# EFFICIENT OPTIMAL VIA SHIFTING ALGORITHMS IN CHANNEL COMPACTION*†

Yang Cai and Martin D. F. Wong

Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712-1188

TR-91-31 October 1991

## ABSTRACT

We consider in this paper the problem of shifting vias to obtain more compactable channel routing solutions. Let $S$ be a grid-based two-layer channel routing solution. Let $v_c$, $w_c$ be the number of grid points on column $c$ that are occupied by vias, horizontal wires in $S$, respectively. We define the *expected height* of column $c$ in $S$ to be $h_c = Av_c + Bw_c + C$, where $A, B, C$ are some design rule dependent constants. A column in $S$ is said to be a *critical column* of $S$ if it has maximum expected height among all columns in $S$. Let $H_S = \max_c h_c$ be the expected height of the critical column(s) of $S$. In general, $H_S$ is a good measure of the height of $S$ after compaction. We show that the problem of shifting vias to minimize $H_S$ can be solved optimally in polynomial time.

**Keywords:** Computer-aided design, Channel routing, Channel compaction, Via shifting, Expected channel channel height.

---

* This work was partially supported by the National Science Foundation under grant MIP-8909586, by an IBM Faculty Development Award, and by an ACM SIGDA scholarship.

† Preliminary versions of this paper appeared in [3, 4].

# 1 Introduction

In VLSI layout design, a significant portion of the chip area is used for channel routing. There are several grid-based two-layer channel routers which can consistently produce routing solutions that are at most one or two tracks within optimal solutions [2, 9, 13, 16, 24]. Recent studies [5, 6, 7, 8, 22] showed that the routing solutions of these routers could be compacted to obtain further area reduction. From experimental results of different channel compactors, it was observed that the amount of area reduction is closely related to both the channel routing solution and the design rules used. Appropriate modification of a given channel routing solution could result in a significant amount of routing area reduction. Techniques developed to modify channel routing solutions include *via shifting* [5, 8], *via offsetting* [5, 8, 23], *track permutation* [6, 7, 22], *local rerouting* [6, 7, 20, 21], and *via minization* [5, 20, 21]. In this paper, we will focus our attention on the technique of via shifting.

In a grid-based two-layer channel routing solution, the vias typically have some freedom to move along some rectilinear graphs (rectilinear trees if we assume there are no redundant wires in the channel routing solution) without violating design rules, as illustrated in Figure 1. In this example, the rectilinear graph associated with each via along which the via can move is indicated by heavy lines. If the rectilinear graphs associated with two vias share a common grid point, then the vias can be merged into a single via by changing the layer assignment of some wires. For example, vias 1 and 2 in Figure 1 can be merged into a single via, so are vias 3 and 4. These vias are said to be *mergeable.*

Given a grid-based two-layer channel routing solution $S$, we can obtain a new routing solution $S'$ by simply shifting the vias in $S$ without violating design rules. In this case, $S'$ is said to be *derivable* from $S$. For example, the routing solution $S'$ in Figure 2(b) is derivable from the routing solution $S$ in Figure 2(a). In current fabrication technologies, wires are usually narrower than vias, and hence the positions of the vias in a channel routing solution can affect the final routing area after compaction. For example, consider the two routing solutions $S$ and $S'$ in Figure 2 with the following design rules: via height = 2.0, wire width = 1.0, and minimum spacing between adjacent *features* (vias and wires) = 1.0. Figure 3(a) and 3(b) show the compacted results of $S$ and $S'$, respectively. In this case, by shifting the vias in $S$, we obtained a reduction of channel height from 9.0 to 7.0. In general, for a given channel routing solution $S$, there are many different channel routing solutions which are derivable from $S$. This paper addresses the problem of how to obtain a channel routing

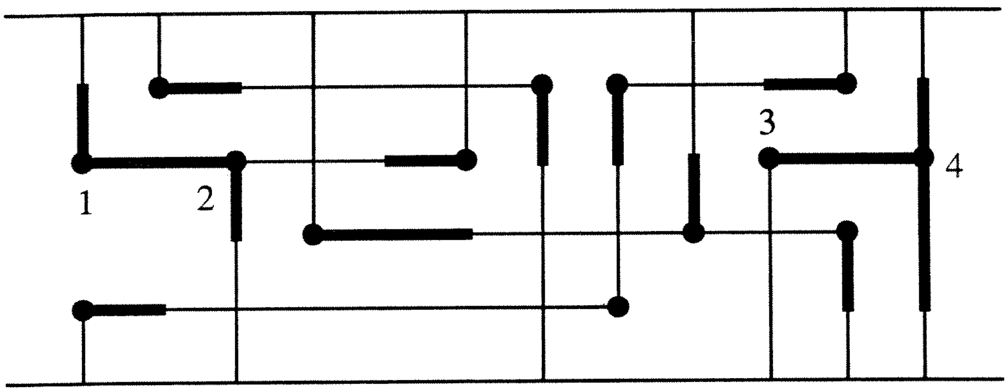



Figure 1: Shifting vias in a channel routing solution

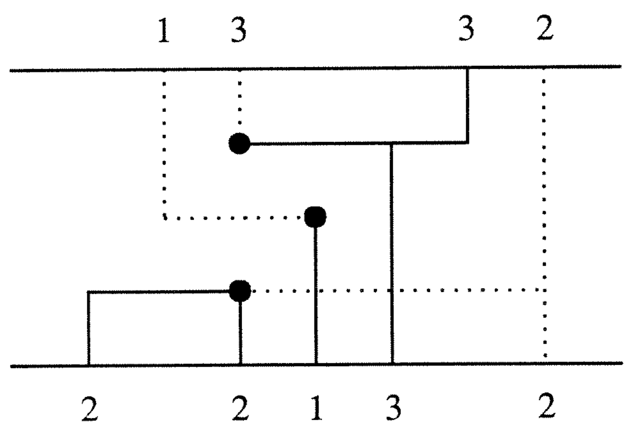
solution $S'$ from a given solution $S$ by shifting vias so that $S'$ achieves minimum channel height after compaction.

The rest of this paper is organized as follows. Section 2 introduces our measure of channel height after compaction. Section 3 studies the effects of via shifting. Basic concepts and preliminary results are presented in Section 4. Section 5 defines a boolean procedure which given a grid-based two-layer channel routing solution $S$ and a target expected height $H$, determines whether there is a channel routing solution derivable from $S$ with expected height $\leq H$. Two implementations of the procedure are given, one is more robust while the other is more efficient. Section 6 presents our optimal via shifting algorithm. Finally, Section 7 concludes the paper with some general remarks.

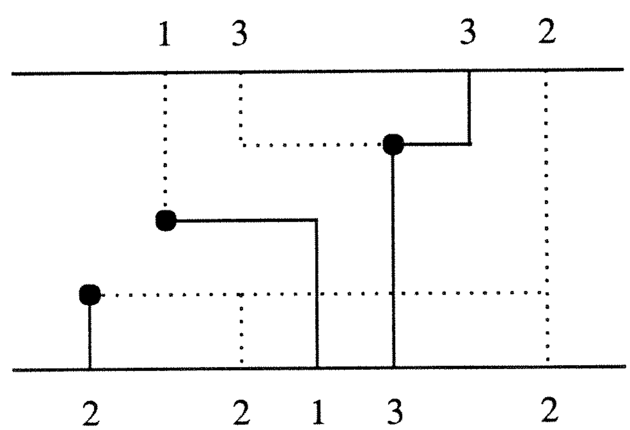
## 2 A Measure of Channel Height after Compaction

Let $v_c$, $w_c$ be the number of grid points on column $c$ in a grid-based two-layer channel routing solution $S$ that are occupied by vias, horizontal wires, respectively. We shall refer to $v_c$, $w_c$ as the *via count*, *wire count* at column $c$ in $S$, respectively, and refer to $f_c = v_c + w_c$ as the *feature count* at column $c$ in $S$. Let $\alpha$ be the height of a via, $\beta$ be the width of a wire, and $\gamma$ be the minimum spacing between adjacent features. According to current fabrication technologies, $\alpha > \beta$. (For example, the values of $\alpha$, $\beta$ and $\gamma$ used in [5, 6, 7, 8] are $\alpha = 2.0$, $\beta = \gamma = 1.0$.) There are three major factors that contribute to the height of column $c$ in a channel routing solution $S$ after compaction, namely,

1. The sum of the heights of vias on column $c$ in $S$, *i.e.*, $\alpha v_c$;

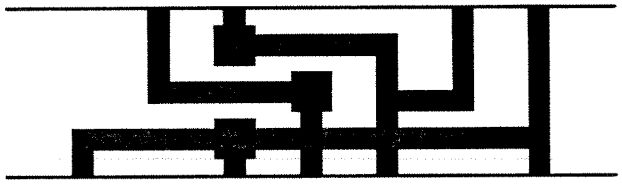



(a) The original solution $S$

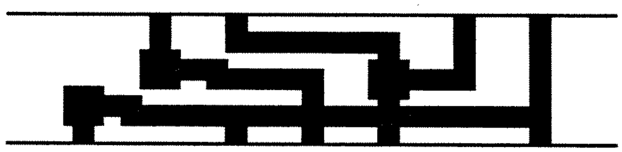



(b) A derivable solution $S'$

Figure 2: Derivable channel routing solutions



(a) The original solution $S$



(b) A derivable solution $S'$

Figure 3: Compaction of channel routing solutions

2. The sum of the widths of horizontal wires on column $c$ in $S$, *i.e.*, $\beta w_c$, and

3. The total minimum spacings between adjacent features on column $c$ in $S$, *i.e.*, $\gamma(v_c + w_c + 1)$.

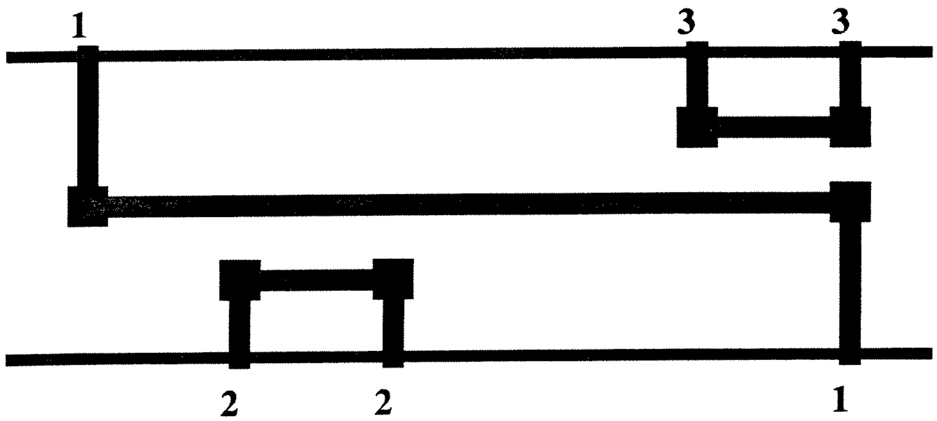
Therefore, we define the *expected height* of a column $c$ in $S$ to be

$$\begin{aligned} h_c &= \alpha v_c + \beta w_c + \gamma(v_c + w_c + 1) \\ &= (\alpha + \gamma)v_c + (\beta + \gamma)w_c + \gamma \\ &= A v_c + B w_c + C, \end{aligned}$$

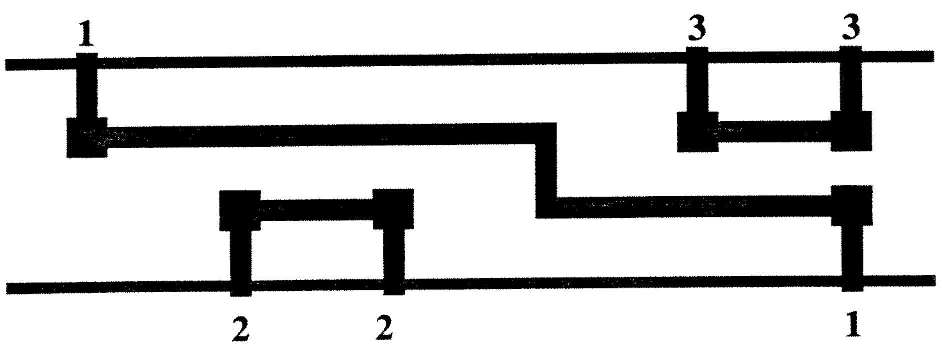
where $A = \alpha + \gamma$, $B = \beta + \gamma$, and $C = \gamma$, are design rule dependent constants. It can be seen from the above reasoning that $h_c$ is an estimation of the height of column $c$ in $S$ after compaction.

A column of a channel routing solution $S$ is said to be *critical* if it has maximum expected height among all columns of $S$. Let $H_S = \max_c h_c$ be the height of the critical column(s) of $S$. In general, $H_S$ is a good measure of the height of $S$ after compaction, and will be referred to as the *expected height* of $S$. For example, the channel routing solution shown in Figure 4(a) has expected height 7 (using $\alpha = 2.0$, $\beta = \gamma = 1.0$.), and the actual height of the channel after compaction (Figure 4(b)) is also 7.

Note that shifting vias can change the expected height of a channel routing solution, as illustrated in Figure 5. After shifting the two vias to the new positions indicated by the two arrows, the expected height of the new channel routing solution is $3(\alpha + \beta)$, while the




(a) A channel routing solution




(b) A compacted channel

Figure 4: A measure of channel height after compaction

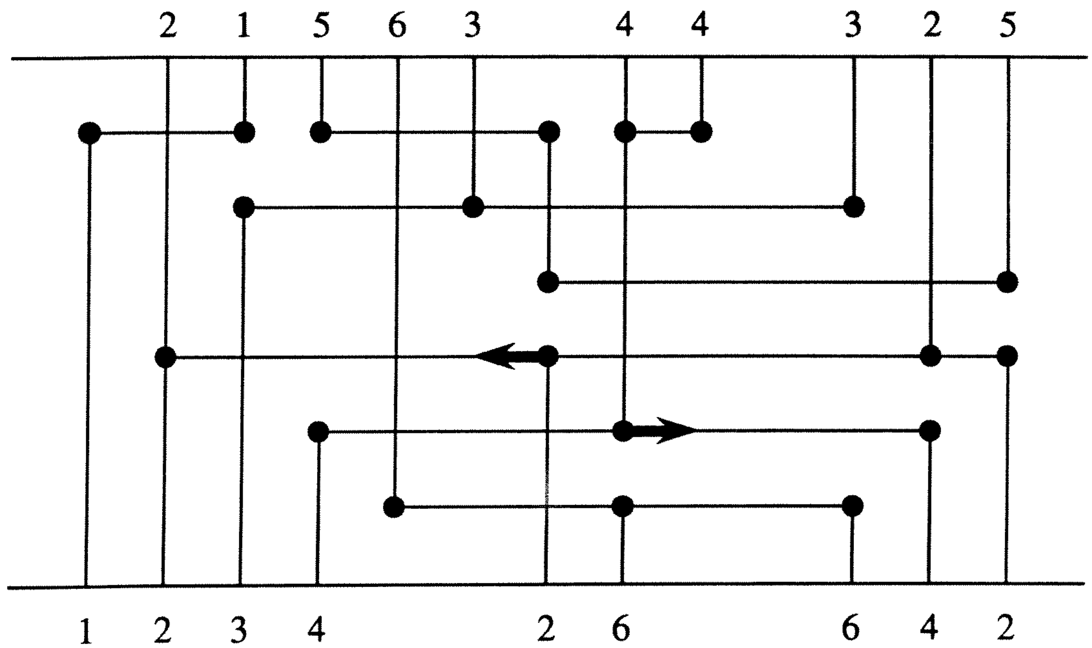



Figure 5: Shifting vias to reduce the expected height of a channel

expected height of the original channel routing solution is $2\alpha + 4\beta$, a difference of $\alpha - \beta$. The main contribution of this paper is efficient algorithms that compute a channel routing solution with minimum expected height derivable from a given channel routing solution.

# 3 Effects of Via Shifting

In this section we study the effects of shifting the vias in a grid-based two-layer channel routing solution on the expected height of the columns of the channel.

Figure 6 depicts different ways of shifting vias. Note that via shifting may introduce overlappings of wires of the same net, but never introduces overlappings of wires of different nets. We consider the following two kinds of via shiftings.

1. *Horizontal via shifting.*
   Suppose we shift a via $v$ in a channel routing solution $S$ from column $c$ to another column to obtain another channel routing solution $S'$. Let $v_c'$, $w_c'$ and $f_c'$ be, respectively, the via count, wire count and feature count at column $c$ in $S'$, and let $h_c'$ be the expected height of column $c$ in $S'$. Since the grid point where $v$ was located in $S$ is

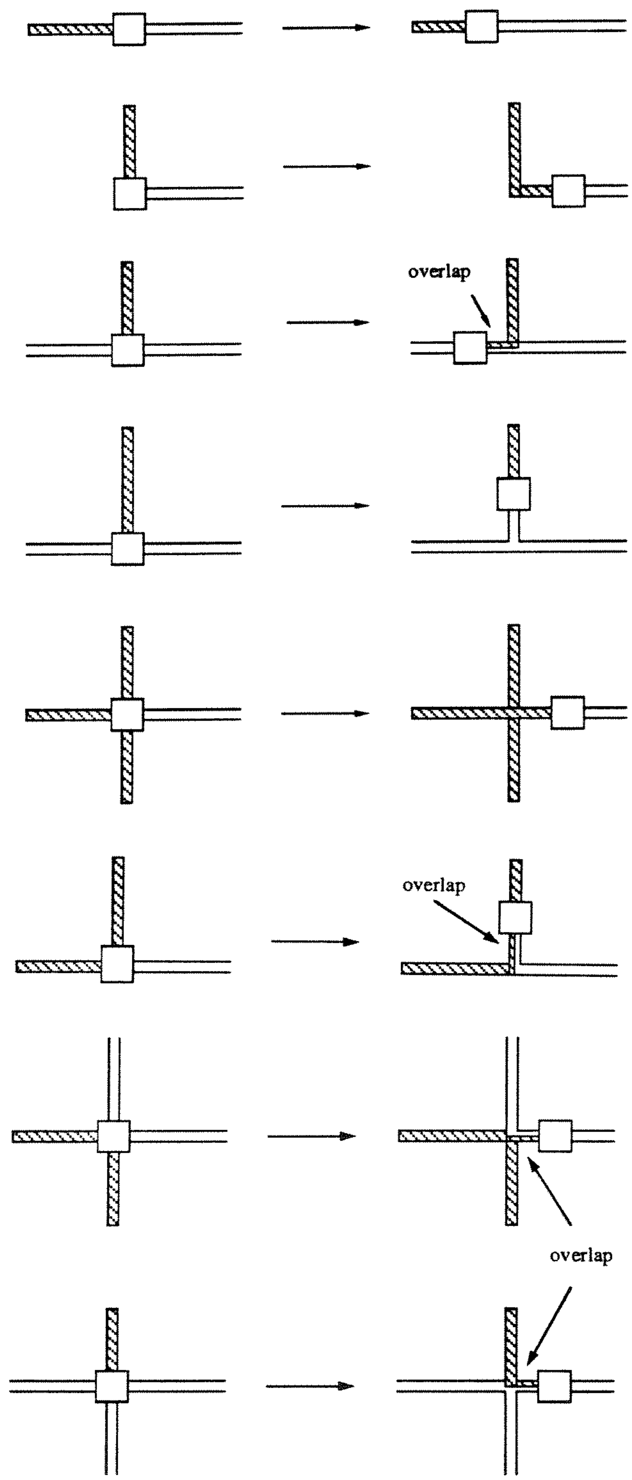



Figure 6: Different ways of shifting vias

now replaced by a horizontal wire in $S'$, we have

$$\begin{aligned} v_c' &= v_c - 1, \text{ and} \\ w_c' &= w_c + 1. \end{aligned}$$

Therefore,

$$\begin{aligned} f_c' &= v_c' + w_c' \\ &= v_c + w_c \\ &= f_c, \text{ and} \\ h_c' &= (\alpha+\gamma)v_c' + (\beta+\gamma)w_c' + \gamma \\ &= (\alpha+\gamma)(v_c - 1) + (\beta+\gamma)(w_c+1) + \gamma \\ &= ((\alpha+\gamma)v_c + (\beta+\gamma)w_c + \gamma) - (\alpha - \beta) \\ &= h_c - (\alpha-\beta) \\ &< h_c. \end{aligned}$$

That is, the expected height of column $c$ in $S'$ is decreased by $\alpha - \beta$ from the height of column $c$ in $S$. Similarly, if a via is shifted to column $c$ from another column, then the expected height of column $c$ is increased by $\alpha - \beta$.

2. *Vertical via shifting.*

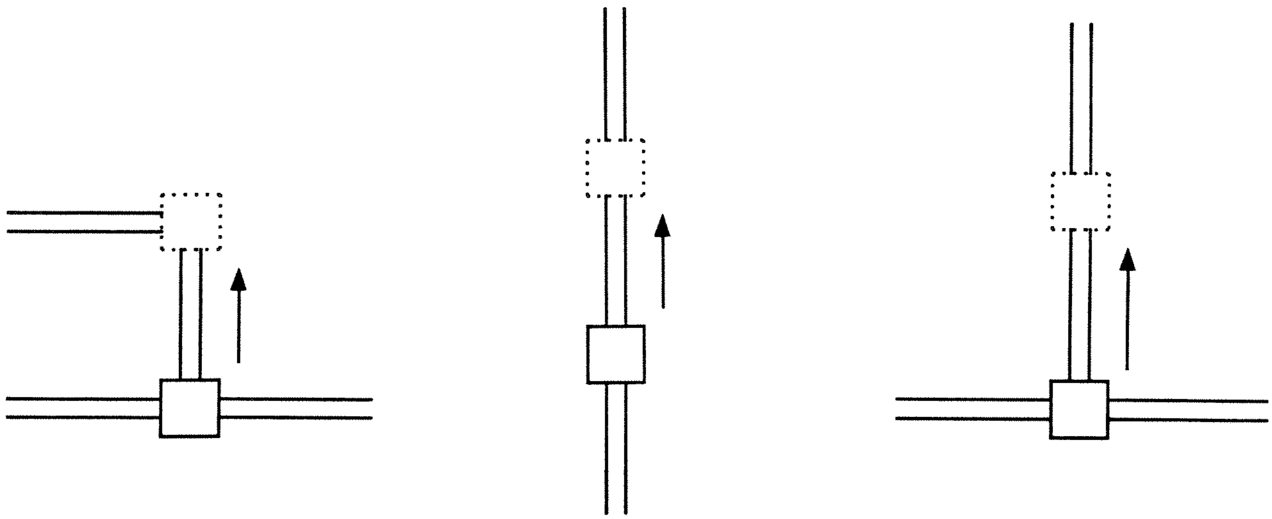
   Suppose the given grid-based two-layer channel routing solution $S$ has no mergeable vias. For each via $v$ in $S$, if possible, we can vertically shift $v$ to a "corner" (*i.e.*, an intersection of a vertical wire segment and a horizontal wire segment). Hence, without loss of generality, we may assume that the original channel routing solution $S$ has the property that if a via $v$ is not at a corner, then it is not possible to shift it to a corner. It follows that we only need to consider the following three cases of vertical shifting of vias:

   (a) From corner to corner (Figure 7(a));

   (b) From non-corner to non-corner (Figure 7(b));

   (c) From corner to non-corner (Figure 7(c)).

   If we shift a via $v$ along a column $c$ in a channel routing solution $S$ so that $v$ stays at column $c$ to obtain a new channel routing solution $S'$. Let $v_c'$, $w_c'$ and $f_c'$ be,



(a) Corner to corner (b) Non-corner to non-corner (c) Corner to non-corner

Figure 7: Vertical shifting of vias

respectively, the via count, wire count and feature count at column $c$ in $S'$, and let $h_c'$ be the expected height of column $c$ in $S'$. Then we have

$$v_c' = v_c,$$

$$w_c' = \begin{cases} w_c & \text{Cases (a) and (b)} \\ w_c + 1 & \text{Case (c),} \end{cases}$$

$$\begin{aligned} f_c' &= v_c' + w_c' \\ &= \begin{cases} f_c & \text{Cases (a) and (b)} \\ f_c + 1 & \text{Case (c),} \end{cases} \end{aligned}$$

$$\begin{aligned} h_c' &= (\alpha+\gamma)v_c' + (\beta+\gamma)w_c' + \gamma \\ &= \begin{cases} h_c & \text{Cases (a) and (b)} \\ h_c + (\beta+\gamma) & \text{Case (c)} \end{cases} \\ &\geq h_c. \end{aligned}$$

From the last formula we conclude that shifting vias vertically cannot decrease the height of a column. Thus only horizontal shifting of vias can decrease the expected height of a column, and hence the expected height of a channel. Therefore, in order to minimize the expected height of a channel, we only need to consider horizontal shifting of vias. There is no point of shifting a via vertically unless the via is eventually shifted horizontally out of the column it was originally in, in that case, the effect is the same as a horizontal shifting. In summary, we have

**Theorem 3.1** *Given a grid-based two-layer channel routing solution $S$ without mergeable vias such that each via of $S$ is located at a corner of $S$ if it is possible. Then any channel routing solution derivable from $S$ by shifting vias vertically has expected height $\geq H_S$, the expected height of $S$.*

According to Theorem 3.1, instead of consider the rectilinear graph associated with each via in a channel routing solution, we can consider an interval of consecutive columns between the leftmost and rightmost column intersecting the rectilinear graph associated with the via. Such an interval is called the *interval* of the via. Each via in a channel routing solution can be shifted to a position of any column in its interval without violating design rules. In doing so, we may have to shift the via from one track to another, or introduces overlapping of wires of the same net. For the example in Figure 1, the interval of vias 1 and 2 is $I_1 = I_2 = [1, 3]$, and the interval of vias 3 and 4 is $I_3 = I_4 = [10, 12]$.

Note that the rectilinear graph associated with a via in a channel routing solution $S$ is the same as the rectilinear graph associated with the same via in any channel routing solution derivable from $S$, assuming there is no mergeable vias. Therefore, the interval of a via is the same in any channel routing solution derivable from the same channel routing solution.

# 4 Preliminaries

We introduce in this section some necessary definitions and preliminary results. A *network* is a 5-tuple $G_\psi = (N, E, \psi, s, t)$, such that $G = (N, E)$ is a directed graph with vertex set $N$ and arc set $E$, and $s$, $t$ are distinguished vertices of $G$, called the *source* and *sink* of $G_\psi$, respectively, such that $s$ has no incoming arcs and $t$ has no outgoing arcs. The non-negative function $\psi$ defined on $E$ is called the *capacity function* of $G_\psi$. For each arc $a \in E$, $\psi(a)$ is called the *capacity* of the arc $a$. A *flow* $\phi$ of $G_\psi$ is a non-negative function defined on $E$, such that for each $a \in E$, $\phi(a) \leq \psi(a)$, and such that for each $v \in N - \{s, t\}$,

$$\sum_{a=(u,v)\in E} \phi(a) = \sum_{b=(v,w)\in E} \phi(b).$$

The value

$$\sum_{a=(s,v)\in E} \phi(a) = \sum_{b=(v,t)\in E} \phi(b)$$

is called the *value* of the flow $\phi$. A *maximum flow* of $G_\psi$ is a flow of $G_\psi$ with maximum value. It is shown in [17, 18] that the maximum flow of a network with integer capacity function can be computed in $O(mn \log n)$ time, where $m$, $n$ are the number of arcs and vertices of the network, respectively.

Given a grid-based two-layer channel routing solution $S$, we can assume without loss of generality that

- no two vias are mergeable in $S$, and
- each via is located at some corner of $S$ if possible.

Note that these conditions can be guaranteed by a preprocessing procedure. We can use depth-first search [1] to compute the rectilinear graph associated with each via, and merge vias that share a common rectilinear graph. For the remaining vias, if there is a corner in its associated rectilinear graph, then move the via to any one of such a corner by appropriately changing the layer assignment along the path that the via travels, otherwise the via cannot be moved to a corner. This preprocessing procedure can be done in $O(WL)$ time, where $W$ is the number of tracks in $S$, and $L$ is the length (*i.e.*, the number of columns) of $S$. As observed earlier, it suffices to consider the horizontal positions of the vias. Consequently, we only need to consider for each via $j$ an interval of columns $I_j = [l_j, r_j]$, such that via $j$ can and only can be moved to positions in columns $k$, $l_j \leq k \leq r_j$. We will refer to $l_j$ ($r_j$, respectively) as the *left* (*right*, respectively) *endpoint of via* $j$.

A via is said to be *active* if it intersects some horizontal wire. Imagine that we remove all the active vias from $S$. Then the via count at column $c$ becomes $d_c$, the number of non-active vias at column $c$, while the wire count at column $c$ is increased by $v_c - d_c$ because each active via at column $c$ is now replaced by a horizontal wire. Hence the height of column $c$ now becomes

$$\begin{aligned} p_c &= \alpha d_c + \beta((v_c - d_c) + w_c) + \gamma(v_c + w_c + 1) \\ &= (\alpha - \beta)d_c + (\beta + \gamma)(v_c + w_c) + \gamma \\ &= (\alpha - \beta)d_c + (\beta + \gamma)f_c + \gamma. \end{aligned}$$

Whenever a via is put back to column $c$, its height is increased by $\alpha - \beta$ because a horizontal wire is replaced by a via. Therefore, if $x_c$ vias are put back to column $c$ in $S'$, $0 \leq x_c \leq f_c - d_c$, then the height of column $c$ in $S'$ is $h'_c = p_c + (\alpha - \beta)x_c$, and the expected height of $S'$ is $H_{S'} = \max\{h'_c : 1 \leq c \leq L\}$. It should be observed that $p_c \leq h'_c$ for any solution $S'$ derivable from $S$. Therefore, we have

**Lemma 4.1** *Let $S$ be a grid-based two-layer channel routing solution with length $L$, then $l = \max\{p_c : 1 \leq c \leq L\}$ is a lower bound of the expected height of any channel routing solution derivable from $S$.*

Furthermore, we have

**Theorem 4.2** *Let $S$ be a grid-based two-layer channel routing solution with length $L$, then for any channel routing solution $S'$ derivable from $S$ to have expected height $H_{S'} \leq H$, if and only if at most*

$$z_S(H, c) = \lfloor \frac{H - p_c}{\alpha - \beta} \rfloor$$

*active vias are on column $c$ in $S'$, $1 \leq c \leq L$.*

The next lemma is important in guaranteeing the termination of our algorithm to be presented in Section 6.

**Lemma 4.3** *Let $S'$, $S''$ be two grid-based two-layer channel routing solutions derivable from the same channel routing solution, then there exists $\delta > 0$, such that either $H_{S'} = H_{S''}$ or $|H_{S'} - H_{S''}| > \delta$, where $H_{S'}$, $H_{S''}$ are, respectively, the expected height of $S'$ and $S''$.*

**Proof:** Since both $H_{S'}$ and $H_{S''}$ are of the form $i(\alpha + \gamma) + j(\beta + \gamma) + \gamma$, $1 \leq i, j \leq W$, where $W$ is the number of tracks in $S'$ $(S)$, it can be easily verified that

$$\delta = \min\{|i(\alpha + \gamma) + j(\beta + \gamma)| \neq 0 : -W \leq i, j \leq W\},$$

satisfies the properties listed in the lemma. □

# 5 A Boolean Procedure

We describe in this section a boolean procedure Feasible($S, H$) which given a grid-based two-layer channel routing solution $S$ and a target expected height $H$, determines whether there is a channel routing solution derivable from $S$ with expected height $\leq H$. Section 5.1 describes an implementation of Feasible($S, H$) based on a network flow formulation of the problem. Section 5.2 presents a faster implementation of Feasible($S, H$) based on a greedy method.

## 5.1 A Network Flow Formulation

Given a grid-based two-layer channel routing solution $S$ with $W$ tracks and $L$ columns, and a target expected height $H$, we construct a network $G_{\psi_{S,H}} = (N, E, s, t, \psi_{S,H})$, such that

$$
\begin{aligned}
N &= \{s,t\} \cup \{v : \; v \text{ is a via of } S\} \cup \{c : \; c \text{ is a column of } S\} \\
E &= \{(s,v) : \; v \text{ is a via of } S\} \cup \{(c,t) : \; c \text{ is a column of } S\} \cup \\
&\quad\ \{(v,c) : \; v \text{ is a via, } c \text{ is a column of } S \text{ and } c \text{ is in the interval of } v\} \\
\psi_{S,H}((s,v)) &= 1 \qquad \text{for all vias } v \text{ of } S \\
\psi_{S,H}((v,c)) &= 1 \qquad \text{for all vias } v \text{ and all columns } c \text{ of } S \text{ with } (v,c) \in E \\
\psi_{S,H}((c,t)) &= \lfloor \frac{H - p_c}{\alpha - \beta} \rfloor \\
&= z_S(H,c) \qquad \text{for all columns } c \text{ of } S.
\end{aligned}
$$

Figure 8(a) shows a channel routing solution $S$. Figure 8(b) shows its corresponding network $G_{\psi_S}$ (the capacity function is not shown in the figure). We now have

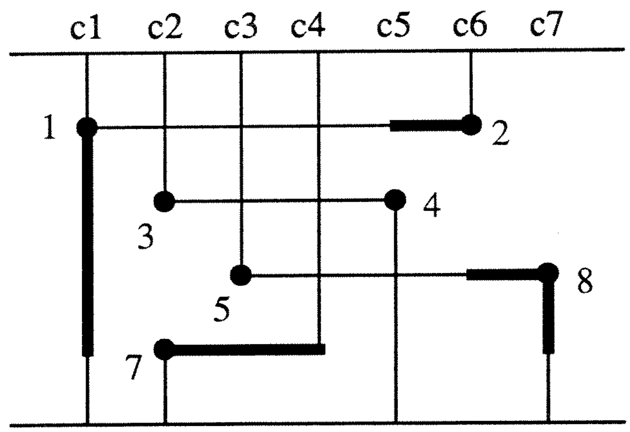
**Theorem 5.1** *Let $G_{\psi_{S,H}} = (N, E, s, t, \psi_{S,H})$ be the network corresponding to a grid-based two-layer channel routing solution $S$ as constructed above, let $V$ be the total number of vias of $S$, then $G_{\psi_{S,H}}$ has a flow of value $V$ if and only if there exists a channel routing solution $S'$ derivable from $S$ such that $H_{S'} \leq H$.*

**Proof:** Suppose there exists a channel routing solution $S'$ derivable from $S$ with $H_{S'} \leq H$. We can construct a flow $\phi$ of $G_{\psi_{S,H}}$ as follows:

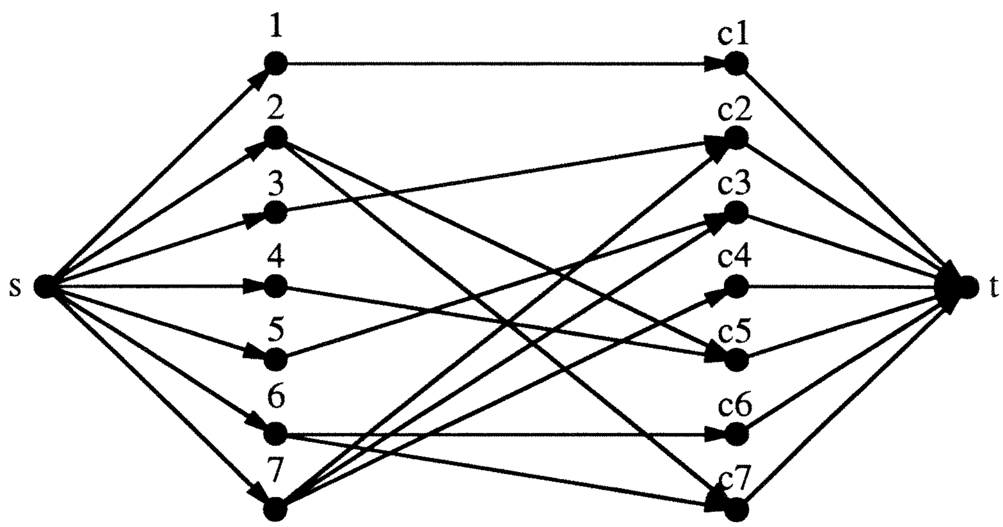
$$
\begin{aligned}
\phi((s,v)) &= 1 \\
\phi((v,c)) &= \begin{cases} 1 & \text{if via } v \text{ is on column } c \text{ in } S' \\ 0 & \text{otherwise} \end{cases} \\
\phi((c,t)) &= \sum_{(v,c)\in E} \phi((v,c))
\end{aligned}
$$

for all vias $v$ and columns $c$ of $S$ ($S'$). According to the definition of $\phi$, $\sum_{(v,c)\in E} \phi((v,c))$ is the number of active vias on column $c$ in $S'$. Because $H_{S'} \leq H$, by Theorem 4.2, we have

$$
\begin{aligned}
\phi((c,t)) &= \sum_{(v,c)\in E} \phi((v,c)) \\
&\leq z_S(H,c) \\
&= \psi_{S,H}((c,t)).
\end{aligned}
$$




(a) A channel routing solution




(b) The corresponding network

Figure 8: A channel routing solution and its corresponding network

Hence for any $a \in E$, we have $\phi(a) \leq \psi_{S,H}(a)$. Since each via is placed on exactly one column in $S'$, for each via $v$ in $S$ ($S'$), we have

$$\sum_{(u,v)\in E} \phi((u,v)) = \phi((s,v)) = \sum_{(v,c)\in E} \phi((v,c)) = 1.$$

Also, by the definition of $\phi$, we have

$$\sum_{(v,c)\in E} \phi((v,c)) = \phi((c,t)) = \sum_{(c,w)\in E} \phi((c,w)).$$

Therefore, $\phi$ is a flow of $G_{\psi_{S,H}}$, and its value is

$$\sum_{(s,v)\in E} \phi((s,v)) = V.$$

On the other hand, assume $G_{\psi_{S,H}}$ has a flow of value $V$, then it also has a integer-valued flow $\phi$ of value $V$ because all of its arcs have integral capacities. Hence we must have $\phi((s,v)) = 1$ for all vias $v$ of $S$. Therefore, $\phi((v,c)) = 1$ for exactly one column $c$ for each via $v$, and $\phi((v,c)) = 0$ for the other $(v,c)$ pairs. We construct a channel routing solution $S'$ derivable from $S$ by placing via $v$ on column $c$ if and only if $\phi((v,c)) = 1$, then according to the definition of $G_{\psi_{S,H}}$, all vias are properly assigned in $S'$. Since

$$\psi_{S,H}((c,t)) = z_S(H,c)$$

for each column $c$, we have

$$\sum_{(v,c)\in E} \phi((v,c)) \leq z_S(H,c),$$

*i.e.*, the number of active vias assigned to column $c$ in $S'$ is at most $z_S(H,c)$. Hence according to Theorem 4.2, the expected height of $S'$ is $\leq H$. □

According to Theorem 5.1, Feasible$(S,H)$ = **true** if and only if $G_{\psi_{S,H}}$ has a flow of value $V$. Note that the capacity function $\psi_{S,H}$ of $G_{\psi_{S,H}}$ is integer-valued and hence $G_{\psi_{S,H}}$ has an integer-valued maximum flow, and it can be computed in $O(|N||E|\log|N|) = O(WL(V+L)\log(V+L))$ time because $O(|E|) = O(WL)$, assuming $S$ has no mergeable vias (hence no two vias have a common column in their interval, and therefore the sum of the indegrees of the columns in $G_{\psi_{S,H}}$ is at most $O(WL)$). The network $G_{\psi_{S,H}}$ can be constructed in $O(WL)$ time. Therefore, the network flow implementation of Feasible$(S,H)$ runs in $O(WL(V+L)\log(V+L))$ time.

## 5.2 A Faster Approach

We now describe a faster implementation of Feasible$(S, H)$. For $1 \leq c \leq L$, let $Q(c)$ denote the set of active vias $j$ with left endpoint $l_j = c$. In procedure *Feasible*$(S, H)$, each column $c$ is considered in order from left to right. Among the set of yet to be assigned vias whose interval contains column $c$, the $z_S(H, c)$ vias with smallest right endpoints are assigned to column $c$ (if there are that many such vias). The procedure returns **true** if and only if all vias are assigned to some column in this way. It is described more formally as follows.

```
Procedure Feasible (S, H) : boolean;
  (* S is a grid-based two-layer channel routing solution with length L *)
  (* H > 0 is the target expected height *)
  Begin
    for c := 1 to L do
      cap(c) := z_S(H, c);
      (* cap(c) is the number of vias column c can still accommodate *)
    cap(L + 1) := +∞;          (* an auxiliary variable *)
    c := 1;
    failed := false;
    Ready := Q(1);          (* the set of vias ready to be assigned *)
    while (c ≤ L + 1) and not (failed) do
      if (cap(c) = 0) or (Ready = φ)
        then begin
               c := c + 1;
               Ready := Ready ∪ Q(c)
             end
        else begin
               Let via j be the via in Ready with smallest right endpoint r_j;
               if c > r_j
                 then failed := true
                 else begin
                        Remove via j from Ready;
                        Assign via j to column c;
                        cap(c) := cap(c) - 1
                      end
             end;
    if failed
      then Feasible := false
      else Feasible := true
  End.
```

A column $c$ of $S$ is said to be *saturated* if exactly $z_S(H, c)$ vias are assigned to it by procedure Feasible$(S, H)$, otherwise it is *unsaturated*, *i.e.*, fewer then $z_S(H, c)$ vias are assigned to it by procedure Feasible($S, H$). An interval $[i, j]$ is said to be *invaded from the left* (*right*, respectively) *by a via* $k$ *of* $S$ if via $k$ has left endpoint $l_k < i$ (right endpoint $r_k > j$, respectively), and it is assigned to some column $c$, $i \leq c \leq j$, by procedure Feasible($S, H$). The correctness of procedure Feasible($S, H$) is now stated in the following theorem.

**Theorem 5.2** *There exists a channel routing solution* $S'$ *derivable from* $S$ *with expected height* $H_{S'} \leq H$ *if and only if Feasible*($S, H$) = **true**.

**Proof:** If Feasible($S, H$) = **true**, then each active via is properly assigned in $S'$ and at most $z_S(H, c)$ active vias are assigned to column $c$, $1 \leq c \leq L$. Hence $S'$ so obtained is a channel routing solution derivable from $S$ with expected height $\leq H$. Assume Feasible($S, H$) = **false**, then there exists an active via $j$ with left endpoint $l_j$ and right endpoint $r_j$ that is not assigned to any column between $l_j$ and $r_j$ (inclusive) by procedure Feasible($S, H$). Let via $j$ be the first of such vias and let

$$l_0 = \max\{l \leq l_j : \text{the interval } [l, r_j] \text{ is not invaded from the left}\},$$

then $l_0$ is well defined because the interval $[1, r_j]$ is not invaded from the left and $1 \leq l_j$. We claim that

- for $l_0 \leq c \leq r_j$, column $c$ is saturated, and
- the interval $[l_0, r_j]$ is not invaded from the right either.

Assume column $c$ is unsaturated for some $l_0 \leq c \leq r_j$, then we must have $c < l_j$, for otherwise via $j$ would have been assigned to some column between $l_j$ and $r_j$ (inclusive) by procedure Feasible($S, H$). Hence $l_0 < c + 1 \leq l_j$ and the interval $[c + 1, r_j]$ is not invaded from the left (for according to procedure Feasible($S, H$), column $c$ would have been saturated before any via is allowed to invade the interval $[c + 1, r_j]$), contradicting the choice of $l_0$. Suppose the interval $[l_0, r_j]$ is invaded from the right and let column $c$, $l_0 \leq c \leq r_j$ be the rightmost column such that an active via $k$ with right endpoint $r_k > r_j$ is assigned to it by procedure Feasible($S, H$). For similar reasons, we must have $c < l_j$, $l_0 < c + 1 \leq l_j$ and the interval $[c + 1, r_j]$ is not invaded from the left, a contradiction with the choice of $l_0$. Therefore, the number of vias of $S$ with left endpoint $\geq l_0$ and right endpoint $\leq r_j$ is at least

$$\sum_{c=l_0}^{r_j} z_S(H, c) + 1,$$

and hence in any channel routing solution $S'$ derivable from $S$, there exists a column $c$, $l_0 \leq c \leq r_j$, such that at least $z_S(H, c) + 1$ active vias are assigned to column $c$. According to Theorem 4.2, the expected height of $S'$ is $> H$. □

**Theorem 5.3** *Given the values of the $p_c$'s, Procedure Feasible$(S, H)$ can be implemented to run in $O(L + V \log V)$ time, where $W$, $L$, and $V$ are respectively, the number of tracks, columns and vias in $S$.*

**Proof:** Given the values of the $p_c$'s, then everything outside of the while loop in Procedure Feasible$(S, H)$ can be done in $O(V + L)$ time (*Ready* can be sorted in $O(V)$ time using bucket sort [1]). The while loop is executed at most $O(L)$ time. Each iteration of the while loop takes constant time except for the updating of the set *Ready*. The set *Ready* can be maintained as a heap, and new elements are inserted one at a time in $O(\log |Ready|)$ time each insertion. Since there are a total number of $O(V)$ insertions into *Ready*, the total amount of time spent on updating *Ready* is $O(V \log V)$ because $|Ready| \leq V$. Hence the while loop can be done in $O(L + V \log V)$ time. Therefore the theorem follows. □

Note that the values of the $p_c$'s can be computed in $O(WL)$ time by scanning $S$. Since the vias can be numbered consecutively from 1 to $V$, we can use the results in [10, 11, 12] to reduce the time complexity of Procedure Feasible$(S, H)$ to $O(L + V \log \log V)$. Using the efficient UINON-FIND algorithm in [19], we can further reduce the running time of Procedure Feasible$(S, H)$ to $O(L + V\Gamma(V))$, where $\Gamma(V)$ is a extremely slowly growing function related to the functional inverse of the Ackermann's function. The techniques used are similar to those in [15]. Observe that by using more sophisticated data structures, Procedure Feasible$(S, H)$ can also be implemented to run in $O(L + V \log W)$ time.

# 6 An Optimal Via Shifting Algorithm

Based on the materials presented in the last section, we are now ready to present our main algorithm. The algorithm consists of a binary search loop. In each iteration of the loop, a procedure Feasible$(S, H)$ is used to determine whether there exists a channel routing solution derivable from the given solution $S$ with expected height $\leq H$. The search terminates when the length of the search interval is $\leq \delta$, because according to Lemma 4.3, there is no channel routing solution derivable from $S$ has expected height lies in the search interval.

```
Algorithm Via_Shifting (S);
  (* S is a grid-based two-layer channel routing solution with W tracks and L columns *)
  Begin
    (* preprocessings *)
    Merge vias;
    Shift vias to corners if possible;
    for c := 1 to L do
      p_c := (α − β)d_c + (β + γ)f_c + γ;
    l := max{p_c : 1 ≤ c ≤ L};
    u := H_S;
    δ := min{|(α + β)i + (β + γ)j| ≠ 0 : −W ≤ i, j ≤ W};
    for j := 1 to V do
      Compute the interval I_j = [l_j, r_j] of via j;
    (* the main loop *)
    while (l ≤ u) do
      begin
        H := (l + u)/2;
        if Feasible(S, H)
          then u := H − δ
          else l := H + δ
      end;
    (* postprocessings *)
    if not(Feasible(S, H))
      then Feasible(S, u + δ);
    Assign active vias according to the last call of procedure Feasible
  End.
```

**Theorem 6.1** *Algorithm Via_Shifting correctly computes a channel routing solution derivable from a given grid-based two-layer channel routing solution $S$ with minimum expected height in $O(W(W+L) + V\log^2 W)$ time, where $W$, $L$, and $V$ are, respectively, the number of tracks, columns and vias in $S$.*

**Proof:** It is clear that Algorithm Via_Shifting returns a channel routing solution derivable from $S$. It can be seen that at the beginning of each iteration of the main loop, there is a channel routing solution derivable from $S$ with expected height $\leq u + \delta$, and no channel routing solution derivable from $S$ with expected height $\leq l - \delta$, *i.e.*, Feasible$(S, u+\delta) =$ **true** and Feasible$(S, l-\delta) =$ **false**. When the main loop terminates, we have $0 < l - u \leq \delta$ and $H = (l+u)/2$. Hence

$$H - (l - \delta) = (u + \delta) - H = \delta - \frac{l-u}{2} < \delta.$$

Therefore, if Feasible($S, H$) = **true**, and let $S'$ be the channel routing solution derivable from $S$ obtained by assigning the active vias according to Feasible($S, H$), then $H_{S'}$ is the minimum expected height of any channel routing solution derivable from $S$. For otherwise let $h < H_{S'}$ be the minimum expected height of any channel routing solution derivable from $S$, then $H_{S'} - h > \delta$ according to Lemma 4.3. Hence we have $h < H_{S'} - \delta \leq H - \delta < l - \delta$, contradicting the fact that there is no channel routing solution derivable from $S$ with expected height $\leq l - \delta$. On the other hand, if Feasible($S, H$) = **false**, then Feasible($S, u + \delta$) = **true**. And we can similarly prove that the channel routing solution returned by the algorithm has minimum expected among all channel routing solutions derivable from $S$. This establishes the correctness of the algorithm. The first two steps of the algorithm can be done in $O(WL)$ time, $\delta$ can be computed in $O(W^2)$ time and $l$ and $u$ can be computed in $O(L)$ time. The two for loops each requires $O(WL + V)$ time. The main loop is executed

$$\begin{aligned} \log \frac{u-l}{\delta} &= \log \frac{H_S - l}{\delta} \\ &\leq \log \frac{\alpha - \beta}{\delta} W \\ &= O(\log W) \end{aligned}$$

times, each iteration requiring $O(L + V \log W)$ time. The last two steps can be done in $O(L + V \log W)$ time. Hence the overall complexity of the algorithm is:

$$O(W^2 + WL + (L + V \log W) \log W) = O(W(W + L) + V \log^2 W). \square$$

The time complexity statement in Theorem 6.1 is based on the second implementation of Feasible($S, H$) as described in Section 5.2. It is easy to see that if we use the network flow implementation of Feasible($S, H$), then algorithm Via_Shifting runs in $O(W^2 + WL(V + L) \log W \log(V + L))$ time. Even though the algorithm based on the network flow implementation of Feasible($S, H$) is slower, it is a more robust algorithm in the sense that it can handle additional via placement constraints. In fact, if for some reasons we disallow a via to be shifted to a certain subset of columns within its interval, all we need to do is to delete the corresponding edges in the network, while the algorithm based on the greedy method cannot be adapted to solve the problem optimally in this case. Thus we have two implementations of the algorithm Via_Shifting, the advantage of the first one is robustness and the advantage of the second one is speed.

## 7 Conclusions

Many techniques have been proposed to modify channel routing solutions in order to obtain more compactable results. The problem of using these techniques to minimize channel height after compaction is known to be NP-hard [6, 7]. By focusing on a single technique, *i.e.*, via shifting, and by reasonably choosing a measure of channel height after compaction, we are able to develop two efficient polynomial time algorithms to optimally solve the problem. One algorithm is based on a network flow formulation of the problem. This algorithm is very robust and can handle additional via placement constraints. Another algorithm is based on a technique similar to the one used in [15] for computing a maximum matching of a convex bipartite graph. This algorithm is less robust by runs faster.

It is well known that the expected height of the critical column(s) of a channel routing solution is the most significant factor that determines the channel height after compaction. There is another factor called *bump propagation* [8] that can also affect the channel height after compaction. We have not included the effects of bump propagation into our measure of channel height after compaction. Further research is needed to design efficient algorithms that can minimize a new channel height measure which consider both the height of the critical column(s) and the effects of bump propagation.

**Acknowledgements.** The authors wish to thank Professor J. Cong of UCLA for very helpful discussions.

## References

[1] A.V. Aho, J.E. Hopcroft and J.D. Ullman, *The design and analysis of computer algorithms*, Addison Wesley, Reading, MA, 1974.

[2] M. Burstein and R. Pelvin, "Hierarchical channel router", *INTEGRATION, the VLSI journal, vol.* **1**, 21-38, 1983.

[3] Y. Cai and D.F. Wong, "Optimal via-shifting in channel compaction", *Proc. of the First European Design Automation Conference*, 186-190, 1990.

[4] Y. Cai and D.F. Wong, "A fast algorithm for optimal via-shifting in channel compaction", *Proc. of the 1991 IEEE International Symposium on Circuits and Systems*, 1940-1943.

[5] C.K. Cheng and D.N. Deutsch, "Improved channel routing by via minimization and shifting", *Proc. of the 25th ACM/IEEE Design Automation Conference*, 677-680, 1988.

[6] J. Cong and D.F. Wong, "How to obtain more compactable channel routing solutions", *Proc. of the 25th ACM/IEEE Design Automation Conference*, 663-666, 1988.

[7] J. Cong and D.F. Wong, "Generating more compactable channel routing solutions", *INTEGRATION, the VLSI journal, vol.* **9**, 199-214, 1990.

[8] D.N. Deutsch, "Compacted channel routing", *Proc. of the 1985 IEEE International Conference on Computer-Aided Design*, 223-225.

[9] D.N. Deutsch, "A dogleg channel router", *Proc. of the 13rd ACM/IEEE Design Automation Conference*, 425-433, 1976.

[10] P. van Emde Boas, "Preserving order in a forest in less than logarithmic time", *Proc. of the 16th IEEE Annual Symposium on Foundations of Computer Science*, 75-84, 1975.

[11] P. van Emde Boas, "Preserving order in a forest in less than logarithmic time and linear space", *Info. Proc. Letts., vol.* **6**, 80-82, 1977.

[12] P. van Emde Boas, P. Kaas, and E. Zijlstra, "Design and implementation of an efficient priority queue", *Math. Systems Theory, vol.* **10**, 99-127, 1977.

[13] C.M. Fiduccia and R.L. Rivest, "A greedy channel router", *Proc. of the 19th ACM/IEEE Design Automation Conference*, 418-424, 1982.

[14] A. Hashimoto and J. Stevens, "Wire routing by optimizing channel assignment within large apertures", *Proc. of the Eighth ACM/IEEE Design Automation Workshop*, 155-163, 1971.

[15] W. Lipski and F.P. Preparata, "Efficient algorithms for finding maximum matchings in convex bipartite graphs and related problems ", *Acta Informatica, vol.* **15**, 329-346, 1981.

[16] J. Reed, A. Sangiovanni-Vincentelli and M. Santomauro, "A new symbolic channel router: YACR2", *IEEE Trans. on CAD, CAD-***4**, 208-219, 1985.

[17] D.D. Sleator, "An $O(mn \log n)$ algorithm for maximum network flow", *Technical Report #STAN-CS-80-831*, Computer Science Department, Stanford University, 1980.

[18] D.D. Sleator and R.E. Tarjan, "A data structure for dynamic trees", *Journal of Computer and System Sciences, vol.* **26**, 362-391, 1983.

[19] R.E. Tarjan, "Efficiency of a good but not linear set union algorithm", *Journal of the Association for Computing Machinery, vol.* **22**, 215-224, 1975.

[20] K.-S. The, D.F. Wong and J. Cong, "Via minimization by layout modification", *Proc. of the 26th ACM/IEEE Design Automation Conference*, 799-802, 1989.

[21] K.-S. The, D.F. Wong and J. Cong, "A layout modification approach to via minimization", *IEEE Trans. on CAD, CAD-***10**, 536-541, 1991.

[22] D.F. Wong and C.L. Liu, "Compacted channel routing with via placement restrictions", *INTEGRATION, the VLSI journal, vol.* **4**, 287-307, 1986.

[23] X. Xiong and E.S. Kuh, "Netcracker: an efficient and intelligent channel router", *Proc. of the 24th ACM/IEEE Design Automation Conference*, 298-304, 1987.

[24] T. Yoshimura and E.S. Kuh, "Efficient algorithms for channel routing", *IEEE Trans. on CAD, CAD-***1**, 25-35, 1982.