Recent Developments in Wiring and Via Minimization

Paul Molitor

Institut für Informatik, Martin-Luther Universität Halle,
D-06099 Halle (Saale), Germany

A main feature of VLSI design systems is the placement and routing aspect. A routing problem is given by a routing region, a set of multiterminal nets (the demands) and the number of available layers. In this chapter the routing region will always be a planar graph, called grid. Cross-overs, junctions, knock-knees, bends, and vias may only be placed on vertices of the routing region. Routing itself typically consists of two steps. The first step determines the placement of the routing segments, which is the wire layout. In the second step which is called wiring or layer assignment, each wire segment of the wire layout has to be assigned to one of the $k$ available layers so that the segments are electrically connected in the right way.

In order to state the problem more precisely, we need some definitions (cf. [9]). The routing region $R$ is given by a connected planar graph (the grid) which is embedded in the euclidian plane. The set of multiterminal nets is given by a set $N = \{N_1, \ldots, N_p\}$, where each $N_i$ is a subset of the grid points, such that $N_i \cap N_j = \emptyset$ for all $i \neq j$. Each $N_i$ is called a net. The elements of $N_i$ are called its terminals. We assume that there are $k$ conducting layers $L_1, \ldots, L_k$, each is a copy of the grid. $L_{i+1}$ is considered to be laid upon $L_i$, $1 \leq i \leq k - 1$. A wire layout of a routing problem is a mapping that associates each net $N_i$ to a subgraph $W_i$.
of the grid $R$ connecting all terminals of $N_i$. Such a connected subgraph (or a part of it) is called wire or wire segment. If $W_i$ does not share an edge with $W_j$ for all $i \neq j$, the wire layout is a generalized knock-knee mode wire layout. In literature, knock-knee mode wire layouts are only defined for square grids, where a square grid is a subgraph of the integer grid which has vertices $x \in \mathbb{N}_0 \times \mathbb{N}_0$ and edges $\{x, y\}$, where $x = (x_1, x_2), y = (y_1, y_2)$ and $|x_1 - y_1| + |x_2 - y_2| = 1$, i.e., which models the popular constraint that wires can only run horizontally and vertically in a natural way. If the wire layout is not in generalized knock-knee mode, it is said to be an overlap mode wire layout. A $k$-layer wiring of a wire layout $W = \{W_1, \ldots, W_p\}$ is a mapping that, for each $W_i \in W$, associates each edge in $W_i$ to a layer in $\{L_1, \ldots, L_k\}$. This is done in such a way that for any $i \neq j$, if there are edges $(v_1, v_2)$ and $(v_2, v_3)$ in $W_i$ which are assigned to $L_s$ and $L_t$, respectively, and an edge $(v_2, v_4)$ in $W_j$ which is assigned to $L_u$, then $u > \max \{s, t\}$ or $u < \min \{s, t\}$. It follows that in a wiring all terminals from the same net are made electrically common – if edges $(v_1, v_2)$ and $(v_2, v_3)$ in $W_i$ are assigned to $L_s$ and $L_t$, the segments can be connected through a via – and no two distinct nets are electrically connected. A wire layout is called $k$-layer wirable if there is a $k$-layer wiring of this wire layout.

In this chapter we first give a historical review on $k$-layer wiring and its co-herent problems (section 1). The remainder sections focus on 2-layer wiring which is discussed in detail. Two-layer wiring is important because the current technologies do wire the signal nets with 2 layers, using either two metal layers or one polysilicium and one metal layer. Section 2 discusses the decision problem as to whether a generalized knock-knee mode or overlap mode wire layout is 2-layer wirable, the problem of minimally stretching a wire layout to ensure 2-layer wirability, and the constrained via minimization problem which is the problem of computing a 2-layer wiring with a minimal number of vias. Section 3 and 4 discuss new formulations of the wiring problem for VLSI circuits in the case of two layers available. The first one has recently been given by Ciesielski [5] and Kaufmann, Molitor, and Vogelgesang [15]. The objective of this problem is to minimize the interconnect delay by taking into account the resistance and capacity of interconnection wires and vias. The second one arises during hierarchical physical synthesis where the following problem has to be addressed. Let $A$ be a circuit composed of macro cells whose input and output pins lie in certain but
fixed layers. Assume that the placement and routing phase is already completed. Find a 2-layer wiring of the wire segments of $A$ such that the pins of the macro cells lie in the preassigned layers and the number of vias is minimal on this condition. A solution of this problem induces a hierarchical bottom-up 2-layer wiring algorithm which preserves the original layout hierarchy of the circuit.

1 Review on wiring

The results on the decision problem as to whether a given wire layout is $k$-layer wirable were achieved by Brady and Brown [2] ($k \geq 4$), by Lipski [18] ($k = 3$), and Pinter [27] and Molitor [22] ($k = 2$). Brady and Brown showed that any knock-knee mode wire layout in the square grid can be wired using four layers – the problem is still open for the case of overlap mode and generalized knock-knee mode wire layouts. Lipski proved that it is NP-complete to decide whether a (square) grid based wire layout can be wired using three layers. In [22, 27] the decision problem for 2 layers is shown to be solvable in runtime $O(w)$ for the case of generalized knock-knee mode wire layouts, where $w$ is the sum of the wire lengths of the wire layout. This result can be extended to the case of overlap mode wire layouts. An efficient uniform algorithm for $k$-layer wiring has been presented by Tollis [30]. It wires each wire layout in two layers if it is 2-layer wirable. Wire layouts which are not 2-layer wirable are wired in three layers in most cases. Of course, some wire layouts that actually need only three layers are wired in four layers because of the NP-completeness of the decision problem for $k = 3$. Nevertheless, to the best of our knowledge, Tollis’ algorithm is the best wiring algorithm with respect to the number of layers used for the case that there are no further optimization criterion as, e.g., finding the best $k$-layer wiring with respect to area, interconnect delay, number of vias and so on.

In the context of wirability, the problem arises of how to minimally stretch a wire layout in one dimension in the square grid to get a 2- or 3-layer wirable wire layout. Brady and Sarrafzadeh [3] proved that the problem in three layers is NP-complete for the case of knock-knee mode wire layouts. It is a straightforward conclusion from Lipski’s result on the NP-completeness of the decision problem as to whether a wire layout is 3-layer wirable. They also presented an algorithm for the problem restricted to 2 layers available and knock-knee mode wire layouts.
It runs in time $O(A)$, where $A$ is the area of the routing region. This result has been improved by Kaufmann and Molitor [14]. They have presented an algorithm with running time $O(w)$ and showed that the problem is NP-hard for overlap mode wire layouts. When stretching in two dimensions is allowed the problem is already NP-hard for knock-knee mode wire layouts.

If a wire layout is $k$-layer wirable, one looks for a best $k$-layer wiring. The goal is to optimize the performance of the circuit and possibly minimize its manufacturing costs. In this context, it is important to minimize the number of vias between conductors on different layers since excess vias lead to decreased performance of the electrical circuit, decreased yield of the manufacturing process and increased amount of area required for interconnections. This problem is called CVM (Constrained Via Minimization problem) $(CVM_k)$ and was first formulated by Hashimoto and Stevens [12]. In [22] it has been shown that $CVM_k$ is NP-complete for any $k \geq 3$ for the case of knock-knee mode wire layouts even when the maximum junction degree $\Delta(W)$ is limited to four, where maximum junction degree is defined to be the maximum number $q$ of wire segments which meet at a grid point and which are to be electrically connected. This result has been extended by Choi, Nakajima and Rim [4]. They proved that $CVM_2$ is NP-complete when $\Delta(W) = 4$. In 1982, Pinter [27] presented an algorithm for $CVM_2$ for the case that $\Delta(W)$ is limited to three and the wire layout is in knock-knee mode. This algorithm was based on Hadlock’s maximum cut algorithm for planar graphs [11]. Unfortunately, in Pinter’s approach for wiring, the planar graph on which maximum cut has to be performed has negative and positive weights associated with its edges. However, Hadlock’s algorithm only runs for the case of positive weights (cf. [17]). In 1987, Naclerio, Masuda, and Nakajima [26] and Molitor [22] have independently presented a polynomial time algorithm for the above case which runs in time $O(w^3)$. This algorithm also works for overlap mode wire layouts. In 1988, Kuo, Chern and Shih [17] presented a new algorithm based on Pinter’s approach which has the time complexity of $O(w^{3/2} \log w)$. Thus, the case of $CVM_k$ ($k \geq 3$) and maximum junction degree $\Delta(W) \leq 3$ has remained the only open problem.
2 Two-layer wirability and coherent problems

We first review the results on the decision problem as to whether a given wire layout is 2-layer wirable. They reveal some structural properties of 2-layer wirable wire layouts used in the remainder of this chapter. Secondly, we discuss the problem of how to minimally stretch a wire layout in order to make it 2-layer wirable. Then, we present the main results on the constrained via minimization problem.

2.1 Two-layer wirability

There are two different approaches for 2-layer wiring presented in literature. The first one was designed by Pinter [27]. The second one was independently presented by Naclerio, Masuda, and Nakajima [26] and by Molitor [22]. Here, the second approach – we call it odd-even approach – will be presented as it gives nice insights of the problem which are important for further investigations of wiring. The table below summarizes the results of the decision problem. \( w \) denotes the sum of the wire lengths of the wire layout.

<table>
<thead>
<tr>
<th>Decision Problem</th>
<th>knock-knee</th>
<th>generalized knock-knee</th>
<th>overlap mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>( O(w) )</td>
<td>( O(w) )</td>
<td>( O(w) )</td>
<td></td>
</tr>
</tbody>
</table>

2.1.1 Odd-even approach

Let \( W \) be any wire layout. For simplicity, assume that it is a generalized knock-knee mode wire layout. We will show how to extend the results to overlap mode wire layouts at the end of this section. Furthermore, we confine ourselves to the case that at most two different wires run over each grid point of the routing region – otherwise, the wire layout is not 2-layer wirable in any case.

An example illustrates the 2-layer wiring problem. Formal definitions of the notions can be found in [16, 22]. Let \( W \) be the wire layout shown in Figure 1.
Knock-knee mode wire layout $W$

$W$ divides the euclidian plan in which the routing region is embedded in elementary faces, 28 inner faces and one outer face. Note that actually the knock-knee nodes are handled as vertices of the graph. The inner faces are enumerated in Figure 1. Each face $r$ has a boundary. We call a face $r$ odd if its boundary is 2-layer wirable without inserting a (further) via. Otherwise, the face is called even. In our example face 1 is even because its boundary consists of exactly four cross-overs. At each of these points the layer has to be changed regardless of the wiring. Face 2 is odd because its boundary consists of three cross-overs and one junction. Obviously, a via has to be inserted at one of the five free grid points of the boundary of face 2 in order to wire it. Face 19 is even because there is one cross-over and one knock-knee, where the two adjacent wire segments do not belong to the same net. In Figure 2 all the odd faces of $W$ are shaded.
Obviously, it is impossible to find a 2-layer wiring without using a via if there is an odd face. In [22, 26] the converse is shown as well. Hence, there is a 2-layer wiring of $W$ needing no via if and only if each face of $W$ is even. It follows that in order to obtain a 2-layer wiring the odd faces have to be transformed into even ones by inserting vias on their boundaries. Here, we assume that a via is a vertex where the layer has to be changed. In our example, if a via is inserted, e.g., at the free grid point of the vertical wire segment between face 2 and face 3, face 2 which was odd becomes even and face 3 which was even becomes odd. The insertion of a via between face 3 and 4 transforms these two odd faces into even ones. To sum it up, we have transformed the two odd faces 2 and 4 into even ones by joining them by a path of vias. We have married face 2 to face 4.

It is easy to show that there is an even number of odd faces in any generalized knock-knee mode wire layout. Going once around any node of a given wire layout, the layer has to be changed an even number of times regardless of the wiring. Thus, the layer has to be changed an even number of times in the whole wire layout. The even faces contribute an even part to this number. Thus, the odd faces also contribute an even part which is only possible if there is an even number of odd faces in the wire layout.

So, marrying each odd face to exactly one other odd face results in a wire layout where each face is even. Conversely, any 2-layer wiring induces such a marriage of the odd faces. Figure 3 shows a 2-layer wiring of $W$. Here, faces 2 and 4, 13 and 16, 12 and 15, and 21 and 25 are joined. Note that the faces 21 and 25 are joined by a path of vias which runs across the outer face.

![Figure 3](image)

*Figure 3* Two-layer wiring of $W*
Let us now concentrate on the dual graph $G_d = (V_d, E_d)$ of a wire layout $W$, which is shown for our example wire layout in Figure 4. The set $V_d$ of the vertices consists of the faces of $W$. $\{f, f'\} \in V_d \times V_d$ is an edge of $E_d$ if and only if a via may be located on a grid point adjacent to the faces $f$ and $f'$. The corresponding wire segments are said to be free. The other ones are called critical wire segments.

**Figure 4** Dual graph $G_d$ of wire layout $W$

The dual graph of the wire layout $W$ of Figure 1 is obviously connected (in the graph theoretical sense). Therefore, there are perfect matchings of the odd faces, each defining a 2-layer wiring of $W$ and vice versa. Figure 5 shows a matching (given by the bold edges) of the odd faces (marked by black vertices) which corresponds to the 2-layer wiring of Figure 3.

**Figure 5** Perfect matching of the odd faces of wire layout $W$

The situation is a little more complex in the case of dual graphs consisting of more than one connected component. However, there is obviously a 2-layer wiring of a wire layout $W$ if and only if each connected component of the corresponding dual graph is even, i.e., if it contains an even number of odd faces. This remark is illustrated by Figure 6 where we consider a coarser grid than in the previous
figures. The dual graph $G_d'$ of the corresponding wire layout $W'$ consists of a lot of connected components. Two of them are odd so that 2-layer wiring of wire layout $W'$ is not possible.

Hence, this results in an algorithm to decide whether a generalized knock-knee mode wire layout is 2-layer wirable which runs in time $O(w)$.

### 2.1.2 Extension to overlap mode wire layouts

Now, let $W$ be a wire layout in overlap mode. Analogously to the case of generalized knock-knee mode wire layouts, we call a face even if its boundary is 2-layer wirable without using a via. Otherwise it is called odd. By eliminating overlapping wire segments by identifying adjacent grid points $p_1, p_2$ of the routing region $R$ whenever $\{p_1, p_2\}$ is an edge of the grid over which two wire segments run, it is easy to see that the algorithm proposed above can also be applied in the case of overlap mode wire layouts.

### 2.1.3 Extension to macro cell designs

Actually, the designs do not only consist of wire segments but contain basic cells and macro cells. Figure 7 shows a design which contains two macro cells $A_1$ and $A_2$. Assume that the pins of these macro cells are not preassigned to some layer. (Two-layer wiring with pin preassignments will be discussed in section 4.) Now, 2-layer wiring is performed by deleting the cells, first. The remaining design is a wire layout which can be wired as described above: compute the faces which are
odd and marry them. The example wire layout of Figure 7 contains 5 inner faces and one outer face after deletion of the cells $A_1$ and $A_2$. The three inner faces 1, 2, and 4 as well as the outer face are odd. The 2-layer wiring shown marries face 2 to face 4 and face 1 to the outer face.

![Diagram](image)

**Figure 7** Two-layer wiring of a macro cell design

### 2.2 Stretching to ensure 2-layer wirability

Assume that a given wire layout in the square grid is not 2-layer wirable. The question which arises is whether it is possible to stretch the wire layout, i.e., to insert new horizontal or vertical grid tracks in order to obtain a 2-layer wirable wire layout. We denote this decision problem by $d$-stretching$_2$ ($d = 1, 2$), when $d$-dimensional stretching is allowed. A wire layout is called 2-layer $d$-stretchable if it can be transformed into a 2-layer wirable wire layout by $d$-dimensional stretching. Stretching a wire layout increases the area. So, if a wire layout becomes 2-layer wirable by stretching it in appropriate locations, it is desirable to construct a minimum area 2-layer wiring, i.e., to insert the minimum number of new tracks necessary. We call this problem $d$-minstretching$_2$.

Again, the table below summarizes the results on computational complexities discussed in the following. Here, a decision problem is said to be trivial if either all instances of the problem have as solution the answer 'yes' or they all have the solution 'no'.

---

**Table**

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(n)$</td>
<td>$d$-stretching$_2$</td>
</tr>
<tr>
<td>$O(n^2)$</td>
<td>$d$-minstretching$_2$</td>
</tr>
</tbody>
</table>
Let us concentrate first on 1-stretching for the case of knock-knee mode wire layouts, i.e., we allow without loss of generality the insertion of new horizontal tracks in the routing region. The insertion of a new track generates new grid points where vias may be located. Now, call the statements on the structure of 2-layer wirable wire layouts to memory. Obviously, if a new track intersects an odd connected component, this component will be connected in the dual graph to the outer face. If all the odd components are intersected by new tracks, then the new dual graph does not contain an odd connected component any more as there was an even number of odd connected components. Hence, any knock-knee mode wire layout is 2-layer 1-stretchable. So, $d$-stretching is trivial for wire layouts in knock-knee mode. For illustration consider once more the wire layout in Figure 6 which is not 2-layer wirable. By inserting two new horizontal tracks 2-layer wirability is ensured. Figure 8 shows the new wire layout and its dual graph.

![New wire layout](image1)

![The corresponding dual graph](image2)

**Figure 8** Successful stretching of the wire layout of Figure 6

When wire segments are allowed to overlap, an odd connected component intersected by a new track does not have to be connected to the outer face in the
new dual graph. Overlapping wire segments prevent the two adjacent faces from being connected in the dual graph even if a new track is inserted at this position. Figure 9 illustrates this remark. The overlap mode wire layout shown contains exactly one odd inner face. The only new horizontal track which intersects this face does not connect it to the outer face (which is the other odd face of this wire layout) because of the two overlapping wire segments at the left and the right. Thus, \(d\)-stretching\(_2\) is not trivial for the case of overlap mode wire layouts. However, it is easy to see that a wire layout is 2-layer \(d\)-stretchable if and only if the wire layout which is obtained by maximally stretching it, i.e., by inserting a new track between every two grid tracks of the routing region, is 2-layer wirable. Hence, \(d\)-stretching\(_2\) is solvable in time \(O(w)\).

![Figure 9](image)

**Figure 9** Unsuccessful stretching of an overlap mode wire layout

Now, let us discuss the problem of minimally stretching a wire layout to ensure 2-layer wirability. Obviously, \(d\)-minstretching\(_2\) (\(d \in \{1, 2\}\)) is equivalent to the problem of finding a minimum number of tracks such that every connected component in the new dual graph is even. Thus, for the case of knock-knee mode wire layouts the problem we have to solve is to find a minimum number of tracks such that each odd connected component is intersected.

If only vertical stretching is allowed, ignore the horizontal dimension and represent the odd connected components by a set

\[
I_m = \{(i, j) \mid 0 \leq i < j \leq m\} \subset \mathbb{N}_0 \times \mathbb{N}_0
\]

of intervals. \(m\) denotes the number of horizontal tracks of the routing region. Hence, 1-minstretching\(_2\) is equivalent to the problem of finding a minimum set \(M\) such that for each \((i, j) \in I_m\) there is an \(x \in M\) with \(i < x < j\). This clique covering problem on interval graphs is shown to be solvable in runtime \(O(|I_m|)\) [14]. An optimal algorithm is given by
begin
    $M = \{\}$;
    $I = I_m$;
    $k = \min \{j \mid (i, j) \in I\}$;
    $M = M \cup \{k - \frac{1}{2}\}$;
    $I = I \setminus \{(i, j) \in I \mid i < k < j\}$;
    if $I \neq \{\}$ then goto (3);
end

Procedure minstretching

It follows that for the case of knock-knee mode wire layouts $1$-minstretching$_2$ is solvable in time $O(w)$.

For the case of overlap mode wire layouts the problem is harder, because an odd connected component intersected by a new track does not have to be connected to the outer face. Recently, Kaufmann and Molitor [14] have shown that for overlap mode wire layouts $1$-minstretching$_2$ is NP-hard by reducing 3SAT to it. To each boolean expression $E$ with $q$ variables, they construct a mintrack problem $P$ in overlap mode such that there is a solution for $P$ by inserting $q$ new tracks if and only if $E$ is satisfiable.

When 2-dimensional stretching is allowed, the mintrack problem for knock-knee mode wire layouts is equivalent to a minimum covering of rectangles. It can be formulated as follows: Given a set $Q$ of rectangles in the euclidian plane, find a minimum number of vertical and horizontal lines such that each rectangle of $Q$ intersects at least one of these tracks. Again, 3SAT can be reduced to this covering problem [14]. Hence, $2$-minstretching$_2$ is NP-hard even for knock-knee mode wire layouts.

All these results completely close the gap between the polynomially solvable cases and NP-complete cases for the 2-layer stretching problem.

### 2.3 The constrained via minimization problem

If a wire layout is 2-layer wirable one looks for a best 2-layer wiring. The goal is to optimize the performance of the circuit and possibly minimize its manufacturing costs. In this context, it is important to minimize the number of vias since excess vias lead to decreased performance of the electrical circuit, decreased yield of the
manufacturing process and increased amount of area required for interconnections. This is the most popular version which is called constrained via minimization problem denoted by CVM$_2$ when 2 layers are available.

The table below summarizes the results on CVM$_2$ which close the gap between the polynomially solvable cases and NP-complete cases. The results hold for wire layouts in knock-knee mode and in overlap mode.

<table>
<thead>
<tr>
<th></th>
<th>$\Delta(W) \leq 3$</th>
<th>$\Delta(W) = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVM$_2$</td>
<td>$O(w^{3/2} \log w)$</td>
<td>NP</td>
</tr>
</tbody>
</table>

2.3.1 Constrained via minimization in the case of 3-way junctions

As already shown, there is a one-to-one correspondence between 2-layer wirings and marriages of the odd faces of a wire layout $W$. Hence, CVM$_2$ for a wire layout $W$ is equivalent to the problem of finding an optimal marriage of the odd faces, i.e., a marriage where a minimal number of grid points are used by the wedding paths.

For the case of $\Delta(W) \leq 3$, i.e., that there is no $q$-way junction with $q > 3$ in $W$, it is easy to see that the sum of the lengths of the wedding paths of a marriage of the odd faces equals the number of vias which have to be inserted by the corresponding 2-layer wiring. Therefore, this restricted version of CVM$_2$ is equivalent to the problem of finding a marriage where the sum of the lengths of the wedding paths is minimal. Formally, let $ODD$ be the set of the odd faces of a wire layout $W$, the problem is to find a disjoint partition of $ODD$ into pairs $\{f_1, f_2\}, \ldots, \{f_q, f_{q_2}\}$ such that

$$\sum_{i=1}^{q} \text{dist}(f_{i_1}, f_{i_2})$$

is minimal. Here, $\text{dist}(g, g')$ is the number of edges of the minimum length path connecting face $g$ and face $g'$ in the dual graph $G_d$. The best known algorithm for solving this minimum weighted perfect matching problem has worst time complexity $O(w^3)$ [6]. This reduction has been independently presented by Naclerio, Masuda and Nakajima [26] and Molitor [22]. It was the first polynomial time
algorithm for $CVM_2$ for the case $\Delta(W) \leq 3$. At present, the best known algorithm for this restricted version of $CVM_2$ is that presented by Kuo, Chern and Shih [17] which is based on Pinter’s approach for 2-layer wiring [27].

2.3.2 Constrained via minimization for the case of 4-way junctions

Intuitively, when $\Delta(W) = 4$ is allowed, $CVM_2$ seems to be harder. There are wire layouts such that in every optimal marriage of the odd faces there is a grid point shared by two different wedding paths. An example is given in Figure 10. The wire layout shown in Figure 10 contains one 4-way junction. Thus, there are four odd faces. Marrying these four faces results in two wedding paths. Thus, the sum of the lengths of these paths is at least 2. Nevertheless, inserting one via at the junction itself converts the odd faces into even ones by marrying face 1 to face 3, and face 2 to face 4. Hence, an algorithm solving the above minimum weighted perfect matching problem does not solve $CVM_2$ in general.

![Figure 10](image-url)  
(a) Wire layout which contains a 4-way junction  
(b) 2-layer wiring requiring the insertion of one via

In fact, Choi, Nakajima and Rim [4] could show that $CVM_2$ is NP-complete when the maximum junction degree is limited by four. They reduce the vertex-deletion graph bipartization problem (denoted by VDB) to $CVM_2$. Given a graph $G = (V, E)$ and an integer $q \geq 0$, VDB is the problem of finding a set $V'$ of $q$ or fewer vertices such that the subgraph $G'$ obtained by deleting all vertices of $V'$ (and the adjacent edges) from $G$ is bipartite. The crucial idea behind is based on the fact that in order to obtain a 2-layer wiring, any odd face (a boundary of an odd face is not bipartite) has to be transformed into an even one by inserting a new vertex, namely a via (the boundary is bipartite after this transformation). Note that, in this context, inserting a via on a wire segment is equivalent to deleting
the wire segment. In [4], VDB is shown to be NP-complete even when $G$ is planar and the maximum degree is limited by four.

3 Wiring for interconnect delay minimization

Unfortunately, the solutions of the CVM$_2$ problem are global (cf. [27]), i.e., in the process of minimizing the total number of vias we may burden one net with an excessive number of vias, or embed delay-critical nets in layers with poor conductivity. Since in some VLSI chips the optimization of the electrical performance of interconnections is more important than the minimization of the number of vias, wiring algorithms are needed which minimize signal delays through interconnection lines. This problem – we call it performance driven 2-layer wiring (PDW$_2$) – was recently formulated by Ciesielski [5]. He investigates the case of two available layers and generalizes Pinter’s approach [27] in order to handle the problem. Unfortunately, his reduction results in a maximum cut problem for nonplanar graphs, which is known to be NP-hard. It remained open whether the problem itself is NP-hard.

This section presents some new results on PDW$_2$. We will discuss two restricted problems of PDW$_2$. First, we handle the case that all the layers have the same conductivity. Then, we discuss the case of two layers which have different conductivities. We present formal approaches to both problems raising hopes for good heuristics despite of their NP-hardness.

3.1 Two layers with same conductivity

Assume that the two layers have same conductivity. Because a via decreases the performance of the electrical circuit, i.e., increases the corresponding net delay, we have to investigate the following problem in order to solve this restricted version of PDW$_2$, which we denote by PDW(=)$_2$.

**Instance:** Let $W = \{W_1, \ldots , W_p\}$ be a 2-layer wirable wire layout and let $v = (v_1, \ldots , v_p) \in \mathbb{N}_0^p$ be a vector consisting of $p$ nonnegative numbers.

**Find** a 2-layer wiring of $W$ such that for any $i \in \{1, \ldots , p\}$ the number $\alpha_i$ of vias inserted on wire $W_i$ is less than or equal $v_i$.
From the theoretical point of view, this problem can be formulated as follows using the odd-even approach. We assign a vector \( \alpha(e) = (\alpha(e)_1, \ldots, \alpha(e)_p) \in \{0, 1\}^p \) of dimension \( p \) to each edge \( e \) of the dual graph \( G_d \) with \( \alpha(e)_j = 1 \) if and only if edge \( e \) corresponds to a wire segment of \( W_j \). Obviously, one can now assign a vector \( \alpha(P) = (\alpha(P)_1, \ldots, \alpha(P)_p) \) to each path \( P = (e_1, e_2, \ldots, e_r) \) in the dual graph by adding up the weight vectors of the corresponding edges, i.e.,

\[
\alpha(P)_j = \sum_{k=1}^{r} \alpha(e_k)_j.
\]

Then the problem we have to solve is the following one.

**Instance:** Let \( W = \{W_1, \ldots, W_p\} \) be a wire layout containing \( 2q \) odd faces and let \( v = (v_1, \ldots, v_p) \in \mathbb{N}_0^p \) be a vector consisting of \( p \) nonnegative numbers.

**Find** \( q \) simple paths \( P_1, \ldots, P_q \) in the dual graph of \( W \) marrying the \( 2q \) odd faces such that for all \( j = 1, \ldots, p \) the inequation

\[
\sum_{k=1}^{q} \alpha(P_k)_j \leq v_j
\]

holds.

As shown in [15], the problem is equivalent to the problem

**Instance:** Let \( G = (V, E) \) be a planar graph and let \( \alpha : E \rightarrow \{0, 1\}^p \) a function assigning a boolean vector \( \alpha(e) \) of dimension \( p \) to each edge \( e \). Furthermore, let \( a, b \) be two distinguished vertices in \( V \), and let \( v = (v_1, \ldots, v_p) \in \mathbb{N}_0^p \) be a vector consisting of \( p \) nonnegative numbers.

**Find** a path \( P \) connecting \( a \) and \( b \) such that \( \alpha(P)_j \leq v_j \) holds \( \forall j = 1, \ldots, p \)

for the case \( q = 1 \), i.e., if \( W \) only contains two odd faces. This problem has been shown to be NP-hard [15]. It remains NP-hard even if the maximum junction degree of the wire layouts is limited to three. Nevertheless the above formulation of the problem reveals a lot of structural properties of the problem giving hopes for efficient heuristics.
3.2 Two layers with different conductivities

For handling wiring for the case of two layers with different conductivities, we first consider the problem for two layers $L_1$ and $L_2$ which have the following two properties. First, the conductivity of $L_1$ is extremely better than the conductivity of $L_2$. Secondly, the costs (with respect to signal delays) of a via are much lower than the costs caused by embedding one unit of the corresponding wire in the poor layer $L_2$. Here, unit length is defined to be the distance between two adjacent grid points, assuming that the routing region is a square grid. Thus, if two different wires $W_i$ and $W_j$ share a common grid point of the routing region, either two units of wire $W_i$ or two units of wire $W_j$ have to be put in the poor layer $L_2$.

3.2.1 The general case

Let $\delta$ be a 2-layer wiring of a wire layout $W = \{W_1, \ldots, W_p\}$. We denote by $u_i(\delta)$ the number of unit lengths of wire $W_i$ which are embedded in the poor layer by $\delta$. Then, the problem – we will denote it by PDW($\neq$) – we have to handle is equivalent to

**Instance:** Let $W = \{W_1, \ldots, W_p\}$ be a 2-layer wirable wire layout and let $v = (v_1, \ldots, v_p) \in \mathbb{N}_0^p$ be a vector consisting of $p$ nonnegative numbers.

**Find** a 2-layer wiring $\delta$ of $W$ such that for all $i \in \{1, \ldots, p\}$ the inequation $u_i(\delta) \leq v_i$ holds.

For illustration we need some further definitions (see Figure 11). Let a cluster be defined to be a maximal set of mutually crossing and overlapping critical wire segments (cf. [27]). Obviously, in each cluster, once a wire segment is assigned to a certain layer, wiring of the rest of the cluster is forced as no via may be inserted. Thus, there are two possible 2-layer wirings of every cluster of a wire layout, i.e., for any cluster $c_q$, there are two states which we denote by state 0 and state 1. Each state $j$ of $c_q$ is associated with a weight vector $d_{q,j} = (d_{q,1}, \ldots, d_{q,p})$ of dimension $p$ in which the $i$th component $d_{q,j}^i$ is the number of units of wire $W_i$ which are embedded in the poor layer $L_2$ if cluster $c_q$ is in state $j$.

Actually, this approach can be used to manage the exact relation between the conductivities of the two layers available, too. The definition of the weight vectors for both states of a cluster have to be extended by taking into consideration not
only the wire segments assigned to the poor layer $L_2$ but also those assigned in the preferred layer $L_1$ and the vias inserted. Free wire segments are considered as trivial clusters.

![Diagram](image_url)

**Figure 11** A cluster with its two dual states and weight-vectors

Now, the problem can be stated in the following way.

**Instance:** Given a wire layout $W = \{W_1, \ldots, W_p\}$ consisting of $l$ clusters $c_1, \ldots, c_l$, and let $v = (v_1, \ldots, v_p) \in \mathbb{N}_0^p$ be a vector consisting of $p$ nonnegative numbers.

**Assign** a state $j(q)$ to any cluster $c_q$ such that for all $i \in \{1, \ldots, p\}$ the inequation

$$\sum_{q=1}^{l} d_{q,i}^{j(q)} \leq v_i$$

holds.

As shown in [15], this problem is NP-hard even if the maximum junction degree of the wire layout is limited to three. However, it can be reduced to an integer valued generalized flow problem [7, 13]. The reduction, which is presented in [15], raises hope for good heuristics. First, compute the maximal real-valued flow value. This value can be computed by applying the polynomial time algorithm of Goldberg et. al. [7, 8]. Convert this maximal real-valued flow to an integer-valued one by using randomized rounding. The results of Raghavan [28] who studied this method for maximum multicommodity flow let us hope that we get solutions close to the optimum. For more details, please refer to [15].
3.2.2 A polynomial time algorithm for a special case

Now, let us consider the case that the wire layout is in knock-knee mode and the routing grid is very fine, i.e., that it is possible to place a via between every two adjacent cross-overs and/or knock-knees. Thus, in order to obtain a 2-layer wiring, it is sufficient to construct a bridge for every cross-over and knock-knee by putting two units of one of the two wires in the poor layer $L_2$; all other wire segments can be embedded in the preferred layer $L_1$. Figure 12 illustrates this remark.

![Figure 12 Illustration of bridges](image)

Obviously, this special case of PDW($\neq$)$_2$ is equivalent to the problem of not burdening one signal net with an excessive number of bridges. Thus, the problem can be formulated as follows:

**Instance:** Let $W_1, \ldots, W_p$ be the routed signal nets of a given wire layout $W$ in knock-knee mode, $v = (v_1, \ldots, v_p) \in \mathbb{N}_0^p$ be a vector consisting of $p$ nonnegative integers, and let $c(\{i, j\}) \in \mathbb{N}_0$ be the number of grid vertices commonly shared by $W_i$ and $W_j$.

**Find** for all $i \in \{1, \ldots, p\}$, nonnegative numbers $c^{(i)}(\{i, j\}) \in \mathbb{N}_0$ such that

$$
c^{(i)}(\{i, j\}) + c^{(j)}(\{i, j\}) = c(\{i, j\})
$$

for all $i, j \in \{1, \ldots, p\}$, and

$$
\sum_{j=1}^{p} c^{(i)}(\{i, j\}) \leq v_i
$$

for each $i \in \{1, \ldots, p\}$.

$c^{(i)}(\{i, j\})$ denotes the number of times wire $W_i$ ($W_j$) is put in the poor layer $L_2$ when sharing a common grid point with wire $W_j$ ($W_i$). Note that $c(\{i, i\}) = 0$. 
This special case of PDW($\neq 2$) can be reduced to a maximum integer network flow problem which can be solved in polynomial running time. We start with some definitions (cf. [20]). A directed network $N = (V, E, \text{cap})$ is given by a directed graph $G = (V, E)$ and a capacity function $\text{cap} : E \rightarrow \mathbb{N}_0$. Let $s, t \in V$ be two designated vertices, the source $s$ and the sink $t$. A function $f : E \rightarrow \mathbb{N}_0$ is a legal $(s, t)$-flow if it satisfies the capacity constraints

$$0 \leq f(e) \leq \text{cap}(e)$$

for all $e \in E$, and the conservation laws

$$\sum_{e \in \text{in}(v)} f(e) = \sum_{e \in \text{out}(v)} f(e)$$

for all $v \in V \setminus \{s, t\}$ where $\text{in}(v)$ (out$(v)$) is the set of edges entering (leaving) $v$. If $f$ is a legal flow then

$$\text{val}(f) = \sum_{e \in \text{out}(s)} f(e) - \sum_{e \in \text{in}(s)} f(s)$$

is the flow value of $f$. The maximum network flow problem is to compute a legal flow with maximum flow value. As shown in [8] a maximum flow from $s$ to $t$ can be computed in time $O(ne \log \frac{n^2}{e})$ in any directed network $N = (V, E, \text{cap})$ with $n = |V|$ and $e = |E|$.

Now, let us reduce our problem to a bipartite maximum network flow problem $(V, E, \text{cap})$. $V$ is partitioned into the set $V_n = \{W_1, \ldots, W_p\}$ representing the signal nets, the set $V_c = \{c_{i,j} \mid 1 \leq i < j \leq p\}$ and the two vertices $\{s, t\}$, i.e., $V = \{s\} \cup V_c \cup V_n \cup \{t\}$. There is an edge $e'_{i,j} \in E$ from $s$ to vertex $c_{i,j} \in V_c$ with capacity $\text{cap}(e'_{i,j}) = c(\{i, j\})$. Edges $e^{\{i,j\}}_i$ and $e^{\{i,j\}}_j$ with capacities $c(\{i, j\})$ connect vertex $c_{i,j}$ to the vertices $W_i$ and $W_j$, respectively. Furthermore, from every vertex $W_i \in V_n$ there is an edge $e'_i$ to vertex $t$ with $\text{cap}(e'_i) = v_i$. Figure 13 illustrates this construction. The edges are marked by $<\text{name}>/<\text{capacity}>$. 
It is easy to see that there is a legal flow with flow value
\[
\sum_{1 \leq i < j \leq p} c(\{i, j\})
\]
if and only if there are nonnegative numbers \(c^{(i)}(\{i, j\}) \in \mathbb{N}_0\) such that
\[
c^{(i)}(\{i, j\}) + c^{(j)}(\{i, j\}) = c(\{i, j\})
\]
and
\[
\sum_{j=1}^{p} c^{(i)}(\{i, j\}) \leq v_i
\]
hold for all \(i, j \in \{1, \ldots, p\}\) in the given instance of the PDW(\(\neq\))_2 problem. If a maximum flow \(f\) with value \(\sum_{1 \leq i < j \leq p} c(\{i, j\})\) does exist, then \(f(e_{i,j}^{(i,j)})\) specifies the number of times wire \(W_i\) has to be put in the poor layer \(L_2\) when sharing a common grid point with wire \(W_j\).

Actually, if the wires \(W_i\) and \(W_j\) do not share any common grid point, \(c_{\{i,j\}}\) needs not be a vertex of the network as \(c(\{i, j\})\) equals 0. Thus, the running time strongly depends on the number \(q\) of pairs of signal nets which share a common grid point. By applying the results shown in [8] it is easy to see that the running time of the proposed algorithm is \(O(K^2 \log K)\) for \(K = \max\{q, p\}\).

4 Hierarchical wiring

Another development in wiring we want to discuss is hierarchical wiring. Today’s integrated circuits have up to several hundred thousand transistors. Processing
such designs in a naive manner requires very large internal representations with some millions of data so that even linear space (time) optimization algorithms can be too expensive. However, large designs have a regular structure. In such arrangements, there are lots of identical subcircuits so that the designs can be described by small hierarchical representations. During synthesis, it is desirable to handle all instances of a subcircuit identically in order to guarantee identical electrical behavior of all instances, to allow further hierarchical processing and to decrease the running time of the synthesis itself.

Hierarchical physical synthesis is not highly developed yet. Results on hierarchical compaction can be found in [19, 29]. Results concerning the hierarchical physical design of systolic arrays are presented in [21, 23]. Some new results taking advantage of buses during hierarchical wiring appeared in [24].

Here we address the following problem. Let $A$ be a circuit composed of basic cells whose input and output pins lie in a certain but fixed layer, either $L_1$ or $L_2$. Assume that the placement and routing phase is already completed. Find a 2-layer wiring of the wire segments of $A$ such that each wire segment is embedded either into $L_1$ or $L_2$, the pins of the basic cells lie in the preassigned layers, and the number of vias is minimal on these two conditions. A solution of this problem which we denote by CVMPP (Constrained Via Minimization with Pin Preassignment) induces a hierarchical bottom-up 2-layer wiring algorithm which preserves the original layout hierarchy of the circuit.

We start with some definitions we need in the following. A circuit or a macro cell is said to be of level 1 if it contains no macro cell but only basic cells, and is said to be of level $i + 1$ if it contains at least one macro cell of level $i$ and no macro cell of level greater than $i$.

Now, assume that $A$ is a circuit of level $i$ composed of basic cells and macro cells $A_1, \ldots, A_q$ which are interconnected by some wire layout. Let $\delta_{A_i}$ be a 2-layer wiring of $A_i$ for $i \in \{1, \ldots, q\}$. A wiring $\delta_A$ of $A$ is said to be induced by the wirings $\delta_{A_1}, \ldots, \delta_{A_q}$ if $\delta_A$ is given by applying $\delta_X$ on all occurrences of $X$ ($X \in \{A_1, \ldots, A_q\}$) and then realizing the 2-layer wiring of the interconnecting wire layout. Thus, we have to consider the following problem which is applied as local step by the bottom up wiring algorithm.
Given 2-layer wirings $\delta_{A_1}, \ldots, \delta_{A_q}$ of the $q$ macro cells $A_1, \ldots, A_q$.

Find a 2-layer wiring $\delta_A$ of circuit $A$ induced by the wirings $\delta_{A_1}, \ldots, \delta_{A_q}$ which has no more vias than any other 2-layer wiring $\delta'_A$ of $A$ which is also induced by the wirings $\delta_{A_1}, \ldots, \delta_{A_q}$.

A first suggestion for solving CVMPP in polynomial time has been made by Pinter [17, 27]. His procedure however is only applicable if the preassignments are restricted to exterior pins (located at the outer border of the routing area), a condition normally not fulfilled by hierarchical circuit representations. Grötschel et al. [10] generalized Pinter’s procedure to deal with arbitrary pin preassignments. They reduce CVMPP to the maxcut problem for almost planar graphs, i.e., graphs which can be transformed into planar graphs by removal of only one node. As this maxcut problem has been shown to be NP-hard [1], CVMPP is not solved exactly in [10] but attacked by approximation techniques from combinatorial optimization. Grötschel’s procedure implied the question as to whether it is adequate to solve CVMPP by heuristics. Or, does there exist a fast exact algorithm? In [25] CVMPP has been proved to be NP-hard, thus shattering hopes for a fast exact solution. The proof is done by showing the opposite direction to Grötschel’s reduction, i.e., reducing maxcut for almost planar graphs to CVMPP.

Clearly, the fact that CVMPP is NP-hard in general does not exclude exact polynomial time algorithms for important special cases. Here we will characterize such an important subclass of wiring problems. We restrict to designs with power supply nets routed in layers $L_1$ and $L_2$. The routing of these nets is extremely critical because of voltage drop and current density constraints, and thus, vias on these wire segments are much more expensive than signal wire vias. As a consequence, power and ground wiring is done in advance, without consideration of signal nets. The subsequent signal wiring phase then has to meet the fixed wiring of power supply lines as well as the preassignment of the macro cell pins. This strategy will considerably simplify the solution of CVMPP as shown in [25].

Consider the macro cell design of Figure 7. Assuming a bottom-up synthesis process, the pins of the macro cells $A_1$ and $A_2$ are preassigned to certain layers. Figure 14 shows our example wire layout of Figure 7 extended by pin preassignments and wired power supply lines. Note that the pin preassignments of the southern pins of macro cell $A_1$ induce the wiring of some further wire segments
because of the absence of free wire segments, and that the wiring of the power supply nets induces the wiring of each wire segment which shares a common grid point with them.

![Diagram](image)

**Figure 14**  Macro cell design with preassignments

Now, in order to solve the problem, we substitute each macro cell by two critical wire segments which cross each other. Thus, they have to be wired in different layers. The first (second) one crosses each of the signal and power supply pins of the corresponding deleted macro cell which are preassigned to layer $L_1$ ($L_2$); thus, if the wiring meets the preassignments, this new wire has to be totally wired in layer $L_2$ ($L_1$) as no via is allowed to be inserted. The transformation is illustrated in Figure 15.

![Diagram](image)

**Figure 15**  Transformation to a wiring problem
After this transformation step, the results of section 2 are applied, i.e., the odd faces, which are marked by shades in Figure 15, the corresponding dual graph, and an optimal marriage of the odd faces are computed. Note once more that the new wire segments modelling the preassignments are critical, i.e., that no via may be inserted on them. Figure 16 shows the dual graph, the odd faces marked by black vertices, and an optimal marriage represented by bold lines. Obviously, there exist two different 2-layer assignments to this optimal marriage which are dual. Both are shown in Figure 17. The first one meets all the preassignments whereas the second one does not meet any of the preassignments. It can be proved that this is the general case, i.e., that one of the resulting dual wirings meets all the preassignments, as the preassigned wire segments are connected (in the graph theoretical sense) to each other through critical wire segments in the transformed wire layout. Those paths connecting wire segments which have to be wired in the same layer (different layers) have even parity (odd parity), i.e., the layers have to be changed an even (odd) number of times on these paths.

Because of the discussions of section 2, the number of vias inserted is minimal on the preassignment condition, obviously. The algorithm presented can be applied bottom up to boxes of level 2, 3, 4 and so on. This results in a bottom up hierarchical 2-layer wiring algorithm which locally minimizes the number of vias at each level, and which preserves the hierarchical structure of the specification of the circuit.
Actually, the general problem of hierarchical wiring with respect to via minimization is harder than presented here. It can be formulated in the following manner.

**Instance:** Given a box $A$ of level $i$ which is composed of macro cells $A_1, \ldots, A_q$.

**Find** 2-layer wirings $\delta_{A_1}, \ldots, \delta_{A_q}$ of the $q$ macro cells such that there is a 2-layer wiring $\delta_A$ of $A$, which is induced by them, with a minimal number of vias compared to any other 2-layer wiring $\delta'_A$ of $A$ which is induced by any other 2-layer wirings $\delta'_{A_1}, \ldots, \delta'_{A_q}$.

This problem is still open.

5 References


5 REFERENCES


