

# Combining Graph Labeling and Compaction<sup>\*</sup>

## (Extended Abstract)

Gunnar W. Klau and Petra Mutzel

Max-Planck-Institut für Informatik  
Im Stadtwald, D-66123 Saarbrücken, Germany  
{guwek,mutzel}@mpi-sb.mpg.de

**Abstract.** Combinations of graph drawing and map labeling problems yield challenging mathematical problems and have direct applications, *e.g.*, in automation engineering. We call graph drawing problems in which subsets of vertices and edges need to be labeled *graph labeling problems*. Unlike in map labeling where the position of the objects is specified in the input, the coordinates of vertices and edges in a graph labeling problem instance have yet to be determined and thus create additional degrees of freedom. We concentrate on the *Compaction and Labeling (COLA) Problem*: Given an orthogonal representation—as produced by algorithms within the topology–shape–metrics paradigm—and some label information, the task is to generate a labeled orthogonal embedding with minimum total edge length. We characterize feasible solutions of the *COLA* problem extending an existing framework for solving pure compaction problems. Based on the graph-theoretical characterization, we present a branch-and-cut algorithm which computes optimally labeled orthogonal drawings for given instances of the *COLA* problem. First computational experiments on a benchmark set of practical instances show that our method is superior to the traditional approach of applying map labeling algorithms to graph drawings. To our knowledge, this is the first algorithm especially designed to solve graph labeling problems.

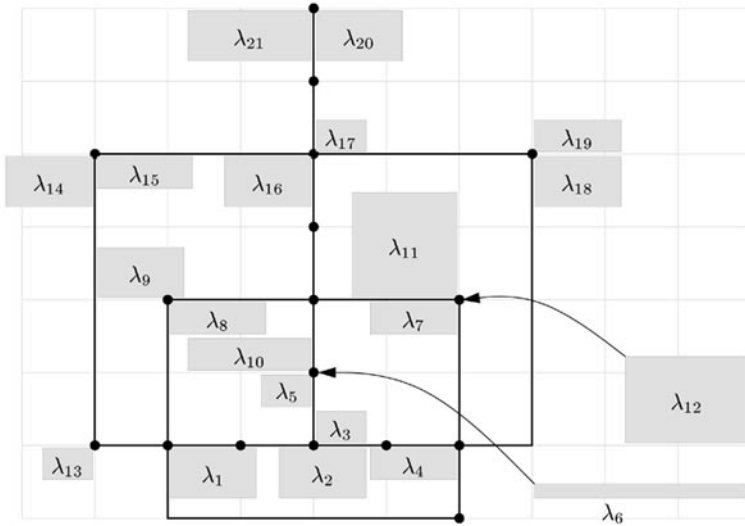
## 1 Introduction

The area of graph drawing provides an ample variety of algorithms that produce aesthetically pleasing layouts of graphs [DETT99]. At the same time, there are sophisticated techniques to generate provably good labelings of cartographic maps [WS]. Combinations of problems from both areas yield challenging mathematical problems and have direct applications, *e.g.*, the automatic layout of state diagrams in automation engineering or the drawing of schematic maps such as subway maps. Unlike in map labeling where the position of the objects is specified in the input, the coordinates of vertices and edges in a graph drawing problem instance have yet to be determined and thus create additional degrees of freedom.

---

<sup>\*</sup> This work is partially supported by the Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (No. 03-MU7MP1-4).

We call the class of graph drawing problems where subsets of vertices and edges have to be labeled *graph labeling problems*. So far, problems from this class have only been attacked by applying map labeling algorithms to graph drawings [KT97,IL97]. Figure 1 shows a labeled drawing for a state diagram; first the underlying graph has been drawn, in a last step the labels have been placed with an optimal labeling strategy. This forces some labels to either overlap parts of the drawing or to be placed far away from the vertices to which they should be attached. Another approach could be to model each label as a vertex of fixed size and then apply a graph drawing algorithm which can cope with vertices of different sizes. This treatment increases the vertex degrees and may destroy existing structural properties in the graph—for the drawing of state diagrams this leads to a loss of orthogonality and unacceptable area consumption.



**Fig. 1.** A state diagram; first drawn then labeled

In this paper we concentrate on a special graph labeling problem: the *Compaction and Labeling (COLA) Problem*. Graph drawing algorithms in the *topology-shape-metrics paradigm* [DETT99] produce in an intermediate step a so-called *orthogonal representation*  $H$ , i.e., an extended planar representation. In addition to the topological information,  $H$  contains for each edge information about the order of bends and the angle formed with the following edge in the appropriate face.

In the *compaction* phase of the paradigm, the goal is to minimize the area or the total edge length of the resulting orthogonal drawing. The *COLA* problem unifies the problem to find a drawing respecting the given shape and an additional labeling task: A set of labels, each of which is fixed in size, is assigned to every vertex  $v$ . Each of the labels must be placed so that it touches  $v$  in at

least one point and does not overlap other objects. We are given a set of  $k$  labels  $L = \{\lambda_1, \lambda_2, \dots, \lambda_k\}$ , two functions  $w : L \rightarrow \mathbb{Q}$ ,  $h : L \rightarrow \mathbb{Q}$  determining the width and height of each label, and a third function  $a : L \rightarrow V$  denoting the vertex  $a(\lambda)$  to which the label  $\lambda$  should be attached. We are looking for a labeled orthogonal embedding with short edges in which each label is placed correctly, and no labels overlap the drawing.

In this paper we give a characterization of feasible solutions for the *COLA* problem. We extend an existing framework for two-dimensional compaction problems in graph drawing [KM99]. Based on the graph-theoretical characterization, we present a first branch-and-cut based algorithm prototype which computes optimally labeled orthogonal drawings for given instances of the *COLA* problem. To our knowledge, this is the first algorithm especially designed to solve graph labeling problems.

Section 2 introduces the graph-theoretical framework. We give a formal definition of the *COLA* problem in Sect. 3 and show how to integrate the additional labeling problem into the framework. In Sect. 4 we extend an optimal branch-and-cut algorithm so that it solves the *COLA* problem to optimality and give some first computational results. Finally, we conclude with Sect. 5. Due to space limitations we can only give sketches of the proofs.

## 2 Constraint Graphs

In this section we introduce the graph-theoretical framework we have developed to solve the *COLA* problem. An algorithm within the topology-shape-metrics approach [DETT99] produces at the end of the second phase an orthogonal representation  $H$  which contains the information about the topology and the shape of the drawing. We call  $H$  *simple* if its number of bends is zero. Every orthogonal representation can be transformed into a simple one by just replacing each bend with an artificial vertex.

In order to compute a drawing for  $H$ , we must assign coordinates to the vertices and bends. In this paper, we concentrate on *pure orthogonal embeddings*, but our ideas can be adapted to other drawing standards such as *Kandinsky-like embeddings* (introduced in [FK96]) or *orthogonal box embeddings* (subclasses are the *big node model* from [FK97] and the *TSS model* from [BMT97], a related class is the *quasi-orthogonal model* from [KM98]).

*Pure orthogonal embeddings* are only admissible for 4-planar graphs. Each vertex and bend is mapped to a point and edge segments are mapped to horizontal or vertical non-crossing line segments of some minimum length connecting the images of their endpoints. A special case are pure orthogonal grid embeddings as defined in [Tam87]: Here, vertices and bends have integer coordinates. As with representations, we call orthogonal embeddings simple if they do not contain bends.

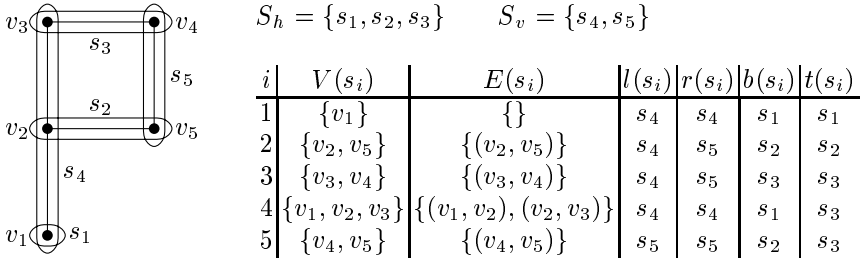
We first focus on the following subproblem of the *COLA* problem: Given an orthogonal representation  $H$ , produce an orthogonal embedding  $\Gamma$  for  $H$

which is—among all such drawings—of minimum total edge length. Recently, Patrignani proved the NP-hardness of this problem [Pat99].

Initial observation leads to the following notion: Incident edges of same direction can be combined, forming the objects to compact. Let  $\Gamma$  be a simple orthogonal drawing of a graph. It induces a partition of the set of edges  $E$  into the horizontal set  $E_h$  and the vertical set  $E_v$ . A horizontal (resp. vertical) *subsegment* in  $\Gamma$  is a connected component in  $(V, E_h)$  (resp.  $(V, E_v)$ ). If the component is maximally connected it is also referred to as a *segment*.

The following observations are crucial (see also Fig. 2).

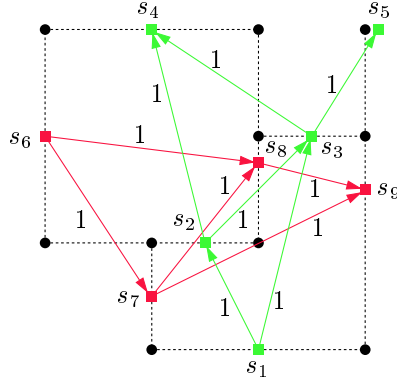
- Each edge is a subsegment.
- Each vertex  $v$  belongs to one unique horizontal and one unique vertical segment, denoted by  $\text{hor}(v)$  and  $\text{vert}(v)$ .
- The *limits* of a subsegment  $s$  are given as follows: Let  $v_l, v_r, v_b$ , and  $v_t$  be the leftmost, rightmost, bottommost, and topmost vertices on  $s$ . Then  $l(s) = \text{vert}(v_l)$ ,  $r(s) = \text{vert}(v_r)$ ,  $b(s) = \text{hor}(v_b)$ , and  $t(s) = \text{hor}(v_t)$ .



**Fig. 2.** Segments of a simple orthogonal grid drawing and its limits

We provide a necessary and sufficient condition for all feasible solutions of a given instance of the compaction problem. This condition is based on existing paths in the so-called *constraint graphs*. This pair of directed graphs is similar to the *layout graphs* known from one-dimensional compaction strategies in VLSI design [Len90]. Nodes in these graphs represent the segments; their weighted arcs characterize relative positioning relations between the segments. We denote by  $c_i$  the coordinate of segment  $s_i$ . A weight  $\omega_{ij} \in \mathbb{Q}$  for an arc  $(s_i, s_j)$  indicates that the coordinate difference  $c_j - c_i$  must be at least  $\omega_{ij}$ . Figure 3 shows an example for a pair of constraint graphs. The arcs specify exactly the relative relationships known from the shape of the graph: We call such special pairs of constraint graphs  $\langle (S_v, A_h), (S_h, A_v) \rangle$  *shape graphs* or a *shape description*. Note that each horizontal edge in the original graph defines a relative positioning constraint between two vertical segments. Similarly, vertical edges determine constraints between horizontal segments.

The characterization of feasible solutions for the two-dimensional compaction problem is based on the following three observations:



**Fig. 3.** An orthogonal grid embedding (dotted) and a pair of constraint graphs

1. The arcs of the shape graphs are contained in every pair of constraint graphs corresponding to a drawing reflecting the given shape.
2. Generally, the information in the shape graphs is not sufficient to produce an orthogonal embedding. If this is the case, however, we call such a pair of constraint graphs *complete*.
3. There are in general many possibilities for extending the shape graphs to a complete pair of constraint graphs.

We denote by  $s_i \xrightarrow{+} s_j$  a path of non-negative weight between  $s_i$  and  $s_j$ . The following is a precise characterization for complete pairs of constraint graphs in terms of paths that must be contained in the arc sets: A pair of graphs is complete if and only if both arc sets do not contain non-negative cycles and for every pair of segments  $(s_i, s_j) \in S \times S$  one of the following four conditions holds:

- |                                    |                                    |
|------------------------------------|------------------------------------|
| 1. $r(s_i) \xrightarrow{+} l(s_j)$ | 3. $t(s_j) \xrightarrow{+} b(s_i)$ |
| 2. $r(s_j) \xrightarrow{+} l(s_i)$ | 4. $t(s_i) \xrightarrow{+} b(s_j)$ |

If one of the non-negative paths in (1) exists we also call the pair of segments  $(s_i, s_j)$  *separated*. Note that this definition generalizes the notion of completeness as introduced for the pure compaction problem in [KM99]. There, we have the special case that  $\omega_{ij} = 1$  for all  $(s_i, s_j)$ .

We can now express a one-to-one correspondence between these complete extensions and orthogonal embeddings. On the basis of the following theorem, the two-dimensional compaction task can be seen as the search for a complete extension of the given shape graphs leading to minimum total edge length.

**Theorem 1 ([KM99]).** *For each simple orthogonal embedding with shape description  $\sigma = \langle (S_v, A_h), (S_h, A_v) \rangle$  there exists a complete extension  $\tau = \langle (S_v, B_h), (S_h, B_v) \rangle$  of  $\sigma$  and vice versa: Every complete extension  $\tau = \langle (S_v, B_h),$*

$(S_h, B_v)\rangle$  of a shape description  $\sigma = \langle (S_v, A_h), (S_h, A_v)\rangle$  corresponds to a simple orthogonal embedding with shape description  $\sigma$ .

### 3 Modeling the Labels

In this section we extend the graph-theoretical formulation of the pure two-dimensional compaction problem so that a feasible solution corresponds to a labeled orthogonal embedding and thus to a solution of the *COLA* problem. We define the characteristics of a solution for the *COLA* problem in order to state the *COLA* problem formally.

**Definition 1.** A labeled orthogonal embedding  $\Gamma_L$  for an orthogonal representation  $H$  of a planar graph  $G = (V, E)$  and a label set  $L$  with functions  $w, h : L \rightarrow \mathbb{Q}$  and  $a : L \rightarrow V$  fulfills the following properties:

1.  $\Gamma_L$  is an orthogonal embedding for  $H$ .
2. A label  $\lambda \in L$  has size  $w(\lambda) \times h(\lambda)$  and its image intersects with the image of vertex  $a(\lambda)$  at at least one point.
3. A label  $\lambda \in L$  does neither overlap nor include other objects.

**Definition 2.** Compaction and Labeling (*COLA*) problem.

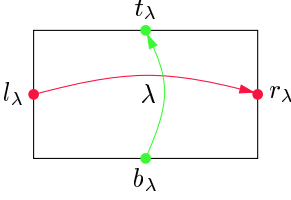
Given an orthogonal representation  $H$  for a planar graph  $G = (V, E)$ , a label set  $L$  with functions  $w, h : L \rightarrow \mathbb{Q}$ , and  $a : L \rightarrow V$ , find a labeled orthogonal embedding for  $H$  with minimum total edge length.

We model each label  $\lambda$  as a rectangle bounded by the segments  $l_\lambda, r_\lambda, t_\lambda$ , and  $b_\lambda$  as in Fig. 4. Its limits are linked by two arcs  $(l_\lambda, r_\lambda) \in A_h$  and  $(b_\lambda, t_\lambda) \in A_v$  of weight  $w(\lambda)$  and  $h(\lambda)$ , respectively. Unlike the arcs from the previous section, the minimum distance requirements for this type of arcs are determined by the width and height of the corresponding label. Observe that we use the same sort of segments we have introduced in the previous section for the solution of the compaction problem. At this point we introduce a set  $O$  containing the objects to compact. It consists of the segments corresponding to consecutive edges of same direction and all the labels. This gives us an important property free of charge: If we can ensure that—as in Sect. 2—all pairs of distinct objects in  $O$  are separated, the labels will neither overlap nor will they be crossed by edges of the graph. We will achieve this by giving a more general definition of completeness.

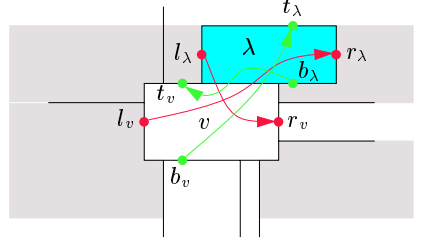
We additionally have to guarantee that each label is drawn in the neighborhood of the vertex to which it belongs. Figure 5 shows the possibilities we want to capture with our formulation. According to Def. 1 a label can be placed anywhere close to a vertex, i.e., it must touch the vertex at at least one point.

The relative positioning of label  $\lambda$  to its vertex  $v = a(\lambda)$  can be equivalently expressed by the following four conditions:

1. The left side of  $\lambda$  must not lie to the right side of  $v$ .
2. The right side of  $\lambda$  must not lie to the left side of  $v$ .



**Fig. 4.** Four segments model a label



**Fig. 5.** Possibilities for placing a label  $\lambda$  in the neighborhood of  $a(\lambda)$

3. The bottom side of  $\lambda$  must not lie above the top side of  $v$ .
4. The top side of  $\lambda$  must not lie below the bottom side of  $v$ .

In our graph-theoretical framework this translates into four special arcs of zero weight that have to be included in the pair of constraint graphs (see Fig. 5). For a label  $\lambda$  with  $a(\lambda) = v$  these are  $(l_\lambda, r_v)$ ,  $(l_v, r_\lambda)$ ,  $(b_\lambda, t_v)$ , and  $(b_v, t_\lambda)$ , linking the limits of the label to the limits of the vertex. Note that we model vertices whose images are boxes of non-zero size in a similar way as the labels by including them in  $O$  and linking them appropriately to the segments that correspond to incident edges.

Having introduced these new types of arcs, we can extend the notion of *shape description* as defined in Sect. 2 to integrate the labels in the framework.

**Definition 3.** A labeled shape description  $\sigma_L$  of an orthogonal representation  $H$  with label information  $L, w, h$ , and  $a$  is a tuple  $\langle (S_v \cup S_{L_v}, A_h \cup A_{L_h}), (S_h \cup S_{L_h}, A_v \cup A_{L_v}) \rangle$  where  $\langle (S_v, A_h), (S_h, A_v) \rangle$  is a shape description for  $H$  and

$$\begin{aligned} S_{L_v} &= \bigcup_{\lambda \in L} \{l_\lambda, r_\lambda\}, & A_{L_h} &= \bigcup_{\lambda \in L} \{(l_\lambda, r_\lambda), (l_\lambda, r_{a(\lambda)}), (l_{a(\lambda)}, r_\lambda)\}, \\ S_{L_h} &= \bigcup_{\lambda \in L} \{b_\lambda, t_\lambda\}, & A_{L_v} &= \bigcup_{\lambda \in L} \{(b_\lambda, t_\lambda), (b_\lambda, t_{a(\lambda)}), (b_{a(\lambda)}, t_\lambda)\}. \end{aligned}$$

Clearly, each instance of the *COLA* problem uniquely determines a labeled shape description. We can now give a more general formulation of completeness by considering all objects in the set  $O$ : A pair of labeled constraint graphs is complete if and only if it does not contain a non-negative cycle and each distinct pair of objects is separated. This generalization leads to the following main result.

**Theorem 2.** For each simple labeled orthogonal embedding with shape description  $\sigma = \langle (S_v, A_h), (S_h, A_v) \rangle$  and label information  $L, w, h, a$  there exists a complete labeled extension  $\tau = \langle (S_v \cup S_{L_v}, A_h \cup A_{L_h}), (S_h \cup S_{L_h}, A_v \cup A_{L_h}) \rangle$  of  $\sigma$  and vice versa: Every complete labeled extension  $\tau = \langle (S_v \cup S_{L_v}, B_h \cup A_{L_h}), (S_h \cup S_{L_h}, B_v \cup A_{L_h}) \rangle$  of a labeled shape description  $\sigma = \langle (S_v \cup S_{L_v}, A_h \cup A_{L_h}), (S_h \cup S_{L_h}, A_v \cup A_{L_h}) \rangle$  corresponds to a simple labeled orthogonal embedding with shape description  $\sigma$ .

*Proof (Sketch).* The proof is similar to the one in [KM99, Theorem 1], so we give only a sketch of it: For the forward direction, we construct a complete extension of the given labeled shape description  $\sigma$  by inserting the missing information according to the visibility properties in the drawing. Clearly, this excludes non-negative cycles. The separation properties are also fulfilled for all segments because otherwise there would be overlapping objects in the drawing, which is not the case.

For the other direction, we give a constructive proof of how to build a labeled orthogonal embedding with a given complete extension of a labeled shape description: As in the proof for the pure compaction setting, we make use of a *length assignment* to assign the coordinates to the segments—and thus to the vertices, bends, and labels. Again, we are able to show that the resulting drawing is a labeled orthogonal embedding reflecting the shape of  $H$  and the labeling information given in  $L, w, h$ , and  $a$ .

## 4 Optimal Graph Labeling

In this section we introduce our the ILP formulation to solve the *COLA* problem to optimality and present first computational results.

One way to characterize the set of complete extensions is by means of an integer linear program. We introduce a binary variable  $x_{ij}$  for each arc  $(s_i, s_j)$  that might be in some extension of the given labeled shape description  $\sigma_L = \langle (S_v, A_h), (S_h, A_v) \rangle$ . If  $(s_i, s_j)$  is contained in the extension, the corresponding variable  $x_{ij}$  is one, otherwise zero. We refer by  $X$  to the set of binary variables. Additionally, there is a variable  $c_s \in \mathbb{Q}$  for each segment  $s \in S$  denoting the coordinate of  $s$ . The ILP then reads as follows:

$$\min \sum_{e \in E_h} c_{r(e)} - c_{l(e)} + \sum_{e \in E_v} c_{t(e)} - c_{b(e)} \quad \text{subject to} \quad (2)$$

$$x_{r_o, l_p} + x_{r_p, l_o} + x_{t_o, b_p} + x_{t_p, b_o} \geq 1 \quad \forall (o, p) \in O \times O, o \neq p \quad (2.1)$$

$$c_j - c_i \geq \omega_{ij} \quad \forall (s_i, s_j) \in A_h \cup A_v \quad (2.2)$$

$$c_j - c_i - (M + \omega_{ij})x_{ij} \geq -M \quad \forall x_{ij} \in X \quad (2.3)$$

$$x_{ij} \in \{0, 1\} \quad \forall x_{ij} \in X \quad (2.4)$$

Inequalities (2.1) model the characterization of separation, i.e., the existence of necessary paths between distinct objects in an extension. Inequalities (2.2) force the coordinates to obey the distance rules coded by the weighted arcs in the underlying labeled shape description; the value  $\omega_{ij}$  denotes the minimum distance between  $s_i$  and  $s_j$ . The same must hold true for the potential additional arcs: Whenever a variable  $x_{ij}$  is one, we want an inequality of type (2.2), otherwise there should be no restriction on the coordinate variables. This situation is modeled by inequalities (2.3) with the help of a big constant  $M$ . We have shown in [KM99] that inequalities (2.2) and (2.3) ensure the absence of non-negative cycles in the extension.



Note that we do not introduce additional decision variables to integrate the labeling task; the choice of where to place a label is performed by the separation properties. A label can be placed wherever it does not interfere with other objects as long as it stays in the neighborhood of the vertex to which it belongs. Moreover, we do not restrict the label placement to a finite number of prescribed places. Of course, the number of possibly non-separated segments is much higher than in an instance of a pure compaction problem.

Like the one-to-one correspondence between complete extensions and orthogonal embeddings there is a one-to-one correspondence between feasible solutions of the ILP and complete extensions of the given labeled shape graphs:

**Theorem 3.** *For each feasible solution  $(x, c)$  of the ILP for a given labeled shape description  $\sigma_L$  there is a labeled orthogonal embedding with appropriate shape and label information and vice versa.*

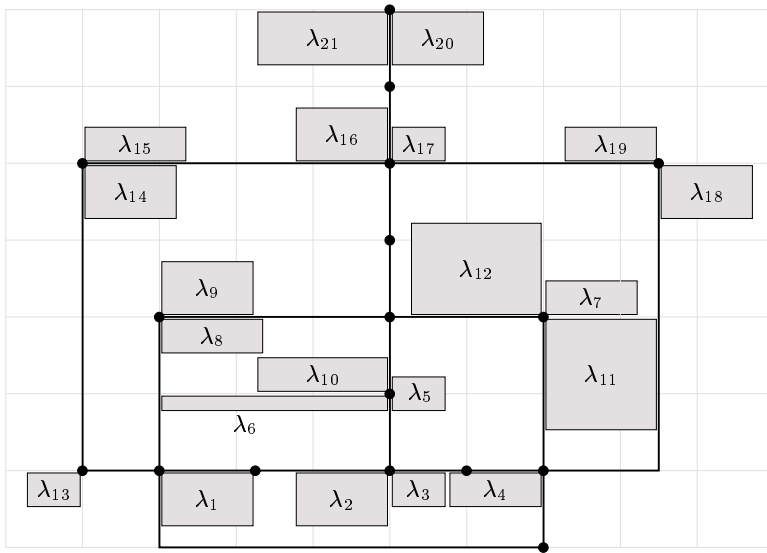
*Proof (Sketch).* A solution of the ILP corresponds to a complete extension of the labeled shape description  $\sigma$  with appropriate length assignment. The forward direction follows with Theorem 1. Conversely, we can use the information of a labeled embedding to construct a feasible solution of the ILP. We can show that none of the inequalities is violated and that the value of the objective function equals the total edge length.

We have extended the existing branch-and-cut framework for the pure compaction problem to solve instances of the *COLA* problem and have developed a prototype of an optimal compaction and labeling algorithm. We have tested the algorithm with real-world instances from the area of automation engineering and our first computational experiments indicate that we are able to draw and label the state diagrams within moderate computation time. Figure 6 shows the output of the optimal algorithm for the example of Fig. 1. It took the algorithm prototype 29 seconds to compute the optimal solution.

## 5 Conclusions and Future Plans

We have introduced a combined compaction and labeling problem arising from a practical application in automation engineering. The task is to simultaneously assign consistent edge lengths for a given orthogonal representation which determines the shape of the drawing and to label subsets of vertices. The resulting drawing should have small total edge length and all labels should be placed at the corresponding vertices so that they do not overlap other objects. Already the pure compaction task is NP-hard.

We have integrated the labeling problem into an existing graph-theoretical framework for solving two-dimensional compaction problems and have introduced the notion of labeled orthogonal embedding. Inside this framework, we have devised a branch-and-cut algorithm that produces optimal solutions for the *COLA* problem. Note that we do not require a label to be placed on a finite



**Fig. 6.** The example from Fig. 1, drawn with the optimal algorithm

set of prescribed places but allow for a continuous movement of the label around the vertex to which it is attached.

In the near future, we will provide the implementation of the branch-and-cut algorithm as a module inside the AGD library [AGD99]. We will extensively test the optimal algorithm on real world and randomly generated instances. We will try to develop heuristics in order to apply them at each node in the branch-and-cut tree and want to investigate a possible extension to edge labeling problems. Furthermore, we plan to extend our framework so that it can accomplish some postprocessing tasks like improving the number of bends or crossings and decreasing the sizes of vertices.

## References

- AGD99. AGD. *AGD User Manual*. Max-Planck-Institut Saarbrücken, Universität Halle, Universität Köln, 1999. <http://www.mpi-sb.mpg.de/AGD>.
- BMT97. T. Biedl, B. Madden, and I. Tollis. The three-phase method: A unified approach to orthogonal graph drawing. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 391–402. Springer-Verlag, 1997.
- DETT99. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing*. Prentice Hall, 1999.
- FK96. U. Fößmeier and M. Kaufmann. Drawing high degree graphs with low bend numbers. In F. J. Brandenburg, editor, *Graph Drawing (Proc. GD '95)*, volume 1027 of *Lecture Notes in Computer Science*, pages 254–266. Springer-Verlag, 1996.

- FK97. U. Fößmeier and M. Kaufmann. Algorithms and area bounds for nonplanar orthogonal drawings. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 134–145. Springer-Verlag, 1997.
- IL97. C. Iturriaga and A. Lubiw. Elastic labels: The two-axis case. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 181–192, 1997.
- KM98. G. W. Klau and P. Mutzel. Quasi-orthogonal drawing of planar graphs. Technical Report MPI-I-98-1-013, Max-Planck-Institut für Informatik, Saarbrücken, May 1998.
- KM99. G. W. Klau and P. Mutzel. Optimal compaction of orthogonal grid drawings. In G. P. Cornuéjols, editor, *Integer Programming and Combinatorial Optimization (IPCO '99)*, number 1610 in Springer Lecture Notes in Computer Science, pages 304–319, 1999.
- KT97. K. G. Kakoulis and I. G. Tollis. An algorithm for labeling edges of hierarchical drawings. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 169–180, 1997.
- Len90. T. Lengauer. *Combinatorial Algorithms for Integrated Circuit Layout*. John Wiley & Sons, New York, 1990.
- Pat99. M. Patrignani. On the complexity of orthogonal compaction. Technical Report RT-DIA-39-99, Dipartimento di Informatica e Automazione, Università degli Studi di Roma Tre, January 1999.
- Tam87. R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, 1987.
- WS. A. Wolff and T. Strijk. The map labeling bibliography. <http://www.inf.fu-berlin.de/map-labeling/bibliography>.