Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

NADINE ABBOUD   MARTIN GRÖTSCHEL   THORSTEN KOCH

Mathematical Methods for Physical Layout of Printed Circuit Boards: An Overview

Work supported the DFG Research Center *Mathematics for key technologies: Modelling, simulation, and optimization of real-world processes* (MATHEON)

ZIB-Report 06-29 (Dec 2006)

Abstract

This article surveys mathematical models and methods used for the physical layout of printed circuit boards, in particular component placement and wire routing. The main concepts are briefly described together with relevant references.

Mathematics Subject Classification (2000): 90C90 90-02 68R05

1 Introduction

Printed circuit boards (PCBs), see Figures 2 and 3 for an example, are ubiquituous. PCBs are the backbones of almost every electronic device, and therefore, PCB design and manufacturing are extremly important components of many industrial production processes. Before a PCB can serve its task it evolves through three main steps. The first one is the logic design, which defines the components to be used and their interconnections. The second step is the physical layout of the PCB where the geometric positions of the components and their physical connections are decided. The final step is the industrial production of the PCB.

In this article we give an overview on the mathematical models and methods used in the second step, the physical design of the board. The two major issues here are *component placement* and *wire routing* as depicted in Figure 1.

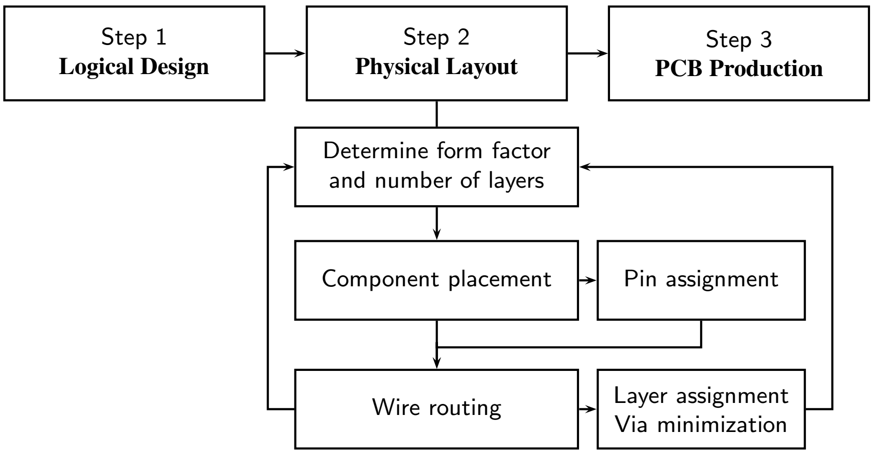

Figure 1: Tasks in PCB design

A digital logic circuit consists of a collection of interconnected parts. The parts come in two varities; active parts, i. e., integrated circuits (IC) and passive parts, such as resistors, and capacitors (Figure 2).

Due to continuously shrinking feature sizes, the functionality of ICs per area unit has significantly increased over the years. This has led to a steady increase of the ratio between passive and active parts of a PCB. Ratios of 20 or 30 passive to one active part are not uncommon. Each part can be represented by the shape and area occupied on the board together with the locations of one or more *pins* or *pads* where the electrical connections are made (Figure 3). One IC may have up to several hundred pins. To have a single term we will call the pins and pads of the components *terminals*. A set of terminals that have to be electrically connected is called a *net*. In PCB and chip design it is implicitly understood that the electrical connection has to be realized by a tree (on a routing graph to be described later). This standard definition of a net just specifies the nodes that have to be connected and leaves open how the actual wiring is done. Two different nets (i. e., their physical realization) must be insulated from each other.

A circuit board is built as a stack of layer pairs. Each pair starts as an insulating sheet with copper deposited on one or both sides. The copper sides of the sheet are first etched with different wiring patterns. Then the sheets are stacked into a sandwich separated by insulating material. Small holes are drilled into the board. Finally, the holes are plated with metal, so that electrical contact is made with each layer that has copper left at hole locations. Hence, a hole can form a conductive path between two or more layers. The standard term used in PCB and chip layout for a hole is *via*.

Parts can be attached to boards in two ways. The classical method is to solder the pins of the components into a via pattern on the board. The other possibility is the utilization of *surface mount devices*, which are glued to the board. Little connection pads on the devices make contact with their counterparts on the board without penetrating it. Note that the drill holes for vias that are only used to make connections between layers can be much smaller than those holes in which pins must be inserted.

The first task of the physical design is to decide on the size of the board and on the number of layers. In practice the form factor of the board is often predetermined or at least limited by the kind of device to be built, such as a PCI slot card, for example. Hom and Granacki (1996) describe a statistics based model to estimate the number of routing layers and total wire length for a printed circuit board.

The problems faced in the design of VLSI circuits are in many aspects similar to those encountered on PCBs. Again components (cells) are to be placed and have to be connected. Many algorithms can be applied in both cases. In fact, there are a lot more publications on the VLSI aspects of these problems than on their PCB counterparts. On the other hand, many technical constraints in PCB design are different from those in VLSI design. In this article, we focus on PCB based publications where possible. For an introduction into the specifics of VLSI design and a precise definition of what is meant by this, see, e. g., Gerez (1998), Sait and Youssef (1999), Vygen (2001).

Most of the articles dealing explicitly with PCB design describe a PCB design issue and a heuristic for its solution. The utilization of heuristics is not surprising, as the problems encountered in physical PCB layout are usually $\mathcal{NP}$-hard in theory and computationally expensive in practice. At this point, one particular feature of PCBs should be noted: The vast majority of the nets to be routed between components has only two terminals. This property is an important reason why many of the heuristics used for routing (and described later) work quite well in practice despite their simplicity.

Regarding theoretical aspects there seems to be only limited progress in recent years. On the practical side, several commercial placement and routing software systems are purchasable. Assessing their quality is very difficult, as there are no generally accepted benchmarks for PCB placement and routing publicly available. (The situation is somewhat better in the area of VLSI design (Whitaker, 1996, Harlow, 2000).) PCB layout involves a large amount of technical and other constraints (collectively called *design rules*), which are specified and handled differently in each program, making any comparision difficult.

In the next section we will describe shortly the relation between placement and routing, then an overview on placement methods is presented in Section 3 and several routing models are presented in Section 4.

2 Placement and Routing: General Remarks

The task in placement is to position all components on the board in a way that is feasible with respect to the limitations of the production process. After that all nets, i. e., all connections between the pins of the components according to the logical design, have to be routed on the board. Obviously, the placement has tremendous repercussions on the routing. Since the conditions on the placement itself are usually relatively simple, e. g., components are not allowed to overlap and have to adhere to some minimum distances, the main objective in placement is to allow a good and feasible routing. In principle it would be best to perform placement and routing in one step. But due to the difficulties involved, both tasks are typically done sequentially in practice (Mo et al., 2001, Kumar et al., 2005).

As said above, since finding a physically feasible placement is generally easy, the main objective of the placement is to permit a high quality routing. However, it is difficult to define "high quality" in a somewhat precise mathematical sense. Instead, substitute objectives are defined. A widely used objective in practice is to minimize the total wire length of all connections.

Unfortunately, the exact wire length of each net is not known until the nets are actually routed. Since even the computation of the minimum length of a single net is in general $\mathcal{NP}$-hard, as it corresponds to computing the length of a minimum Steiner tree, the total wire length has to be estimated. This is usually done by summing up an estimate of the length of each individual net. Methods to estimate the length of a specific net include: heuristically computing a Steiner tree between the terminals, computing a minimum spanning tree between the terminals, building a chain through the terminals, or taking half the length of the perimeter of the bounding box enclosing all the terminals of the net. Keep in mind, that for nets with two terminals and using Manhattan distances (a typical metric in PCB and VLSI design) all these measures give the same result. A survey on placement methods and substitute objectives can be found in Preas and Karger (1986).

Another possible objective during placement is to minimize the number of wire crossings. According to Stallmann et al. (2001) this can be modeled as a *bigraph crossing problem* in the following way: Let $G = (V, E)$ be a bipartite graph with partitions V_1 and V_2 and let G be embedded in the plane so that the nodes in V_i occupy distinct positions on the line $y = i$ and the edges are straight lines. For an embedding $f(G)$ of G in the plane, the *crossing number* $C_f(G)$ is the number of line intersections induced by f. This number depends only on the permutation of V_i along $y = i$ and not on specific x-coordinates. The *bigraph crossing number* $C(G)$ is defined as $C(G) = \min_f C_f(G)$. The computation of $C(G)$ is $\mathcal{NP}$-hard (Garey and Johnson, 1979). Some heuristics for crossing minimization are presented and compared in Stallmann et al. (2001).

3 Placement Methods

As the physical restrictions on component placement are few, every method, including repeated trials of random placements (Hanan et al., 1976a,b, Magnuson, 1977), can be used to produce a feasible placement. Consequently, practically every known heuristic scheme, including cluster development (Areibi and Yang, 2004, Hanan and Kurtzberg, 1972a, Hanan et al., 1976a, Magnuson, 1977, Cox and Carroll, 1980), knowledge based systems (Pannérec, 2003), randomized local search algorithms such as simulated annealing (Sechen and Sangiovanni-Vincentelli, 1986, Sechen, 1988, Wong et al., 1988, Wang et al., 2000, Murata et al., 1998), and genetic algorithms (Cohoon and Paris, 1987, Shahookar and Mazumder, 1990, Valenzuela and Wang, 2002, Sait et al., 2005, Areibi and Yang, 2004), as well as combinations of these approaches (Zhang et al., 2005) have been used to compute placements. Often, computed placements are improved by iterative heuristics based on component interchange (Magnuson, 1977, Coté and Patel, 1980). Placement methods can be classified roughly into three categories: recursive minimum cut placement, analytical placement and local search methods.

3.1 Recursive Minimum Cut Placement

The idea of recursive minimum cut placement is to partition the circuit into subcircuits subject to minimizing the number of wires running between the two partitions. Simultaneously, the board surface is divided vertically or horizontally into subregions and each subcircuit is assigned to one region. This procedure is repeatedly applied to each of the subcircuits and subregions until the remaining circuit can be trivially placed, for instance, if it consists of a single component only.

The problem can be stated by representing the circuit as a hypergraph. The components correspond to the nodes and the nets correspond to the hyperedges connecting these components (nodes). Thus, decomposing the circuit into two subcircuits with an approximately equal share of components subject to minimizing the number of wires between the two subcircuits corresponds to finding a balanced bipartition (or a bisection) of the hypergraph with a minimal cut. This results in the so called *balanced hypergraph bipartition problem*, and *hypergraph bisection problem*, respectively. Both problems are $\mathcal{NP}$-hard (Lengauer, 1990), but many heuristics were proposed in literature. Two very common approaches in practice are the heuristic of Fiduccia and Mattheyses (1982) for solving the balanced hypergraph bipartition problem and the algorithm of Kernighan and Lin (1970) for solving the bisection problem on graphs after transforming the hypergraph to a graph by replacing each hyperedge by a clique. Both approaches are based on the idea of iterative improvement. Starting with an initial balanced bipartition (or bisection) of the hypergraph (graph), nodes are interchanged between the two subsets, to reduce the size of the cut.

The recursive minimum cut placement can be extended to partitions with more than two subsets. The placement problem then becomes a *multiway partition problem*. Efficient implementations of the hypergraph multiway partition problem are presented in Karypis and Kumar (1999). Recently, Zhao et al. (2005)

proposed a unified framework for developing and analyzing approximation algorithms for various multiway partition problems.

A more extensive discussion on partition-based placement methods for VLSI design is given by Lengauer (1990) and a survey is presented in Alpert and Kahng (1995). In Yang and Wong (1996) a partition approach based on the max-flow-min-cut theorem is proposed. The experimental results demonstrate that their approach outperforms the Kernighan-Lin heuristics in terms of the number of crossings. In Mak and Wong (2000) a fast non-flow-based algorithm for computing a minimum cut in a hypergraph is suggested. Finally, in Papa and Markov (2006) a survey of partitioning and clustering methods is presented.

3.2 Analytical Placement

So called analytical placement is an approach widely used in VLSI design (Sigl et al., 1991, Jünger et al., 1994, Kahng and Wang, 2004, Viswanathan and Chu, 2004), which can also be applied to PCB placement. An advantage of analytical placement over partitioning-based methods is the global view of the problem. In recursive minimum cut placement, minimizing the number of wires crossing a cut in the first step may lead to poor results in the succeeding steps. We present two classes of global analytical placement approaches: quadratic or linear assignment and quadratic placement.

3.2.1 Quadratic assignment

The quadratic assignment approach is based on the following idea: given a number of components and a number of positions on the board surface, we want to assign each component to a position on the board, so that certain physical constraints are satisfied and the total wire length is minimized. This can be formulated as a linear integer optimization problem with either a linear (Akers, 1981) or a quadratic objective function (Hanan and Kurtzberg, 1972b, Hanan et al., 1976a, Weismantel, 1992).

The board surface is assumed to be modeled as a grid which decomposes the board into rectangular grid cells with precisely defined dimensions. A component placed on the grid covers a rectangular area consisting of a set of grid cells.

The placement problem is to assign the components to disjoint rectangular areas of grid cells such that the total wire length is minimized. A component i is said to be assigned to a grid cell k, if i is placed on the board in such a way, that its lower left corner coincides with grid cell k, and a grid cell k is called feasible for a component i, if i fits on the board when assigned to the grid cell k. Let n be the number of components and m the number of grid cells, then we define $Z(i) \subseteq \{1,\dots,m\}$ as the set of feasible grid cells for component i.

For each component $i \in \{1,\dots,n\}$ and each feasible grid cell $k \in Z(i)$ a binary variable x_{ik} is introduced, such that x_{ik} is equal to one if and only if component i is assigned to grid cell k. The quadratic assignment model for component placement proposed by Weismantel (1992) is then defined as follows:

$$\min \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k\in Z(i)} \sum_{l\in Z(j)} c_{ij}\, d(i,k,j,l)\, x_{ik}\, x_{jl} \tag{1}$$

$$+\lambda_0 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k\in Z(i)} \sum_{l\in Z(j)} o(i,k,j,l)\, x_{ik}\, x_{jl} \tag{2}$$

$$\text{subject to} \sum_{k\in Z(i)} x_{ik} = 1 \qquad \text{for all } i = 1,\dots n \tag{3}$$

$$x_{ik} \in \{0,1\} \qquad \text{for all } i = 1,\dots n, k \in Z(i) \tag{4}$$

The objective function is composed of two terms: (1) is an approximation of the total wire length computed in the following way: $c_{ij} \geq 0$ denotes the *affinity coefficients* between the components i and j and can be calculated by $c_{ij} = \sum_{t\in T} \frac{1}{\alpha_t - 1}$ if t is a net connecting i and j, and zero otherwise. T is the set of all nets, α_t the cardinality of a net $t \in T$. $d(i,k,j,l)$ defines the shortest Manhattan distance between the components i and j when assigned to the grid cells k and l.

The second term (2) of the objective function evaluates the number of overlaps. The coefficients $o(i,k,j,l)$ defined for $i \in \{1,\ldots,n\}$, $j \in \{1,\ldots,n\}$, $k \in Z(i)$ and $l \in Z(j)$ counts the number of grid cells that are shared by components i and j when assigned to grid cells k and l. Since a feasible placement has to be free of overlaps, the second term of the objective function is scaled with a large penalty factor λ_0. Equations (3) ensure that each component is assigned to exactly one grid cell.

This quadratic optimization formulation allows the following extension: when some components are allowed to be rotated 90°, four different realizations exist to place each of these components. In this case, binary variables x_{ik}^a are defined for a component i in grid cell $k \in Z(i)$ and for realization $a \in A(i)$, such that x_{ik}^a is equal to one if and only if the component i is assigned to the feasible grid cell k in realization a. The extended quadratic placement problem is the following:

$$\min \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k \in Z(i)} \sum_{l \in Z(j)} \sum_{a_1 \in A(i)} \sum_{a_2 \in A(j)} c_{ij} d(i,k,a_1,j,l,a_2) x_{ik}^{a_1} x_{jl}^{a_2}$$
$$+\lambda_0 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k \in Z(i)} \sum_{l \in Z(j)} \sum_{a_1 \in A(i)} \sum_{a_2 \in A(j)} o'(i,k,a_1,j,l,a_2) x_{ik}^{a_1} x_{jl}^{a_2}$$

$$\text{subject to} \sum_{k \in Z(i)} \sum_{a \in A(i)} x_{ik}^a = 1 \qquad \text{for all } i = 1,\ldots n$$
$$x_{ik}^a \in \{0,1\} \qquad \text{for all } i = 1,\ldots n, k \in Z(i), a \in A(i).$$

Where $d(i,k,a_1,j,l,a_2)$ denotes the shortest Manhattan distance between components i and j when placed on grid cells k and l in realization a_1 and a_2, and $o'(i,k,a_1,j,l,a_2)$ counts the number of overlapping components i and j when placed on grid cells k and l in realization a_1 and a_2. The quadratic model for component placement described above belongs to the class of $\mathcal{NP}$-hard problems. In Weismantel (1992), a decomposition approach is applied to the placement problem.

3.2.2 Quadratic Placement

The main idea in the quadratic placement approach is to model the placement in such a way, that it can be solved with methods of convex optimization. Some quadratic placement approaches (Quinn and Breuer, 1979, Quinn, 1975, Johannes and Eisenmann, 1998, Kleinhans et al., 1991, Khan and Sait, 2004) use a physical force scheme to model the placement problem as a convex optimization problem.

Other approaches use an approximation of the squared total wire length as an objective function, that is minimized under certain constraints (Hall, 1970, Blanks, 1985). With this quadratic program, a placement is first computed without worrying about overlaps, then the overlaps are removed by adding certain constraints to the quadratic program. Viswanathan and Chu (2004) give an overview of analytical placement methods and presents an algorithm called "FastPlace" for the quadratic placement approach in standard cell VLSI design. Recently, new theoretical results on quadratic placement were presented by Vygen (2006).

In quadratic placement the clique model and the star model are traditionally used as models for wire length estimation. In the clique model, each multi-terminal net is represented by a clique and the wire length is approximated by the sum of rectilinear distances over all pairs of points. In the star model, a set of terminals in a net is replaced by a star with uniform edge weights through connecting all terminals to a new additional point, the so called the star node.

4 Modelling Routing

Although wire routing is a long studied problem, fully automatic routing of densely packed boards is a goal difficult to achieve. Even showing the existence of a feasible routing is $\mathcal{NP}$-complete. Since all routing programs used in practice employ heuristics, it is never clear if the problem is infeasible per se or if just the algorithm is not able to find a feasible routing. To quote Dion (1988):

It is always easy to specify a routing problem that is too hard for a program to solve. One need only add wiring to the problem, or remove routing layers. In this sense, designing a completely automatic router is an impossible task. A better program will simply encourage engineers to design harder problems. The only realistic goal for a routing program is to solve practical layout problems well enough that manual intervention is unnecessary.

4.1 Steiner trees

Mathematically, routing a single net can be seen as a *Steiner tree* problem: Given an edge-weighted graph $G = (V,E,c)$ and a non-empty subset of nodes $T \subseteq V$ called *terminals*, find a minimal weight tree in G that spans T. The problem is $\mathcal{NP}$-hard (Karp, 1972), but it is easy to find feasible suboptimal solutions. As a result a large variety of heuristics exists, e. g., Takahashi and Matsuyama (1980), Rayward-Smith and Clare (1986), Winter and Smith (1992), Duin and Voß (1994), Duin and Voß (1999), Ribeiro et al. (2002). Solving large-scale Steiner tree problems to optimality is also possible, see, e. g., Wong (1984), Chopra et al. (1992), Koch and Martin (1998), Polzin and Daneshmand (2001), Polzin (2003).

Integrated circuits sometimes have several pins with the same functionality. It suffices to connect a net to any of several possible terminals. This can be modelled as a *Group Steiner tree* problem: Given a weighted graph $G = (V,E,c)$ and $N \in \mathbb{N}$ pairwise disjoint non-empty subsets of nodes $Z_n \subseteq V$, $n \in \{1,\dots N\}$ called *terminal groups*, find an edge set S^* such that $(V(S^*),S^*)$ is a tree with minimal weight containing at least one node from each group, i. e., $V(S^*) \cap Z_n \neq \emptyset$ for all $n \in N$. For a graph $G = (V,E)$ we denote, for a subset of edges $F \subseteq E$, by $V(F)$ the set of all nodes incident to some edge $e \in F$.

It is possible to transform a group Steiner tree problem back into a normal Steiner tree problem by introducing an artificial node for each terminal group and connecting all terminals of the group with this node. The edges need to have weights that are high enough to ensure that only one of these edges per terminal group is part of the solution. In the transformed problem, only the artificial nodes are terminals. The approach usually taken in practice is to choose the terminal to use from each group before the routing. This is called the *pin assignment problem* (Koren, 1972, Mory-Rauch, 1978, Brady, 1984). Usually the objective is similar to the one used in component placing. For large ICs, e. g., microprocessors, even the routing between the chip itself and the pins of the package has to be considered (Yu et al., 1996, Kubo and Takahashi, 2005). Zachariasen and Rohe (2003) present an algorithm to solve group Steiner tree problems to optimality in the context of VLSI design.

4.2 Steiner tree packing

Routing all the nets at once can be modelled as a *Steiner tree packing* problem: Given a weighted graph $G = (V,E,c)$ and a set of $N \in \mathbb{N}$ pairwise disjoint non-empty subsets of nodes $T_n \subseteq V$, $n \in \{1,\dots,N\}$ called *nets*, find for each net an edge set S_n^* such that $(V(S_n^*),S_n^*)$ is a tree that spans T_n, all edge sets are pairwise disjoint, and the total weight $\sum_{e\in\bigcup_{n\in N} S_n^*} c_e$ is minimal. Since the Steiner tree problem is a special case of the Steiner tree packing problem, the packing problem is also $\mathcal{NP}$-hard. But there is an important difference: for the packing problem even finding a feasible solution, without regarding the weights, is $\mathcal{NP}$-complete, see Kramer and van Leeuwen (1984). A survey of different integer programming models for Steiner tree packing can be found in Chopra (1994). We will examine two of the models, which have been subject to mathematical and practical investigation, in more detail.

4.2.1 Multicommodity flow formulation

The multicommodity flow formulation as proposed by Wong (1984) has the advantage that it has only a polynomial number of variables and constraints: Given a weighted bidirectional grid digraph $G = (V,A,c)$, and sets $T_1,\dots,T_N$, $N > 0$, $\mathcal{N} = \{1,\dots,N\}$ of terminals, we arbitrarily choose a root $r_n \in T_n$ for each $n \in \mathcal{N}$. Let $R = \{r_n | n \in \mathcal{N}\}$ be the set of all roots and $T = \bigcup_{n\in\mathcal{N}} T_n$ be the union of all terminals. We introduce binary variables $\bar{x}_{ij}^n$ for all $n \in \mathcal{N}$ and $(i,j) \in A$, where $\bar{x}_{ij}^n = 1$ if and only if arc $(i,j) \in S_n$. Additionally we introduce non-negative variables y_{ij}^t, for all $t \in T \setminus R$. For all $i \in V$, we define $\delta_i^+ := \{(i,j) \in A | j \in V\}$ and $\delta_i^- := \{(j,i) \in A | j \in V\}$. For all $t \in T_n$, $n \in \mathcal{N}$, we define $\sigma(t) := n$. The following formulation models the Steiner tree packing problem for any graph requiring edge disjoint but not node disjoint routings of all nets.

$$\min \sum_{n\in\mathcal{N}} \sum_{(i,j)\in A} c^n_{ij}\vec{x}^n_{ij}$$

$$\sum_{(i,j)\in\delta^-_j} y^t_{ij} - \sum_{(j,k)\in\delta^+_j} y^t_{jk} = \left\{\begin{array}{rl} 1 & \text{if } j=t \\ -1 & \text{if } j=r_{\sigma(t)} \\ 0 & \text{otherwise} \end{array}\right\} \qquad \text{for all } j\in V, t\in T\setminus R \tag{5}$$

$$0 \le y^t_{ij} \le \vec{x}^{\sigma(t)}_{ij} \qquad \text{for all } (i,j)\in A, t\in T\setminus R \tag{6}$$

$$\sum_{n\in\mathcal{N}} (\vec{x}^n_{ij} + \vec{x}^n_{ji}) \le 1 \qquad \text{for all } (i,j)\in A \tag{7}$$

$$\vec{x}^n_{ij} \in \{0,1\} \qquad \text{for all } n\in\mathcal{N}, (i,j)\in A \tag{8}$$

In the model above two different nets may share a common node. To obtain a node disjoint solution we have to add:

$$\sum_{n\in\mathcal{N}} \sum_{(i,j)\in\delta^-_j} \vec{x}^n_{ij} \le \left\{\begin{array}{ll} 0 & \text{if } j\in R \\ 1 & \text{otherwise} \end{array}\right. \qquad \text{for all } j\in V \tag{9}$$

Results using this model to compute optimal routings can be found in Koch (2004). The advantage of the formulation is that it modells all the layers simultaneously. On the other hand the size of the graph grows rapidly with the number of terminals. To circumvent this problem another formulation can be used.

4.2.2 Undirected partitioning formulation

This formulation is used by Grötschel et al. (1996a) and by Jorgensen and Meyling (2002). Given a weighted grid graph $G=(V,E,c)$, and terminal sets $T_1,\dots,T_N$, $N>0$, $\mathcal{N}=\{1,\dots,N\}$, we introduce binary variables x^n_{ij} for all $n\in\mathcal{N}$ and $ij\in E$, where $x^n_{ij}=1$ if and only if edge $ij\in S_n$. We define $\delta(W)=\{ij\in E | i\in W, j\notin W\}$ for $W\subseteq V$. The following formulation models edge disjoint one-layer routing:

$$\min \qquad \sum_{n\in\mathcal{N}} \sum_{ij\in E} c_{ij}x^n_{ij}$$

$$\sum_{ij\in\delta(W)} x^n_{ij} \ge 1 \qquad \text{for all } W\subset V, W\cap T_n \neq \emptyset, (V\setminus W)\cap T_n \neq \emptyset, n\in\mathcal{N} \tag{10}$$

$$\sum_{n\in\mathcal{N}} x^n_{ij} \le 1 \qquad \text{for all } ij\in E \tag{11}$$

$$x^n_{ij}\in\{0,1\} \qquad \text{for all } n\in\mathcal{N}, ij\in E \tag{12}$$

The model can be further strengthened with several valid inequalities as described in Grötschel et al. (1996a,b), Grötschel et al. (1997). By using capacities on the edges the formulation can be extended to model an arbitrary number of layers.

Since there is only one layer explicitly in the model, the assignment of the wires to the layers has to be done in a subsequent step. This is called the *layer assignment* problem. Assigning layers means deciding the number and location of the vias on the PCB. As every via increases the poduction costs and decreases the production yield, *via minimization* is the goal of this step. Brady and Brown (1984) have designed an algorithm that guarantees that any solution in the above model can be routed in four layers, but deciding whether three layers are enough is shown to be $\mathcal{NP}$-complete by Lipski (1984).

In practice, since the components are already placed (and can only be placed on the outer layers of the PCB) the layers for the terminals are already fixed. Grötschel et al. (1989) show how to transform this problem into a max-cut problem and describe exact and heuristic solution approaches. Further graph based approaches can be found in Chen et al. (1983), Naclerie et al. (1987), Xiong and Kuh (1988), Fang et al. (1991), Chang and Cong (1997), Chou and Lin (1998).

4.3 Heuristic routing

In practice heuristics are used to route the nets (Dysart and Koifman, 1979, Aranoff and Abulaffio, 1981, Naveda et al., 1986, Dion, 1988). There are two basic techniques for finding paths between terminals (Pecht, 1993): grid based maze routers as introduced by Lee (1961), Akers (1972) and gridless routers, as described by Lauther (1980), Finch et al. (1985), Schiele et al. (1990). Since just trying to route each connection one after the other is not likely to result in a feasible routing usually one of the following two techniques is employed: In so-called *dynamic* or *rip-up and retry* routing whenever a wire cannot be routed, the blocking wires are removed and rerouted later after the new wire has been successfully routed (Dees and Smith, 1981, Rosenberg, 1987, Raith and Bartholomeus, 1991). In *iterative* routing several passes are made. In each pass all the connections are routed regardless whether illegal crossings occur. In the next iteration some penalty on those areas of the board is increased that experience congestion. The hope is that this will lead to different routes of some wires in the next pass (Fisher, 1978, Moosa and Edwards, 1995). Even hybrid approaches of these techniques are possible (Cong and Madden, 1998, Hadsell and Madden, 2003). A further technique that can be used on top of the ones described is to build routing hierarchies by partitioning the board and first routing between important (congested) areas only and then going into smaller detailed areas (Ozdal and Wong, 2004b). For example, Kawamura et al. (1986) describe a hierarchical dynamic router that employs three levels of hierarchy. The hierarchy can also be used to include further concepts like electromagnetic compatibility as reported by Schmidt et al. (1995). Ozdal and Wong (2004a) describe an algorithm for high performance single-layer bus routing, where the objective is to match the lengths of all nets belonging to each bus.

5 Conclusion

Continuing advances in production techniques increase the requirements on physical layout algorithms. From a theoretical point of view it would be obviously best to use a holistic approach to find a physical PCB design. It is clear that this is far from reality in practice. While some subproblems such as via minimization and layer assignment tend to be reintegrated with routing, new subproblems like pin assignment for the IC packages emerge.

For research, one of the biggest shortcomings is the absence of publicly available benchmark cases. Readers with more than superficial interest in the subject presented here may have wished to obtain a sound and fact based judgment of the relative merits and disadvantages of the various approaches to placement and routing surveyed in this paper. We would love to present such findings, but for many reasons it is impossible to make fair test runs necessary for such comparisons. Clearly, every group of authors shows in their papers that their approach has some advantages in comparison to other methods on the examples considered. But it is not possible to obtain the codes and the test instances to make "neutral" runs. Moreover, production codes are usually fine tuned to specific customer demands, particular layout properties, and design rules to which competititors (usually) do not get access. In this sense no two routing or no two placement programs address exactly the same problem. This deficiency could be remediated if the PCB/VLSI layout community would have access to a large collection of real instances, including all specific layout requirements, allowing to test new codes and ideas in an open competitive environment. Unfortunately, the electronic industry does not seem to be ready to make up-to-date realistic test instances publicly available and thus, "benchmarking suites" consist only of small academic (made-up) examples with questionable bearing on real layout problems.

This situation not only makes it difficult to test new ideas, especially for people without access to tool suites, it also makes it difficult to measure advancement in the field. While finding an optimal solution for a contemporary PCB layout problem might be completely out of reach, it could be possible to solve the problems from 20 years ago to optimality. This might give an interesting insight in how much is lost by the standard approach of partitioning the problem into several hierarchically executed steps.

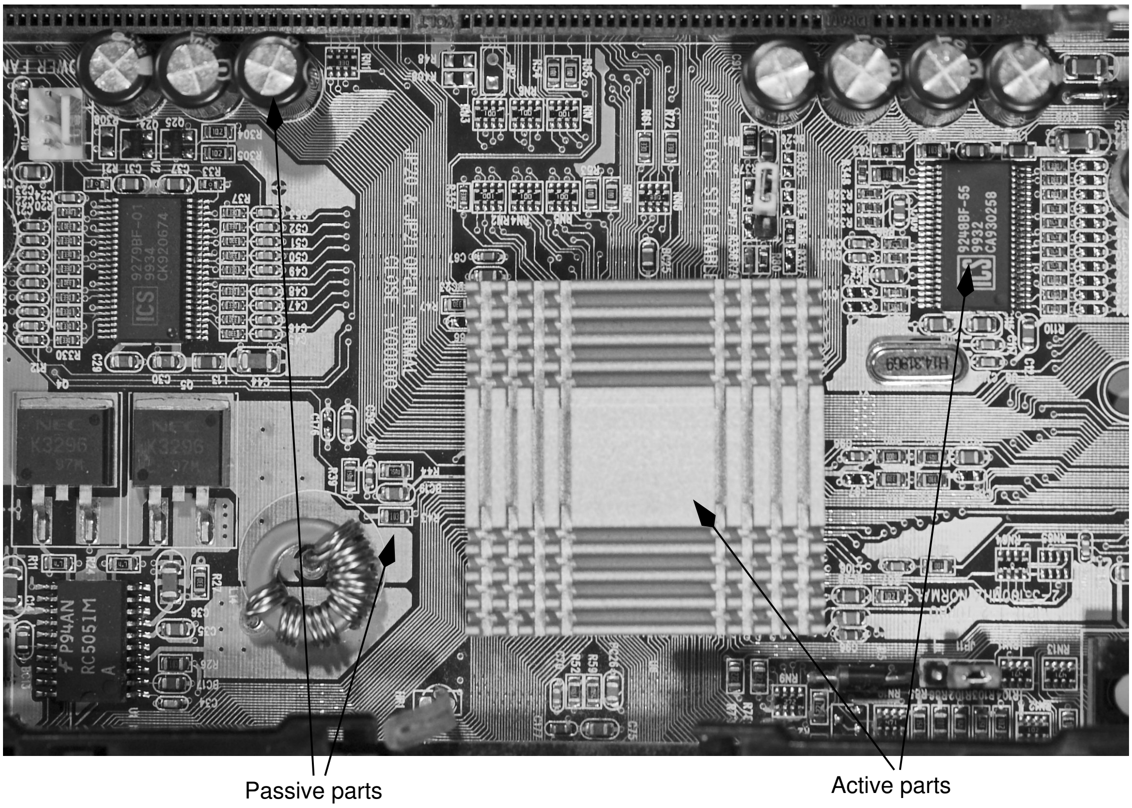

Figure 2: Component side of a printed circuit board

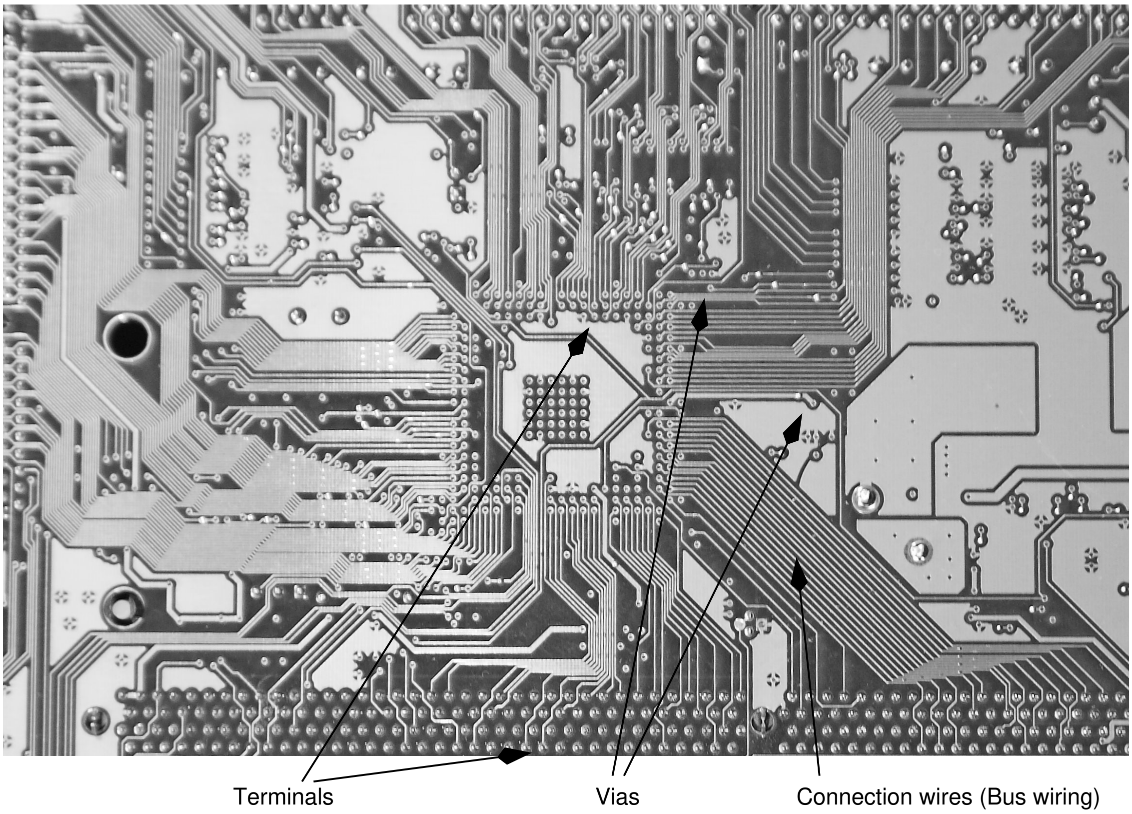

Figure 3: Bottom side of a printed circuit board

References

S. B. Akers. Routing. In M. A. Breuer, editor, *Design Automation of Digital Systems, Volume 1: Theory and Techniques*, pages 283–333. Prentice Hall, 1972.

S. B. Akers. On the use of the linear assignment algorithm in module placement. In *Proceedings of the 18th conference on Design automation*, pages 137–144. IEEE Press, 1981.

C. J. Alpert and A. B. Kahng. Recent directions in netlist partitioning: a survey. *Integration, the VLSI Journal*, 19:1–81, 1995.

S. Aranoff and Y. Abulaffio. Routing of printed circuit boards. In *Proceedings of the 18th conference on Design automation*, pages 130–136. IEEE Press, 1981.

S. Areibi and Z. Yang. Effective memetic algorithms for VLSI design = genetic algorithms + local search + multi-level clustering. *Evolutionary Computation*, 12(3):327–353, 2004.

J. P. Blanks. Near-optimal placement using a quadratic objective function. In *Proceedings of the 22nd conference on Design automation*, pages 609–615. IEEE Press, 1985.

H. N. Brady. An approach to topological pin assignment. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 3:250–255, 1984.

M. L. Brady and D. J. Brown. VLSI routing: Four layers suffice. In F. P. Preparata, editor, *Advances in Computing Research: VLSI theory*, volume 2, pages 245–258. Jai Press, London, 1984.

C.-C. Chang and J. Cong. An efficient approach to multi-layer layer assignment with application to via minimization. In *Proceedings of the 34th conference on Design automation*, pages 600–603. ACM Press, 1997.

R. W. Chen, Y. Katjitani, and A. P. Chan. A graph-theoretic via minimization algorithm for two-layer printed circuit boards. *IEEE Transactions on Circuits and Systems*, 30:284–299, 1983.

S. Chopra. Comparison of formulations and a heuristic for packing steiner trees in a graph. *Annals of Operations Research*, 50:143–171, 1994.

S. Chopra, E. R. Gorres, and M. R. Rao. Solving the steiner tree problem on a graph using branch and cut. *ORSA Journal on Computing*, 4(3):320–335, 1992.

Y.-C. Chou and Y.-L. Lin. A graph-partitioning-based approach for multi-layer constrained via minimization. In *ICCAD '98: Proceedings of the 1998 international conference on Computer-aided Design*, pages 426–429. ACM Press, 1998.

J. P. Cohoon and W. D. Paris. Genetic placement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 6(6):956–964, 1987.

J. Cong and P. H. Madden. Performance driven multi-layer general area routing for PCB/MCM designs. In *Proceedings of the 35th conference on Design automation*, pages 356–361. ACM Press, 1998.

L. C. Coté and A. M. Patel. The interchange algorithms for circuit placement problems. In *Proceedings of the 17th conference on Design automation*, pages 528–534. IEEE Press, 1980.

G. W. Cox and B. D. Carroll. The standard transistor array (star), part II: automatic cell placement techniques. In *Proceedings of the 17th conference on Design automation*, pages 451–457. IEEE Press, 1980.

J. W. A. Dees and I. R. J. Smith. Performance of interconnection rip-up and reroute strategies. In *Proceedings of the 18th conference on Design automation*, pages 382–390. IEEE Press, 1981.

J. Dion. Fast printed circuit board routing. Technical report, Digital Equipment Corporation, Western Research Laboratory, 1988.

C. Duin and S. Voß. Steiner tree heuristics — A survey. In H. Dyckhoff, U. Derigs, M. Salomon, and H. C. Tijms, editors, *Operations Research Proceedings*, pages 485–496. Springer, 1994.

C. Duin and S. Voß. The Pilot method: A strategy for heuristic repetition with application to the Steiner problem in graphs. *Networks*, 34:181–191, 1999.

L. Dysart and M. Koifman. An application of branch and bound method to automatic printed circuit board routing. In *Proceedings of the 16th Conference on Design automation*, pages 494–499. IEEE Press, 1979.

S.-C. Fang, K.-E. Chang, W.-S. Feng, and S.-J. Chen. Constrained via minimization with practical considerations for multi-layer VLSI/PCB routing problems. In *Proceedings of the 28th conference on Design automation*, pages 60–65. ACM Press, 1991.

C. M. Fiduccia and R. M. Mattheyses. A linear time heuristic for improving network partitions. In *Proceedings of the 19th conference on Design automation*, pages 175–181. IEEE Press, 1982.

A. C. Finch, K. J. Mackenzie, G. J. Balsdon, and G. Symonds. A method for gridless routing of printed circuit boards. In *Proceedings of the 22nd conference on Design automation*, pages 509–515. ACM Press, 1985.

R. S. Fisher. A multi-pass, multi-algorithm approach to PCB routing. In *Proceedings of the 15th conference on Design automation*, pages 82–91. IEEE Press, 1978.

M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the theory of NP-Completeness*. Freeman, 1979.

S. H. Gerez. *Algorithms for VLSI Design Automation*. Wiley, 1998. ISBN 0471984892.

M. Grötschel, M. Jünger, and G. Reinelt. Via minimization with pin preassignments and layer preference. *ZAMM – Zeitschrift für Angewandte Mathematik und Mechanik*, 69(11):393–399, 1989.

M. Grötschel, A. Martin, and R. Weismantel. Packing Steiner trees: A cutting plane algorithm and computational results. *Mathematical Programming*, 72:125–145, 1996a.

M. Grötschel, A. Martin, and R. Weismantel. Packing Steiner trees: Further facets. *European Journal of Combinatorics*, 17:39–52, 1996b.

M. Grötschel, A. Martin, and R. Weismantel. The Steiner tree packing problem in VLSI design. *Mathematical Programming*, 78(2):265–281, 1997.

R. T. Hadsell and P. H. Madden. Improved global routing through congestion estimation. In *Proceedings of the 40th conference on Design automation*, pages 28–31. ACM Press, 2003.

K. M. Hall. An r-dimensional quadratic placement algorithm. *Management Science*, 17(3):219–229, 1970.

M. Hanan and J. M. Kurtzberg. Placement techniques. In M. A. Breuer, editor, *Design Automation of Digital Systems. Volume One: Theory and Techniques*, pages 213–282. Prentice-Hall, 1972a.

M. Hanan and J. M. Kurtzberg. A review of the placement and quadratic assignment problem. *SIAM Review*, 14:324–342, 1972b.

M. Hanan, P. K. Wolff Sr., and B. J. Agule. Some experimental results on placement techniques. In *Proceedings of the 13th conference on Design automation*, pages 214–224. IEEE Press, 1976a.

M. Hanan, P. K. Wolff Sr., and B. J. Agule. A study of placement techniques. *Journal of Design Automation and Fault-Tolerant Computing*, 1(1):28–61, 1976b.

J. E. Harlow. Overview of popular benchmark sets. *IEEE Design and Test of Computers*, 17(3):15–17, 2000.

I. Hom and J. Granacki. Estimation of the number of routing layers and total wirelength in a PCB through wiring distribution analysis. In *Proceedings of the European Design automation conference*, pages 310–315. 1996.

F. M. Johannes and H. Eisenmann. Generic global placement and floorplanning. In *Proceedings of the 35th conference on Design automation*, pages 269–274. IEEE Press, 1998.

D. G. Jorgensen and M. Meyling. A branch-and-price algorithm for switch-box routing. *Networks*, 40(1):13–26, 2002.

M. Jünger, A. Martin, G. Reinelt, and R. Weismantel. Quadratic 0/1 optimization and a decomposition approach for the placement of electronic circuits. *Mathematical Programming*, 63(3):257–279, 1994.

A. B. Kahng and Q. Wang. An analytic placer for mixed-size placement and timing-driven placement. In *International Conference on Computer Aided Design*, pages 565–572. IEEE Press, 2004.

R. M. Karp. Reducibility among combinatorial problems. In J. W. Thatcher, editor, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.

G. Karypis and V. Kumar. Multilevel *k*-way hypergraph partitioning. In *Proceedings of the 36th conference on Design automation*, pages 343–348. 1999.

K. Kawamura, M. Umeda, and H. Shiraishi. Hierarchical dynamic router. In *Proceedings of the 23nd conference on Design automation*, pages 803–809. ACM Press, 1986.

B. W. Kernighan and S. Lin. An Efficient Heuristic Procedure for Partitioning Graphs. *Bell Systems Technical Journal*, 49(2):291–307, 1970.

J. A. Khan and S. M. Sait. Fast force-directed/simulated evolution hybrid for multiobjective VLSI cell placement. In *International Symposium on Circuits and Systems*, pages 53–56. 2004.

J. Kleinhans, G. Sigl, F. Johannes, and K. Anstreich. GORDIAN: VLSI Placement by Quadratic Programming and Slicing Optimization. *IEEE Transactions on Computer-Aided Design*, 10(3):356–365, 1991.

T. Koch. *Rapid Mathematical Programming*. Ph.D. thesis, Technische Universität Berlin, 2004.

T. Koch and A. Martin. Solving Steiner tree problems in graphs to optimality. *Networks*, 32:207–232, 1998.

N. L. Koren. Pin assignment in automated printed circuit board design. In *Proceedings of the 9th conference on Design automation*, pages 72–79. ACM Press, 1972.

M. R. Kramer and J. van Leeuwen. The complexity of wire-routing and finding minimum area layouts for arbitrary VLSI circuits. In F. P. Preparata, editor, *Advances in Computing Research*, volume 2, pages 129–146. Jay Press, 1984.

Y. Kubo and A. Takahashi. A global routing method for 2-layer ball grid array packages. In *ISPD '05: Proceedings of the 2005 international symposium on Physical design*, pages 36–43. ACM Press, 2005.

R. Kumar, W. kit Leong, and R. J. Heath. An integer programming approach to placement and routing in circuit layout, 2005. `http://clue.eng.iastate.edu/~rkumar/PUBS/layout.ps`.

U. Lauther. A data structure for gridless routing. In *Proceedings of the 17th conference on Design automation*, pages 603–609. IEEE Press, 1980.

C. Y. Lee. An algorithm for path connection and its applications. *IRE Transactions on Electronic Computers*, EC-10(3):346–365, 1961.

T. Lengauer. *Combinatorial Algorithms for Integrated Circuit Layout*. B. G. Teubner, Stuttgart, 1990.

W. Lipski. On the structure of three-layer wireable layouts. In F. P. Preparata, editor, *Advances in Computing Research: VLSI theory*, volume 2, pages 231–244. Jai Press, London, 1984.

W. G. Magnuson. A comparison of constructive placement algorithms. IEEE Region 6 Conf. Rec., 1977.

W.-K. Mak and D. F. Wong. A fast hypergraph min-cut algorithm for circuit partitioning. *Integration, the VLSI Journal*, 30(1):1–11, 2000.

F. Mo, A. Tabbara, and R. K. Brayton. A force-directed maze router. In *International Conference on Computer Aided Design*, pages 404–407. IEEE Press, 2001.

Z. Moosa and D. Edwards. An investigation of iterative routing algorithms. In *Proceedings of the conference on European Design automation*, pages 91–96. IEEE Computer Society Press, 1995.

L. Mory-Rauch. Pin assignment on a printed circuit board. In *Proceedings of the 15th conference on Design automation*, pages 70–73. IEEE Press, 1978.

H. Murata, K. Fujiyoshi, and M. Kaneko. VLSI/PCB Placement with Obstacles Based on Sequence Pair . *IEEE Transactions on computer-aided design of integrated circuits and systems*, 17(1):60–68, 1998.

N. J. Naclerie, S. Masuda, and K. Nakajima. Via minimization for gridless layouts. In *Proceedings of the 24th conference on Design automation*, pages 159–165. ACM Press, 1987.

J. F. Naveda, K. C. Chang, and H. C. Du. A new approach to multi-layer PCB routing with short vias. In *Proceedings of the 23rd conference on Design automation*, pages 696–701. IEEE Press, 1986.

M. M. Ozdal and M. D. F. Wong. A provably good algorithm for high performance bus routing. In *International Conference on Computer Aided Design*, pages 830–837. IEEE Press, 2004a.

M. M. Ozdal and M. D. F. Wong. Simultaneous escape routing and layer assignment for dense PCBs. In *International Conference on Computer Aided Design*, pages 822–829. IEEE Press, 2004b.

T. Pannérec. Knowledge-based automatic components placement for single-layer PCB layout. In *Proceedings of the 7th international conference on Knowledge-Based Intelligent Information and Engineering Systems: Part I*, volume 2773 of *Lecture Notes in Computer Science*, pages 669–675. Springer, 2003.

D. A. Papa and I. L. Markov. Hypergraph Partitioning and Clustering. In T. Gonzalez, editor, *Approximation Algorithms and Metaheuristics*. CRC Press, 2006.

M. Pecht, editor. *Placement and Routing of Electronic Modules*. CRC, 1993.

T. Polzin. *Algorithms for the Steiner Problem in Networks*. Ph.D. thesis, Universität des Saarlandes, 2003.

T. Polzin and S. V. Daneshmand. Improved algorithms for the steiner problem in networks. *Descrete Applied Mathematics*, 112:263–300, 2001.

B. T. Preas and P. J. Karger. Automatic placement: A review of current techniques. In *Proceedings of the 23th conference on Design automation*, pages 622–629. IEEE Press, 1986.

N. R. Quinn. The placement problem as viewed from the physics of classical mechanics. In *Proceedings of the 12th conference on Design automation*, pages 173–178. 1975.

N. R. Quinn and M. A. Breuer. A Forced Directed Component Placement Procedure for Printed Circuit Boards. *IEEE Transactions on Circuits and Systems*, 26(6):377–388, 1979.

M. Raith and M. Bartholomeus. A new hypergraph based rip-up and reroute strategy. In *Proceedings of the 28th conference on Design automation*, pages 54–59. ACM Press, 1991.

V. J. Rayward-Smith and A. Clare. On finding steiner vertices. *Networks*, 16:283–294, 1986.

C. C. Ribeiro, E. Uchoa, and R. F. Werneck. A hybrid GRASP with perturbations for the steiner problem in graphs. *INFORMS Journal on Computing*, 14:228–246, 2002.

E. Rosenberg. A new interactive supply/demand router with rip-up capability for printed circuit boards. In *Proceedings of the 24th conference on Design automation*, pages 721–726. IEEE Press, 1987.

S. M. Sait, M. Faheemuddin, M. R. Minhas, and S. Sanaullah. Multiobjective VLSI cell placement using distributed genetic algorithm. In H.-G. Beyer and U.-M. O'Reilly, editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1585–1586. ACM Press, 2005.

S. M. Sait and H. Youssef. *VLSI Physical Design Automation: Theory and Practice*. World Scientific, 1999. ISBN 9810238835.

W. L. Schiele, T. Krüger, K. M. Just, and F. H. Kirsch. A gridless router for industrial design rules. In *Proceedings of the 27th conference on Design automation*, pages 626–631. ACM Press, 1990.

H. Schmidt, D. Theune, R. Thiele, and T. Lengauer. EMC-driven midway routing on PCBs. In *Proceedings of the European conference on Design and Test*, page 486. IEEE Press, 1995.

C. Sechen. *VLSI placement and global routing using simulated annealing*. Kluwer, 1988.

C. Sechen and A. Sangiovanni-Vincentelli. Timberwolf 3.2 : A new standard cell placement and global routing package. In *Proceedings of the 23rd conference on Design automation*, pages 432–439. IEEE Press, 1986.

K. Shahookar and P. Mazumder. A genetic approach to standard cell placement using meta-genetic parameter optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 9(5):500–511, 1990.

G. Sigl, K. Doll, and F. M. Johannes. Analytical placement: A linear or a quadratic objective function? In *Proceedings of the 36th conference on Design automation*, pages 427–432. 1991.

M. F. M. Stallmann, F. Brglez, and D. Ghosh. Heuristics, experimental subjects, and treatment evaluation in bigraph crossing minimization. *ACM Journal of Experimental Algorithms*, 6:8, 2001.

H. Takahashi and A. Matsuyama. An approximate solution for the steiner problem in graphs. *Mathematica Japonica*, 24:573–577, 1980.

C. L. Valenzuela and P. Y. Wang. VLSI placement and area optimization using a genetic algorithm to breed normalized postfix expressions. *IEEE Transactions on Evolutionary Computation*, 6(4):390–401, 2002.

N. Viswanathan and C. C.-N. Chu. Fastplace: efficient analytical placement using cell shifting, iterative local refinement and a hybrid net model. In *Proceedings of the 2004 international symposium on Physical design*, pages 26–33. ACM Press, 2004.

J. Vygen. *Theory of VLSI Layout*. Ph.D. thesis, Universität Bonn, 2001. Habilitationsschrift.

J. Vygen. New theoretical results on quadratic placement. *Integration, the VLSI Journal*, 2006.

M. Wang, X. Yang, and M. Sarrafzadeh. Dragon2000: Standard-cell placement tool for large industry circuits. *International Conference on Computer Aided Design*, page 260, 2000.

R. Weismantel. Plazieren von Zellen: Theorie und Lösung eines quadratischen 0/1-Optimierungsproblems. Technical Report TR 92-03, Zuse Institute Berlin, 1992.

N. Whitaker. Status report on EDA benchmarks. Technical Report STEED/T1/01/4, MINT Group, University of Manchester, 1996. `http://mint.cs.man.ac.uk/projects/steed/benchmarks.html`.

P. Winter and J. M. Smith. Path-distance heuristics for the steiner problem in undirected networks. *Algorithmica*, pages 309–327, 1992.

D. F. Wong, H. W. Leong, and C. L. Liu. *Simulated annealing for VLSI design*. Kluwer, 1988.

R. T. Wong. A dual ascent approach for the steiner tree problems on a directed graph. *Mathematical Programming*, 28:271–287, 1984.

X.-M. Xiong and E. S. Kuh. The constrained via minimization problem for PCB and VLSI design. In *Proceedings of the 25th conference on Design automation*, pages 573–578. IEEE Computer Society Press, 1988.

H. Yang and D. F. Wong. Efficient Network Flow Based Min-Cut Balanced Partitioning. *IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems*, 15(12):1533–1540, 1996.

M.-F. Yu, J. Darnauer, and W. W.-M. Dai. Interchangeable pin routing with application to package layout. In *International Conference on Computer Aided Design*, pages 668–673. IEEE Press, 1996.

M. Zachariasen and A. Rohe. Rectilinear group steiner trees and applications in VLSI design. *Mathematical Programming*, 94:407–433, 2003.

L. Zhang, R. Raut, Y. Jiang, U. Kleine, and Y. Kim. A hybrid evolutionary analogue module placement algorithm for integrated circuit layout designs . *International Journal of Circuit Theory and Applications*, 33(6):487–501, 2005.

L. Zhao, H. Nagamochi, and T. Ibaraki. Greedy splitting algorithms for approximating multiway partition problems. *Mathematical Programming*, 102(1):167–183, 2005.