Optimal Post-Routing Redundant Via Insertion*

Kuang-Yao Lee†, Cheng-Kok Koh‡, Ting-Chi Wang†, and Kai-Yuan Chao♯
Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan†
School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana‡
Intel Corporation, Hillsboro, Oregon♯
d924347@oz.nthu.edu.tw, chengkok@purdue.edu,
tcwang@cs.nthu.edu.tw, kaiyuan.chao@intel.com

ABSTRACT

Redundant via insertion is highly recommended for improving chip yield and reliability. In this paper, we study the problem of *double-cut via insertion* (DVI) in a post-routing stage, where a single via can have at most one redundant via inserted next to it and the goal is to insert as many redundant vias as possible. The DVI problem can be naturally formulated as a zero-one integer linear program (0-1 ILP). Our main contributions are acceleration methods for reducing the problem size and the number of constraints. Moreover, we extend the 0-1 ILP formulation to handle via density constraints. Experimental results show that our 0-1 ILP is very efficient in computing optimal DVI solution, with up to 35.3 times speedup over existing heuristic algorithms.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids; J.6 [**Computer Applications**]: Computer-Aided Engineering—*Computer-aided design*

General Terms

Algorithms, Design

Keywords

redundant via insertion, via density, integer linear program

1. INTRODUCTION

The scaling of manufacturing technology has made integrated circuits (ICs) more sensitive to process variations. As a consequence, the yield of ICs has suffered [1]. In particular, via-open defect has been identified as one of the main factors to cause chip failures [2]. Via failures may be caused by various reasons, such as electromigration, cut misalignment, and thermal stress induced voiding effects [3, 4, 5]. Partially failed vias add unexpected delay and completely failed vias leave open nets in a circuit. Therefore, it is crucial to reduce the yield loss due to via failures. Since a redundant via provides an alternative current path, redundant via insertion has become one of the well known and highly recommended methods to improve via yield/reliability [2, 6, 7, 8, 9]. Redundant vias can provide 6% increase in yield over the wafers with single-cut vias, a significant improvement considering that a 1% yield loss in a 300 mm wafer fab costs a chipmaker about five million dollars on an annual basis [9].

In general, the redundant via insertion problem can be tackled during routing or in a post-routing stage. In [10, 11, 12, 13], redundant vias are inserted in a post-routing stage. The authors of [10] formulated the problem as a maximum independent set (MIS) problem, which considered all single vias simultaneously. To solve the large-size MIS problem, a divide-and-conquer heuristic was proposed to partition the whole problem into several smaller MIS problems that were solved one at a time. However, optimality was not guaranteed. In [11], the single vias of a design were considered individually for redundant via insertion. Therefore, only locally optimal solutions could be obtained. In [12], the redundant via insertion problem was formulated as a bipartite matching problem that could be solved optimally only when the design is grid-based and involves at most three routing layers. All these works ignored via density constraints, the violation of which could adversely affect the yield and reliability of a design. Such constraints were considered in [13], which solved the problem with a two-stage heuristic approach, but with no guarantee on optimality. Some previous works, such as [14, 15], considered the redundant via insertion problem in the routing stage. Nonetheless, for further improving the yield/reliability of vias and electrical characteristics, it may still be desirable to perform additional redundant via insertion after the routing stage.

In this work, we study the problem of *double-cut via insertion* (DVI) in a post-routing stage, where a single via can have at most one redundant via inserted and the goal is to insert as many redundant vias as possible. The DVI problem can be naturally formulated as a zero-one integer linear program (0-1 ILP). Our main contributions are acceleration methods that exploit the properties of the DVI problem for reducing the problem size and the number of constraints without sacrificing the optimality. Moreover, we

*This work was partially supported by National Science Council of Taiwan under Grant No. NSC-96-2220-E-007-025 and NSC-96-2917-I-007-018.

ISPD'08, April 13–16, 2008, Portland, Oregon, USA.

extend the 0-1 ILP formulation to consider also via density constraints. Experimental results show that our 0-1 ILP approach can always generate optimal solutions, with runtimes that are up to 34.7 times faster than the approach in [10]. When via density constraints are considered, our approach can insert more redundant vias than an approach adapted from the two-stage approach in [13], with runtimes that are up to 35.3 times faster.

The rest of this paper is organized as follows. In Section 2, we give the definition of the double-cut via insertion problem. In Section 3, we present our 0-1 ILP formulation, and the accelerated methods for the DVI problem. In Section 4, we extend our 0-1 ILP formulation to consider via density constraints. We report the experimental results in Section 5, and conclude this paper in Section 6.

2. DOUBLE-CUT VIA INSERTION

Without loss of generality, we assume that a redundant via can be inserted next to a single via in one of four positions. Therefore, four types of double-cut vias can be defined (see Definition 1)[10].

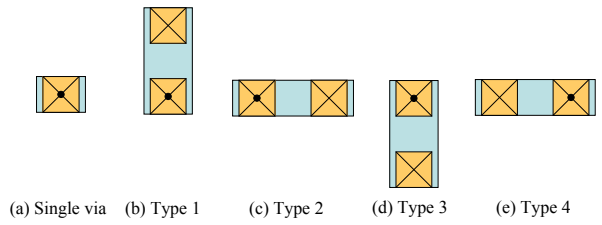


Figure 1: Double-cut via types [10].

Definition 1 (Double-cut via). *A single via together with a redundant via inserted next to it are defined as a double-cut via. According to the position of the redundant via, the double-cut via is one of the four types, as shown in Figure 1. Given a single via i, its double-cut via of type j ($j \in \{1, 2, 3, 4\}$) is denoted by dv(i,j). Besides, a double-cut via is said to be feasible if replacing the single via with the double-cut via does not violate any design rule and does not overlap with any critical area, assuming that other single vias are kept intact; otherwise the double-cut via is an infeasible one.*

The DVI problem is formally defined as follows:

Double-Cut Via Insertion (DVI): *Given a routed design and a user-specified set of single vias, the problem of double-cut via insertion is that of replacing as many single vias in the given via set as possible with double-cut vias, without violating any design rule and without re-routing any net.*

Note that redundant via insertion may change the timing behavior of a design [11]. Besides, inserting redundant vias in congested areas may also hurt the yield and reliability of a design [13]. We assume that designers have taken these into consideration when they specify the set of single vias to be considered for redundant via insertion.

3. 0-1 ILP FORMULATION FOR DOUBLE-CUT VIA INSERTION

In this section, we shall describe how to formulate the DVI problem as zero-one integer linear program (0-1 ILP). We also present methods to improve the efficiency of our 0-1 ILP formulation.

3.1 Preliminaries

For ease of presentation, we follow the notation and definitions used in [10] (see Table 1 and Definition 2).

Table 1: Notation

Notation	Description
$fdv_{i,j}$	Feasible double-cut via *dv(i,j)*
FDV	The set of all $fdv_{i,j}$'s
$fdv_{i,j} \leftrightarrow fdv_{i',j'}$	$fdv_{i,j}$ and $fdv_{i',j'}$ cannot be simultaneously inserted (i.e., they have a conflict relation)
sv_i	single via i that has at least one feasible double-cut via
SV	The set of all sv_i's
$v_{i,j}$	The vertex of a conflict graph corresponding to $fdv_{i,j}$
V_i	The set of vertices $v_{i,j}$'s of single via sv_i
$\#RVI$	The number of inserted redundant vias

Definition 2 (Conflict graph). *A conflict graph $G(V, E)$ is an undirected graph constructed from a detailed routing solution and a user-specified set of single vias. Each feasible double-cut via $fdv_{i,j}$ is associated with a vertex $v_{i,j}$ in V. An edge $(v_{i,j}, v_{i',j'}) \in E$ if and only if $fdv_{i,j} \leftrightarrow fdv_{i',j'}$.*

3.2 0-1 ILP Formulation

Before constructing the 0-1 ILP problem, we have to compute all sv_i's, all $fdv_{i,j}$'s, and the conflict relation between any pair of feasible double-cut vias. All information can be obtained by computing a conflict graph. In a conflict graph (see Definition 2), two vertices are connected by an edge if the corresponding feasible double-cut vias cannot be simultaneously inserted into the design due to some design rules (an external conflict relation) or they originate from the same single via (an internal conflict relation).

We use the algorithm *GCA*, proposed in [10], to construct the corresponding conflict graph $G(V, E)$ from a given design and the user-specified set of single vias. To formulate the problem as a 0-1 ILP, we associate each $v_{i,j} \in V$ with a binary variable $R_{i,j}$. If $R_{i,j} = 1$ in the 0-1 ILP solution, the corresponding $fdv_{i,j}$ is inserted into the design. Therefore, we should add the following *conflict constraints* into our 0-1 ILP formulation;

$$\text{Conflict constraint:} \quad R_{i,j} + R_{i',j'} \leq 1, \ \forall (v_{i,j}, v_{i',j'}) \in E, \quad (1)$$

which means that at most one of $fdv_{i,j}$ and $fdv_{i',j'}$ can be inserted into the design if $fdv_{i,j} \leftrightarrow fdv_{i',j'}$. Since the objective of the DVI problem is to maximize the number of inserted redundant vias (i.e., $\#RVI$ as defined in Table 1),

we can now formally define our 0-1 ILP formulation as follows:

$$\text{Maximize} \sum_{v_{i,j} \in V} R_{i,j}$$

$$\text{Subject to } R_{i,j} + R_{i',j'} \leq 1,\ \forall (v_{i,j}, v_{i',j'}) \in E;\ \text{and}$$
$$R_{i,j} \in \{0,1\},\ \forall v_{i,j} \in V. \quad (2)$$

This formulation has been recognized to finding an maximum independent set (MIS) of G as well (see for example [16]). However the time complexity to solve a 0-1 ILP problem is in general quite high. Therefore, we cannot afford to apply the 0-1 ILP approach to a large design directly. In the following, we will describe three acceleration methods that exploit the properties of the DVI problem to speed up the 0-1 ILP solver without sacrificing the optimality.

3.3 Speed-Up

3.3.1 Reduction in Constraints

Given a conflict graph $G(V, E)$, the edge set E can be divided into two disjoint subsets E_I=$\{(v_{i,j}, v_{i',j'}) | \forall (v_{i,j}, v_{i',j'}) \in E$ with $i = i'\}$ (internal conflict) and $E_X = \{(v_{i,j}, v_{i',j'}) \mid \forall (v_{i,j}, v_{i',j'}) \in E$ with $i \neq i'\}$ (external conflict). For each single via sv_i, there are at most six edges in E_I (and at most six constraints of the form in (1)). However, those internal conflict constraints can be simplified by a single *double-cut constraint* that limits the single via sv_i to have at most one redundant via inserted next to it:

$$\text{Double-cut constraint:} \sum_{fdv_{i,j}\text{'s of } sv_i} R_{i,j} \leq 1. \quad (3)$$

It should be noted that we do not have to construct the above double-cut constraint for a single via that has only one feasible redundant via. With the double-cut constraint (3), we only have to construct the conflict constraints (1) for the edges of E_X in the following modified 0-1 ILP formulation:

$$\text{Maximize} \sum_{v_{i,j} \in V} R_{i,j}$$

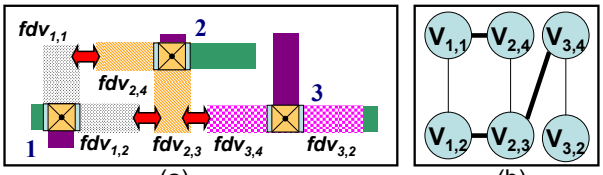
$$\text{Subject to} \sum_{fdv_{i,j}\text{'s of } sv_i} R_{i,j} \leq 1,\ \forall sv_i \in \text{SV};$$
$$R_{i,j} + R_{i',j'} \leq 1,\ \forall (v_{i,j}, v_{i',j'}) \in E_X;\ \text{and}$$
$$R_{i,j} \in \{0,1\},\ \forall v_{i,j} \in V. \quad (4)$$

It should be apparent that the following lemma holds.

Lemma 1. *For a given conflict graph corresponding to a DVI problem, the 0-1 ILP in (4) is equivalent to the 0-1 ILP in (2).*

From our experience, there are many more edges in E_I than in E_X. Therefore, the number of constraints of our 0-1 ILP formulation can be significantly reduced. The numbers of constraints and binary variables of the 0-1 ILP are $|E_X|$+|SV| and $|V|$ (i.e., |FDV|), respectively.

We show in Figure 2 an example of the new 0-1 ILP formulation. Given the routed design as shown in Figure 2(a), we assume that single vias 1, 2 and 3 are allowed to have one redundant via inserted. We further assume that the top redundant via of single via 1 and the left redundant via of single via 2 (the right redundant via of single via 1 and the bottom redundant via of single via 2, the bottom redundant via of single via 2 and the left redundant via of single via 3) cannot be inserted simultaneously due to design rules. The corresponding conflict graph is shown in Figure 2(b) with the edge subset E_X shown in bold. Because $fdv_{1,1} \leftrightarrow fdv_{2,4}$, the conflict constraint $R_{1,1} + R_{2,4} \leq 1$ is constructed. $R_{1,2} + R_{2,3} \leq 1$ and $R_{2,3} + R_{3,4} \leq 1$ are constructed similarly. For single vias 1, 2 and 3, the double-cut constraints $R_{1,1} + R_{1,2} \leq 1$, $R_{2,3} + R_{2,4} \leq 1$ and $R_{3,2} + R_{3,4} \leq 1$ are also constructed. The 0-1 ILP problem is shown in Figure 2(c).



Maximize:

$$R_{1,1} + R_{1,2} + R_{2,3} + R_{2,4} + R_{3,2} + R_{3,4}$$

Subject to

$$R_{1,1} + R_{2,4} \leq 1 \qquad R_{1,1} + R_{1,2} \leq 1$$
$$R_{1,2} + R_{2,3} \leq 1 \qquad R_{2,3} + R_{2,4} \leq 1$$
$$R_{2,3} + R_{3,4} \leq 1 \qquad R_{3,2} + R_{3,4} \leq 1$$
$$R_{1,1}, R_{1,2}, R_{2,3}, R_{2,4}, R_{3,2}, R_{3,4} \in \{0,1\}$$

(c)

Figure 2: An example of the DVI problem.

3.3.2 Pre-Selection

Given a conflict graph $G(V, E = E_I \cup E_X)$, for each single via sv_i, let V_i denote the set of vertices $v_{i,j}$'s of sv_i. We compute the vertex subset CV_i =$\{v_{i,j} \mid v_{i,j} \in V_i$ has no incident edges that belong to $E_X\}$. Suppose CV_i of sv_i is not empty. Since the vertices in CV_i have no conflict relation with the vertices originating from the other single vias, we can arbitrarily include any $v_{i,m} \in CV_i$ in an MIS of G. Besides, since a single via can have at most one redundant via inserted next to it, we can now delete all vertices $v_{i,j}$ originating from sv_i from the conflict graph. Let $RV_{PS}(G)$ denote the pre-selected vertices of G. Let the reduced conflict graph be denoted by $\tilde{G}(\tilde{V}$, $\tilde{E}$=$\tilde{E}_I \cup \tilde{E}_X)$, where $\tilde{V} = V - \bigcup_{sv_i \in \text{SV}} \{v_{i,j} | v_{i,j} \in V_i \wedge |\text{CV}_i| > 0\}$, $\tilde{E}_I = \{(v_{i,j}, v_{i',j'}) | v_{i,j}$ and $v_{i',j'} \in \tilde{V} \wedge (v_{i,j}, v_{i',j'}) \in E_I\}$ and $\tilde{E}_X = \{(v_{i,j}, v_{i',j'}) | v_{i,j}$ and $v_{i',j'} \in \tilde{V} \wedge (v_{i,j}, v_{i',j'}) \in E_X\}$. Since this method can decrease the numbers of variables and constraints of the 0-1 ILP problem, the efficiency of our approach can be improved. It can be shown that the pre-selection method does not hurt the optimality.

Lemma 2. *Let $RV_{PS}(G)$ and $\tilde{G}$ denote the pre-selected vertex set of conflict graph G and the reduced conflict graph, respectively. Besides, let $RV_{01}(G)$ denote the maximum independent set of G (or a 0-1 ILP solution obtained from (4)). Then, $RV_{PS}(G) \cup RV_{01}(\tilde{G})$ is an independent set, and $|RV_{PS}(G)| + |RV_{01}(\tilde{G})| = |RV_{01}(G)|$.*

We show in Figure 3 an example for the pre-selection method. In the given conflict graph, $v_{1,1}$ and $v_{1,4}$ has no incident edge belonging to E_X; $v_{1,2}$ and $v_{1,3}$ each has an incident edge that belongs to E_X and connects to subgraphs SG_2 and SG_3. The computed vertex subset $CV_1 = \{v_{1,1}, v_{1,4}\}$. Assume that $v_{1,1}$ is pre-selected, $v_{1,2}$, $v_{1,3}$ and $v_{1,4}$ shall be deleted from the conflict graph. The reduced conflict graph includes only subgraphs SG_2 and SG_3.

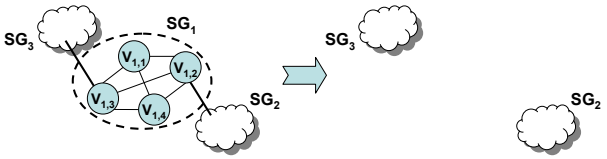


Figure 3: An example of the pre-selection method.

3.3.3 Computing the Connected Components

From the real circuits used in the experiments of [10], we observed that their corresponding conflict graphs are all sparse graphs. For example, the conflict graph of circuit C5 has 529039 vertices; however, it has 232958 connected components[1], and the largest connected component (in terms of the sum of the numbers of vertices and edges) contains only 22 vertices and 34 edges. Moreover, after performing the pre-selection method, a conflict graph is decomposed into even more connected components due to the removal of vertices in non-empty CV_i. Therefore, our 0-1 ILP approach can also be accelerated by first computing the connected components of a conflict graph and then solving a smaller 0-1 ILP problem associated with each connected component. The union of the solutions for individual connected components (and the pre-selected double-cut vias) is the overall solution to the DVI problem. It can be proved that the accelerated 0-1 ILP approach described here also solves the DVI problem optimally.

We use the depth-first search algorithm [17] to determine the connected components of a (reduced) conflict graph. Consider a connected component $G_{cc}(V_{cc}, E_{cc})$. Let $\mathrm{SV}_{cc} \subseteq \mathrm{SV}$ be the set of single vias such that $sv_i \in \mathrm{SV}_{cc}$ if it has at least one feasible double-cut via $fdv_{i,j}$ and $v_{i,j} \in V_{cc}$. The 0-1 ILP formulation for G_{cc} is as follows:

$$\begin{aligned}
&\text{Maximize} && \sum_{v_{i,j} \in V_{cc}} R_{i,j} \\
&\text{Subject to} && \sum_{fdv_{i,j}\text{'s of } sv_i} R_{i,j} \le 1,\ \forall sv_i \in \mathrm{SV}_{cc}; \\
& && R_{i,j} + R_{i',j'} \le 1,\ \forall (v_{i,j}, v_{i',j'}) \in E_{cc} \text{ and } i \neq i'; \text{ and} \\
& && R_{i,j} \in \{0,1\},\ \forall v_{i,j} \in V_{cc}. \qquad (5)
\end{aligned}$$

Lemma 3. *Let $RV_{01}(G)$ denote the maximum independent set obtained from a 0-1 ILP solution of G. $\bigcup_{G_{cc} \in G} RV_{01}(G_{cc})$ is an independent set and $|RV_{01}(G)| = \sum_{G_{cc} \in G} |RV_{01}(G_{cc})|$, where G_{cc} is a connected component of G.*

[1]In an undirected graph, two vertices are in the same connected component if and only if there exists a path between them.

Note that when computing the connected components of G ($\tilde{G}$), we consider both E_I and E_X ($\tilde{E}_I$ and $\tilde{E}_X$). Otherwise, the accelerated 0-1 ILP approach may not solve the original problem optimally or correctly. In the conflict graph as shown in Figure 4(a), if we ignore the edges ($v_{1,1}$, $v_{1,2}$) and ($v_{3,1}$, $v_{3,2}$), there will have three connected components as shown in Figure 4(b). Suppose $v_{1,1}$ and $v_{3,1}$ in connected components SG_1 and SG_3 are included in the final solution. When solving the 0-1 ILP problem of the connected component SG_2, any of $v_{1,2}$, $v_{2,1}$ and $v_{3,2}$ can be in the solution. However, only $v_{2,1}$ can lead to the optimal overall solution while the other two choices lead to illegal solutions. Therefore, the edges ($v_{1,1}$, $v_{1,2}$) and ($v_{3,1}$, $v_{3,2}$) should not be ignored during the computation of connected components.

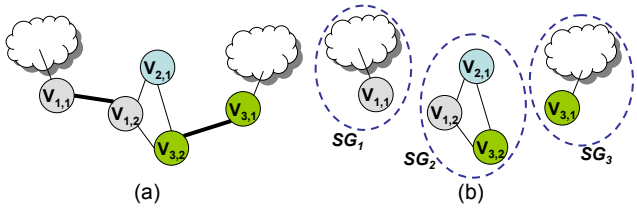


Figure 4: An improper breaking of the original problem.

3.4 Overall Approach

Given a conflict graph $G(V, E)$, we first pre-select a set of redundant vias $RV_{PS}(G)$ and then compute the connected components of the reduced conflict graph $\tilde{G}$. Next, we construct and solve the 0-1 ILP problem of each connected component. Let $RV_{01}(\tilde{G})$ denote the set of inserted redundant vias obtained by taking the union of all 0-1 ILP solutions. The final DVI solution is $RV_{PS}(G) \cup RV_{01}(\tilde{G})$.

Now we can state the following self-evident theorem based on Lemmas 1, 2 and 3.

Theorem 1. *The overall approach can also solve the DVI problem optimally.*

4. VIA DENSITY CONSIDERATION

In this section, we discuss a variant of the DVI problem, called DVI/VD, that considers the maximum via density constraint. As pointed out in [13], if the number of inserted redundant vias is not well controlled, it may violate the maximum via density constraint and adversely affect the yield and reliability of the design. We shall now extend our 0-1 ILP formulation to tackle this problem.

Definition 3 (Maximum via density constraint). *Each via layer is partitioned into a set of overlapping rectangular regions each of which has the same width W and height H, where W and H are process-dependent constants. For each region, its via density is defined as the number of vias located in it. The maximum via density constraint places an upper bound constraint U on the via density, where U is also a process-dependent constant.*

Definition 4 (Candidate violating region). *For a via region r_p, if the summation of the numbers of the involved single vias and the involved feasible redundant vias is greater than the maximum via density constraint U, r_p is a candidate violating region.*

Given a conflict graph $G(V, E = E_I \cup E_X)$, a candidate violating region r_p can be associated with a two-tuple (V_p, K_p), where $V_p \subseteq V$ and $K_p < |V_p|$. V_p is the vertex set corresponding to the $fdv_{i,j}$'s involved in the region r_p, and at most K_p of them can be simultaneously inserted into the design in order to meet the maximum via density constraint. The set of candidate violating regions, denoted R, in a design can be computed by slightly modifying the method proposed in [13]. Given G and R, we should add *density constraints* (6) into the 0-1 ILP formulation (4) in order to handle the maximum via density constraint as follows:

$$\text{Density constraint:} \quad \sum_{v_{i,j} \in V_p} R_{i,j} \leq K_p, \quad \forall r_p \in R. \quad (6)$$

In order to efficiently solve the DVI/VD problem, we have to modify the pre-selection method and the computation of connected components by modifying the definitions of the vertex subset CV_i and the connected component. The vertex subsets CV_i is redefined to be $\{v_{i,j} | v_{i,j} \in V_i$ is not in any candidate violating region and has no incident edges that belong to $E_X\}$. A connected component is redefined as follows. Given a (reduced) conflict graph $G(V, E = E_I \cup E_X)$ and a set R of candidate violating regions, we define another graph $G_{VD}(V, E_{VD})$, where $E_{VD} = \{(v_{i,j}, v_{i',j'}) | \exists r_p = (V_p, K_p) \in R$ such that $v_{i,j}$ and $v_{i',j'} \in V_p\}$. Vertices $v_{i,j}$ and $v_{i',j'}$ belong to the same connected component, if and only if there exists a path between $v_{i,j}$ and $v_{i',j'}$ in either G or G_{VD}. Therefore, we can also use the depth-first search algorithm [17] to determine the connected components of G. It should be noted that we have to consider the edge set E_{VD} for computing the connected components, but none of the edges in E_{VD} will be included in any connected components. Given a connected component $G_{cc}(V_{cc}, E_{cc} \subseteq E)$ and the subset $R_{cc} = \{r_p | r_p \in R$ and $V_p \cap V_{cc} \neq \emptyset\}$ of the candidate violating regions, we should add the following constraint into the 0-1 ILP formulation (5) in order to handle the maximum via density constraint:

$$\sum_{v_{i,j} \in V_p} R_{i,j} \leq K_p, \quad \forall r_p \in R_{cc}. \quad (7)$$

With these modifications, we can also use the flow described in Section 3.4 to efficiently and optimally solve the DVI/VD problem.

5. EXPERIMENTAL RESULTS

We used a 0.18μm technology that has 5 metal layers to implement each test circuit in our experiments. The set of test circuits we used is from [10] and its detailed information is shown in Table 2; for each test case, the first column shows the circuit name, "CU(%)" gives the core utilization, "#Nets" shows the number of nets, "#I/Os" gives the number of I/O pins, "#Vias" shows the total number of single vias, and "#A-Vias" gives the number of single vias that each has at least one feasible redundant via (i,e., |SV|). Finally, "#Objects" gives the total number of layout objects including pins, vias, blockages and wire segments.

Moreover, we used CPLEX [18] as the solver for 0-1 ILP. All the experiments were conducted on a Linux based machine with 2.4GHz processor and 4GB memory. In order to demonstrate the efficiency of our approaches in handling large-sized problems, we assumed that all single vias in each test circuit are allowed to be replaced with double-cut vias. Besides, we adopted the same maximum via density constraint as [13].

For the DVI and DVI/VD problems, we shall show the results of our 0-1 ILP approaches described in Section 3.4 and Section 4. Since our 0-1 ILP approaches use the *GCA* [10] to construct a conflict graph, the reported CPU times (in seconds) are measured after the completion of graph construction.

5.1 Efficiency Improvement

For each test case, the statistics on the given conflict graphs are shown in Table 3, where "$|V|$" gives the number of vertices in the conflict graph (i.e., the number of feasible redundant vias); "$|E_I|$" and "$|E_X|$" show the numbers of edges ($v_{i,j}$, $v_{i',j'}$) with $i = i'$ and $i \neq i'$, respectively. We can see that most of the edges are in E_I. We compare "$|E_I|$" with the column "#A-Vias" as shown in Table 2. For most of the test cases, the number of single vias that have at least one feasible redundant via each is about 54%$\sim$75% of the number of the edges of E_I. Therefore, using the double-cut constraints instead of the conflict constraints corresponding to the edges of E_I, the number of constraints of our 0-1 ILP ($\leq$ #A-Vias) can be reduced significantly.

Table 3: Statistics on conflict graphs

Case	$\|V\|$	$\|E_I\|$	$\|E_X\|$
C1	38829	32466	1524
C2	61369	47406	3024
C3	200051	160377	9030
C4	200311	136231	11122
C5	529039	392968	22797

Table 4 gives the statistics on the connected components of the DVI problems after performing the pre-selection method. The columns below "DVI" and "DVI/VD" show the statistics of the connected components of each test case for the DVI and DVI/VD problems, respectively. "$|\tilde{V}|$" and "$|\tilde{E}|$" respectively give the numbers of vertices and edges of the reduced conflict graph $\tilde{G}$ after applying the pre-selection method. "#CC" shows the number of connected components of $\tilde{G}$. The size of a connected component is measured by the summation of the number of vertices and the number of edges. "NUM" and "%" under "Small-CC" respectively show the number and the ratio of the connected components with fewer than 50 vertices and edges combined. Besides, "Max-CC" shows the numbers of vertices and edges of the largest connected component, denoted by $|V_{cc}|$ and $|E_{cc}|$, respectively.

Without via density constraints, we can see from Table 4 that almost all the vertices can be pre-selected and the sizes of all connected components are very small. However,

Table 2: The test cases

Case	CU(%)	#Nets	#I/Os	#Vias	#A-Vias	#Objects
C1	98	4309	20	24594	17522	218215
C2	70	5252	211	41157	28591	268669
C3	70	18157	85	127059	91727	933852
C4	95	17692	415	151912	102347	934073
C5	70	44720	99	357386	255301	2851612

when via density constraints are considered, the ratio of the small connected components is decreased and the size of the largest connected component is increased. It should be noted that for the DVI/VD problem, we have to consider the edge set E_{VD}, defined in Section 3.3.3, for computing the connected components of $\tilde{G}$, but none of the edges in E_{VD} will be included in any connected component of $\tilde{G}$. Although the sizes of the largest connected component are now much larger than those in the DVI problem, for most cases, they are much smaller than the original conflict graphs. (On the average, $|V_{cc}|$ and $|E_{cc}|$ are only 22% and 8% of $|V|$ and $|E|$, respectively) (i.e., the whole problem is decomposed into several smaller sub-problems). Therefore, as we shall see in Table 6, our 0-1 ILP approach is still very efficient in computing optimal DVI/VD solution.

Table 5 shows the efficiency improvement of our speed-up methods when applied to the 0-1 ILP approach for the DVI/VD problem. "With acceleration" shows the CPU times of our 0-1 ILP approach; on the other hand, "Without acceleration" shows the CPU times of our 0-1 ILP approach without speed-up methods. The CPU time for C5 is unavailable, since the corresponding 0-1 ILP problem of the original conflict graph was too large and the CPLEX ran out of the memory. We can see that the acceleration methods help to improve the efficiency of our 0-1 ILP approach significantly. With the acceleration methods the solution for C4 was generated in 48 seconds. On the other hand, without the acceleration methods, it took 202 seconds to generate a similarly optimal solution for C4.

Table 5: The Results of Speed-Up methods for DVI/VD

Case	Without acceleration	With acceleration
C1	7	3
C2	12	4
C3	158	5
C4	202	48
C5	-	3

5.2 Comparing with Existing Methods

For comparative studies, we implemented the maximum independent set (MIS) based approach [10], called *MIS-DVI*, for the DVI problem. This approach consists of the algorithm *GCA* for conflict graph construction and the heuristic *H2K* to solve the MIS problem on a conflict graph in an iterative manner. We used the qualex-ms [19] as the MIS solver and limited the sub-graph extracted at each iteration of *H2K* to consist of at most 225 vertices. The reported CPU time of *MIS-DVI* is also measured after the completion of graph construction. In [12], a more restrictive version of the DVI problem is considered; therefore, we do not compare with it. For the DVI/VD problem, we adapted the approach proposed in [13] and implemented a two-stage heuristic method: First, it uses *MIS-DVI* to insert as many redundant vias as possible. Then, it uses a 0-1 ILP approach proposed in [13] to eliminate redundant vias for satisfying the maximum via density constraint. We refer to this method as *Two-Stage*. Also, the reported CPU time of *Two-Stage* is measured after the completion of graph construction. *MIS-DVI* and *Two-Stage* inserted the same numbers of redundant vias as reported in [10] and [13], respectively.

Table 6 shows the results of the DVI and DVI/VD problems. The columns "Tool", "MIS-DVI" and "0-1 ILP" under "DVI" respectively show the results generated by a commercial tool reported in [10], *MIS-DVI* and our 0-1 ILP approach, for the DVI problem. The columns "Two-Stage" and "0-1 ILP" under "DVI/VD" respectively show the results generated by *Two-Stage* and our 0-1 ILP approach, for the DVI/VD problem. For each test case, "#RVI" gives the number of inserted redundant vias. "#D-RVI" under "DVI" indicates the difference between the numbers of redundant vias inserted by *MIS-DVI* and our 0-1 ILP approach for the DVI problem. Similarly, "#D-RVI" under "DVI/VD" gives the difference between the numbers of redundant vias inserted by *Two-Stage* and our 0-1 ILP approach for the DVI/VD problem. The columns labeled "T(s)" give the CPU times.

For the DVI problem, our 0-1 ILP approach is up to 34.7 times faster than *MIS-DVI*. Besides, our approach is always able to generate a DVI solution in which the number of inserted redundant vias is the same or larger (i.e., *#RVI* is the same or larger), as compared to the commercial tool and *MIS-DVI*. In particular, for each test case, our approach always generated an optimal solution, but the commercial tool and *MIS-DVI* did not. For the DVI/VD problem, our 0-1 ILP approach is also always able to generate an optimal solution. It can insert more redundant vias into the design than *Two-Stage*, with runtimes that are up to 35.3 times faster. Since C4 is the densest design, the largest connected component is significantly bigger than those of the other test cases (as shown in Table 4); therefore, the runtime for this case is significantly higher, but it is still 2.1 times faster than *Two-Stage*.

Although the 0-1 ILP problems are in general an intractable problem, according to the experimental results mentioned above, our approaches are efficient in optimally solving the DVI and DVI/VD problems.

6. CONCLUSIONS

In this paper, we have studied the double-cut via insertion problems with and without via density constraints. We

Table 4: Statistics on connected components of the DVI problems after applying pre-selection

Case	DVI							DVI/VD						
				Small-CC		Max-CC					Small-CC		Max-CC	
	$\|\tilde{V}\|$	$\|\tilde{E}\|$	#CC	NUM	%	$\|V_{cc}\|$	$\|E_{cc}\|$	$\|\tilde{V}\|$	$\|\tilde{E}\|$	#CC	NUM	%	$\|V_{cc}\|$	$\|E_{cc}\|$
C1	128	67	61	61	100	4	3	5905	5023	149	58	39	178	151
C2	182	98	84	84	100	6	5	8703	6204	205	89	43	229	173
C3	600	334	273	273	100	6	5	21867	16730	645	301	47	272	209
C4	1274	703	583	583	100	8	8	99136	74422	291	192	66	88671	66796
C5	1943	1083	822	822	100	8	11	31926	21597	1530	1037	68	133	107

Table 6: Results of the DVI problems

Case	DVI					DVI/VD			
	Tool	MIS-DVI[10]		0-1 ILP		Two-Stage[13]		0-1 ILP	
	#RVI	#RVI	T(s)	#D-RVI	T(s)	#RVI	T(s)	#D-RVI	T(s)
C1	14402	17461	5	+0	3	17074	6	+175	3
C2	25918	28507	11	+0	3	27881	12	+183	4
C3	80827	91461	86	+0	4	89960	88	+522	5
C4	91574	101765	86	+1	5	91134	100	+2106	48
C5	225142	254428	104	+1	3	252056	106	+712	3
Normalized			15.7X		1X		12.0X		1X

have shown that the double-cut via insertion problems can be formulated as a set of smaller zero-one integer linear program (0-1 ILP) problems. Each smaller problem can be solved independently and efficiently without sacrificing the optimality.

7. REFERENCES

[1] K. Rajkanan, "Yield analysis methodology for low defectivity wafer fabs," in *Proceedings of the International Workshop on Memory Technology, Design and Testing*, 2000, p. 65.

[2] N. Harrison, "A simple via duplication tool for yield enhancement," in *Proceedings of the International Symposium on Defect and Fault-Tolerance in VLSI Systems*, 2001, pp. 39–47.

[3] S. Raghvendra and P. Hurat, "DFM: Linking design and manufacturing," in *Proceedings of the International Conference on VLSI Design held jointly with International Conference on Embedded Systems Design*, 2005, pp. 705–708.

[4] T. Pompl, C. Schlünder, M. Hommel, H. Nielen, and J. Schneider, "Practical aspects of reliability analysis for IC designs," in *Proceedings of Design Automation Conference*, 2006, pp. 193–198.

[5] C. Christiansen, B. Li, J. Gill, R. Filippi, and M. Angyal, "Via-depletion electromigration in copper interconnects," *IEEE Transactions on Device and Materials Reliability*, vol. 6, no. 2, pp. 163–168, June 2006.

[6] *Reference Flow 5.0 and Reference Flow 6.0.* Taiwan Semiconductor Manufacturing Company (TSMC).

[7] Y. Zorian, D. Gizopoulos, C. Vandenberg, and P. Magarshack, "Guest editors' introduction: Design for yield and reliability," *IEEE Design and Test of Computers*, vol. 21, no. 3, pp. 177–182, 2004.

[8] D. Abercrombie, "Via doubling can help to stem yield loss." EE Times, October 10 2005.

[9] J. Wilson and W. Ng, "Improving design robustness with via doubling." Semiconductor International, July 1 2006.

[10] K.-Y. Lee and T.-C. Wang, "Post-routing redundant via insertion for yield/reliability improvement," in *Proceedings of Asia South Pacific Design Automation Conference*, 2006, pp. 303–308.

[11] F. Luo, Y. Jia, and W. W.-M. Dai, "Yield-preferred via insertion based on novel geotopological technology," in *Proceedings of Asia South Pacific Design Automation Conference*, 2006, pp. 730–735.

[12] H.-Y. Chen, M.-F. Chiang, Y.-W. Chang, L. Chen, and B. Han, "Novel full-chip gridless routing considering double-via insertion," in *Proceedings of Design Automation Conference*, 2006, pp. 755–760.

[13] K.-Y. Lee, T.-C. Wang, and K.-Y. Chao, "Post-routing redundant via insertion and line end extension with via density consideration," in *Proceedings of International Conference on Computer-Aided Design*, 2006, pp. 633–640.

[14] G. Xu, L.-D. Huang, D. Z. Pan, and M. D. F. Wong, "Redundant-via enhanced maze routing for yield improvement," in *Proceedings of Asia South Pacific Design Automation Conference*, 2005, pp. 1148–1151.

[15] H. Yao, Y. Cai, X. Hong, and Q. Zhou, "Improved multilevel routing with redundant via placement for yield and reliability," in *Proceedings of Great Lakes Symposium on VLSI*, 2005, pp. 143–146.

[16] F. Eisenbrand, S. Funke, N. Garg, and J. Könemann, "A combinatorial algorithm for computing a maximum independent set in a t-perfect graph," in *Proceedings of Symposium on Discrete Algorithms*, 2003, pp. 517–522.

[17] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, 2nd ed. The Massachusetts Institute of Technology Press, 2001.

[18] CPLEX. [Online]. Available: http://www.ilog.com

[19] QUALEX-MS package. [Online]. Available: http://www.busygin.dp.ua/npc.html