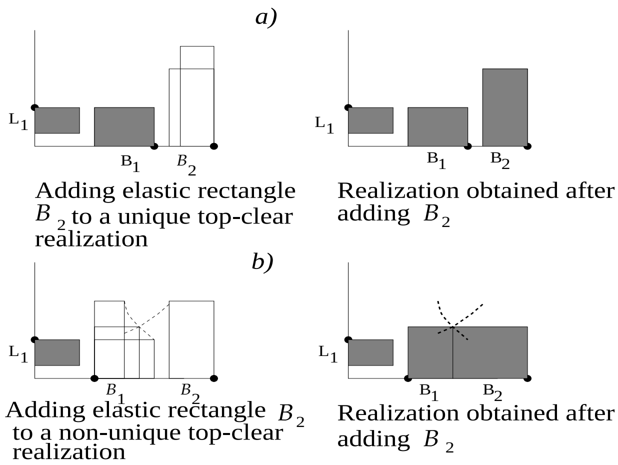

Fig. 6. Case I.1.

Case 2. Building on $\mathcal{S}^R_{i,j}$. If we have a unique realization stored for this subproblem, then we simply add the minimum height realization that fits, i.e. whose left side is greater than or equal to the width of the subproblem's realization (Figure 7.a). Otherwise, $\mathcal{L}_j$ is still elastic in the solution to the subproblem. In this case we must find the minimum height good realization for two elastic rectangles $\mathcal{L}_j$ and $\mathcal{B}_{i+1}$ with their anchors at the bottom left and bottom right respectively. See Figure 7.b. This is very similar to the problem we needed to solve in Case 1—the only difference is that the first elastic rectangle does not lie on the x-axis. The details of the computation can be found in the following section.

This completes the description of the the main step of the algorithm. We can start the algorithm off with the solutions to the subproblems $\mathcal{S}^T_{0,0}$ and $\mathcal{S}^R_{0,0}$ as rectangles of 0 width and height at the origin. When the algorithm has solved $\mathcal{S}^T_{n,m}$ and $\mathcal{S}^R_{n,m}$ we learn whether there is a solution to the whole problem.

It is easy to see that the algorithm takes $O(nm)$ time.

The correctness of the algorithm is fairly obvious based on two results: Theorem 2 which promises a clear decomposition, and justifies solving subproblem $\mathcal{S}^T_{i+1,j}$ by only looking at solutions to $\mathcal{S}^T_{i,j}$ and $\mathcal{S}^R_{i,j}$; and Lemma 3, which tells us about the form of the solution to subproblem $\mathcal{S}^T_{i+1,j}$.

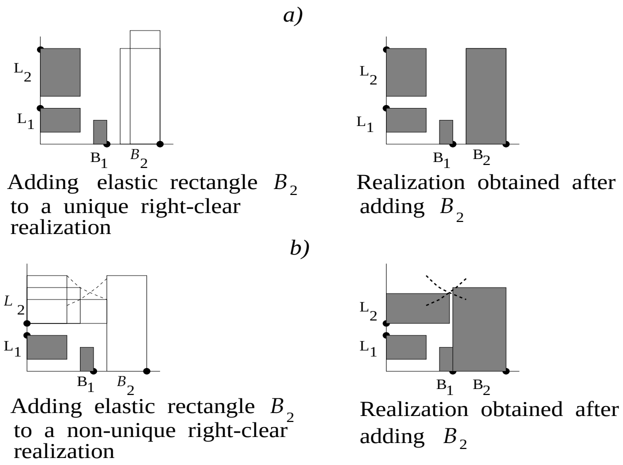

Fig. 7. Case I.2.

3.3 Hyperbolas

In this subsection we will explain the one missing detail of the algorithm: how to compute the minimum height good realization for two x-base elastic rectangles $\mathcal{B}_i$ and $\mathcal{B}_{i+1}$ with their anchors on the left and right respectively. This problem arises in case II, subcase 1 (Figure 6.b).

Consider the hyperbola segments $\Psi(\mathcal{B}_i)$ and $\Psi(\mathcal{B}_{i+1})$ described by the elastic rectangles. If the hyperbola segments intersect in a point c, then we take the rectangles B_i and B_{i+1} that have height equal to $y(c)$. (This is easy on a real-RAM model of computation.) The rectangles will touch each other. Otherwise the minimum height good realization—if it exists— is obtained at a "boundary value" using a rectangle from $\mathcal{B}_i$ or $\mathcal{B}_{i+1}$ (or both) of maximum or minimum possible height. We need to compare the 4 extreme solutions.

The slightly more general problem that arises in case II, subcase 2, where the right-hand elastic rectangle no longer lies on the x-axis, can be solved in exactly the same way.

4 NP-hardness of the One-corner Elastic Labeling Problem

The elastic labeling problem is NP-hard because the special case when we fix the height and width of each elastic rectangular label is precisely the point-feature-label placement problem which was shown to be NP-complete by [FWa91], [KIm88], [KRa92], [MSh91].

In this section we will prove the NP-hardness of the one-corner elastic labeling problem. Thus, either elasticity, or the choice of which corner of the label to use, results in an NP-hard problem. Our proof follows the idea of the proof given by

Formann and Wagner in [FWa91]. The 3-SAT problem is reduced to the labeling problem as follows.

For an instance of the 3-SAT problem we build a configuration of points with elastic rectangles as labels. For each variable we place a special point whose label is forced by the other points to have height in one of two subranges. The subrange that the height of this label takes indicates the setting of the variable. Let $[h^{\min}, h^{\max}]$ be the range of the height for such a label. We will choose an $h^{'} \in [h^{\min}, h^{\max}]$ so that if the label is forced to have height in subrange $[h^{\min}, h^{'}]$ the variable is set to true and otherwise it is set to false.

We will construct *variable gadgets*, *clause gadgets* and *pipes* connecting the variable gadget for x to the clause gadgets for clauses containing x or $\overline{x}$. Each gadget and each pipe is an arrangement of points each with an associated elastic label. The pipes propagate the truth-value chosen for the variable. When the variable is set to true [false] its value will be propagated by the pipes to the bottom [top] pipe connected to the variable gadget. The pipes that propagate the true [false] value are connected to the negated [unnegated] variables in the clause gadget. The idea is that the pipes transmit "flow" to the clauses. For each clause we construct a gadget that can support flow from at most two pipes. Since flow in the pipes corresponds to values that do not satisfy the clause, thus a clause admits flow from at most two pipes. If a clause gadget receives flow from all three pipes then the corresponding clause is not satisfied. So, there is a solution for a given instance of 3-SAT iff the constructed instance of the one-corner labeling problem has a solution.

Figures 8, 9 and 10 show the gadgets needed. A *crossing gadget* is a set of points that allows two pipes to cross, with flow properly transmitted by each pipe. Details of the construction and the proof that it works can be found in [ILu97].

5 Discussion

We are currently working on heuristics for the elastic labeling problem [Itu]. A main motivation for the elastic labeling problem is when labels may consist of multiple lines of text. This is formulated most naturally as a *discretized* version of the problem; in particular, the height of each rectangle should be an integer. Both our NP-hardness result for the one-corner problem, and the good algorithm for the two-axis problem carry over to the discretized versions [Itu].

As mentioned in the introduction, the solution to the two-axis labeling problem is the main ingredient needed in solving the more general and useful rectangle perimeter labeling problem. A paper on this is in progress.

We mention one open problem: find a good algorithm for the generalization of the two-axis labeling problem where, in addition to elasticity, we may choose which corner of the label to place at the anchor point, so long as the labels stay in the first quadrant.

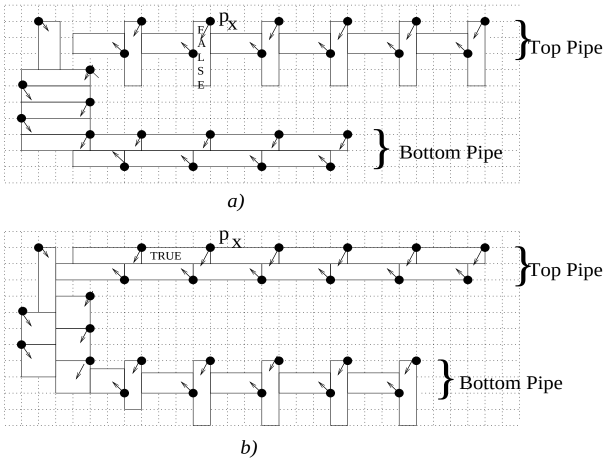

Fig. 8. a) Variable gadget with x set to true and flow transmitted through the top pipe. b) Variable gadget with x set to false and flow transmitted through the bottom pipe.

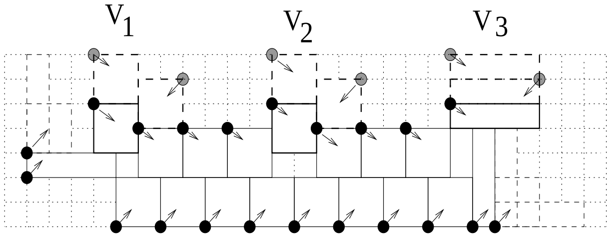

Fig. 9. Clause gadget with flow shown entering on pipes V_1 and V_2.

References

[CMS95] J. Christensen, J. Marks, and S. Shieber. An empirical study of algorithms for point feature label placement . *ACM Transactions on Graphics.* (1995) 14 (3), 203-232.

[DFr92] J.S. Doerschler and H. Freeman. A rule-based system for dense-map name placement. *Communications of the Association of Computing Machinery.* (1992) 35(1), 68-79.

[FWa91] M. Formann and F. Wagner. A packing problem with applications in lettering of maps. *Proceedings of the 7th ACM Symposium on Computational Geometry.* (1991) 281-288.

[Imh75] E. Imhof. Positioning names on maps. *The American Cartographer.* (1975) 2, 128-144.

[ILu97] C. Iturriaga and A. Lubiw. NP-hardness of some map labeling problems. *Technical Report CS-97-18.* University of Waterloo (1997).

[Itu] C. Iturriaga. Ph.D. Thesis, in progress. University of Waterloo.

[KIm88] T. Kato and H. Imai. The NP-completeness of the character placement prob-

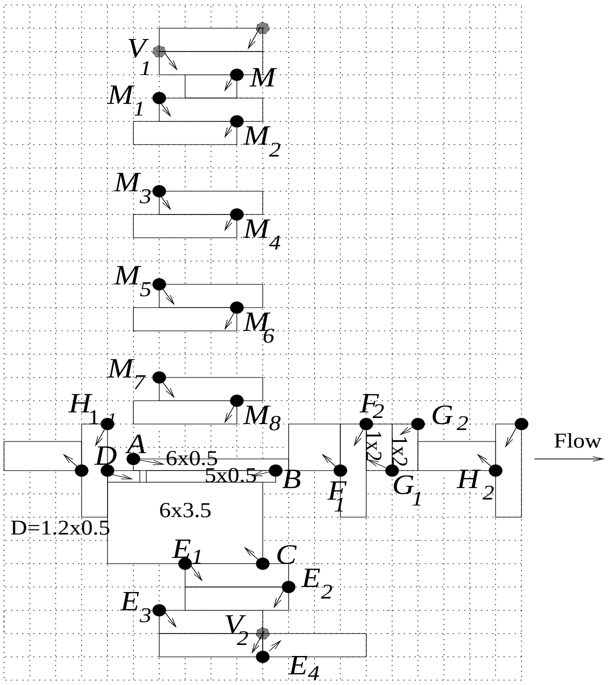

Fig. 10. A Crossing gadget.

lem of 2 or 3 degrees of freedom. *Record of Joint Conference of Electrical and Electronic engineers in Kyushu.* (1988), 1138. In Japanese.

[KRa92] D. Knuth and A. Raghunathan. The problem of compatible representatives. *SIAM Disc. Math.* (1992) 5 (3), 422-427.

[MSh91] J. Marks and S. Shieber. The computational complexity of cartographic label placement. *Technical Report CRCT-05-91.* Harvard University (1991).

[WWo97] F. Wagner and A. Wolff. A practical map labeling algorithm. *Computational Geometry: Theory and Applications.* (1997) 387-404.

[Yoe72] P. Yoeli. The logic of automated map lettering. *The Cartographic Journal.* (1972) 9, 99-108.