An Optimal Network-Flow-Based Simultaneous Diode and Jumper Insertion Algorithm for Antenna Fixing

Zhe-Wei Jiang, *Student Member, IEEE*, and Yao-Wen Chang, *Member, IEEE*

***Abstract*—As technology enters the nanometer territory, the antenna effect plays an important role in determining the yield and reliability of a VLSI circuit. Diode and jumper insertions are the most effective techniques to fix the antenna effect. However, due to the increasing design complexity and the limited routing resource, applying diode or jumper insertion alone cannot achieve a high antenna fixing rate. In this paper, we give a polynomial-time antenna violation detection/fixing algorithm by simultaneous diode and jumper insertion with minimum cost, which is based on a minimum-cost network-flow formulation. Experimental results show that our algorithm consistently achieves much higher antenna fixing rates than the state-of-the-art jumper and diode insertion algorithms alone.**

***Index Terms*—Design for manufacturability, interconnect, network flow algorithm, physical design.**

I. Introduction

MANUFACTURING reliability and yield in VLSI designs are becoming a crucial challenge as the feature sizes shrink into the nanometer scale. The antenna effect arising in the plasma process is an important problem in achieving a higher reliability and yield.

A. Antenna Effect

The antenna effect is caused by the charges collected on the floating interconnects which are connected to only a gate oxide. During the metallization, long floating interconnects act as temporary capacitors and store charges gained from the energy provided by fabrication steps such as plasma etching, chemical mechanical polishing, etc. If the collected charges exceed a threshold, Fowler–Nordheim tunneling current will discharge through the thin oxide and cause gate damage. On the other hand, if the collected charges can be released before exceeding the threshold through a low impedance path, such as a diffusion, the gate damage can be avoided. For example, considering the routing in Fig. 1(a), the interconnects are manufactured in the order of poly, metal 1, and metal 2. After manufacturing metal 1 [see Fig. 1(b)], the collected charges on the right metal 1 pattern may cause damage to the connected gate oxide. The discharging path is constructed after manufacturing metal 2 [see Fig. 1(c)], and thus, the charges can be released through the connected diffusion on the left side.

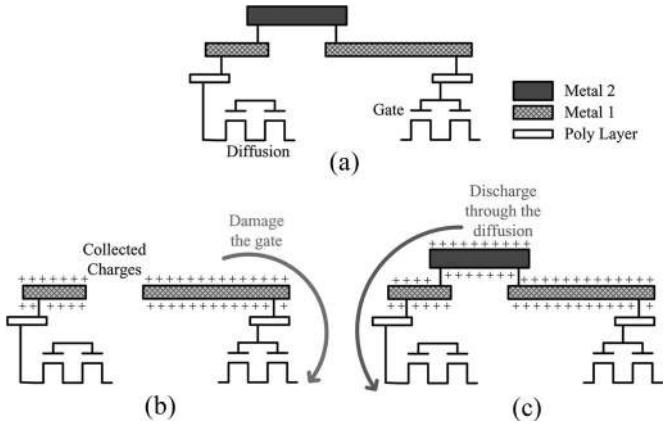

Fig. 1. Illustration of antenna effect: (a) An example routing. (b) Late stage of metal 1 layer pattern etching of figure (a). The collected charges on the right side of the metal 1 pattern may cause damage to the connected gate oxide. (c) Late stage of metal 2 layer pattern etching of figure (a). All the collected charges can be released through the connected diffusion on the left side.

There are three popular solutions proposed to reduce the antenna effect [5].

1) Jumper insertion: Break the signal wires with antenna violation and route them to the top metal layer. This approach reduces the collected charges during the manufacturing process but incurs two vias for each jumper.
2) Embedded protection diode: Add a protection diode on every input port for every standard cell. This approach prevents all input ports from the charge damage but consumes unnecessary areas when there is no antenna violation at the embedded input port.
3) Diode insertion after routing: Fixing only the wires with antenna violations will not waste routing resources. During wafer manufacturing, all the inserted diodes are floating (or ground). Since the input ports are high impedance, the charge on the wire flows through the inserted floating/ground diode.

Manuscript received August 4, 2007; revised November 6, 2007. This paper was presented in part at IEEE/ACM International Conference on Computer-Aided Design (ICCAD'06), November 2006. This work was supported in part by SpringSoft Inc., under Contracts NSC 94-2215-E-002-005, NSC 94-2215-E-002-030, and NSC 94-2752-E-002-008-PAE. This paper was recommended by Associate Editor J. Hu.

Z.-W. Jiang is with the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C. (e-mail: crazying@eda.ee.ntu.edu.tw).

Y.-W. Chang is with the Department of Electrical Engineering and the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C., and also with Waseda University, Kitakyushu 808-0135, Japan (e-mail: ywchang@cc.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/TCAD.2008.923246

The difference between diode and jumper insertions is the consumed resources of the fixed circuit. For jumper insertion, each jumper needs free spaces to route to the top metal layer, and it incurs at least two vias for each jumper. For diode insertion, the consumed resources are the free spaces on the substrate. If a violating wire lies above a space that can insert a diode, the diode is directly inserted below the wire. Otherwise, if there is no free space under the wire, extension wires are necessary to connect the violating wire to a diode insertion space [7], [8]. Both the vias and the extension wires will increase the driving load of the antenna violating wire, and thus, the incurred RC delay will reduce the circuit performance. In current nanometer technology, the induced RC delay of a via is several tens of times larger than that of 1-μm metal wire. Therefore, in order to minimize the cost of fixing the antenna violations, we shall apply both diode and jumper insertions and consider the interaction between them to minimize the cost for the fixing.

B. Previous Work

Maly *et al.* [11] translated the antenna condition detection problem into a layout analysis problem. It can be solved by a general-purpose design-rule checking program. However, the method does not indicate any measure to feedback the antenna information to the diode or jumper insertion. Shirota *et al.* [12], [13] proposed a rip-up and reroute method in a traditional router to reduce the antenna effect damage. Ho *et al.* [6] proposed full-chip routing with antenna avoidance. These works [6], [12], [13] reduce the antenna effects during the routing stage, whereas the works presented in [5], [7], [8], and [16]–[18] try to fix the antenna violations in the postlayout stage. Chen *et al.* [5] presented a heuristic to insert the diode under the wire with antenna violation. However, in modern high-density VLSI circuit, there is little free space for the "under-the-wire" diode insertion. Wu *et al.* [18] proposed a layer assignment technique to handle antenna avoidance by a tree-partitioning algorithm, but routing blockages are not considered in their algorithm. Su and Chang [14], [15] presented an optimal greedy jumper insertion algorithm that uses the minimum number of jumpers to fix the antenna violation on a spanning tree. Recently, Su *et al.* [16], [17] further presented a greedy optimal jumper insertion algorithm, called the bottom up jumper insertion with obstacles (BUJIO), which uses the minimum number of jumpers to fix the antenna violation on a Steiner tree with obstacles. Huang *et al.* [7], [8] solved the diode insertion and routing problem by a minimum-cost network-flow-based algorithm, called the diode insertion and routing by min-cost flow (DIRMCF). The violating wires, the routing grids, and the feasible diode positions are transformed into a flow network, and then, the problem is solved by the minimum-cost network-flow algorithm. Both the positions of inserted diodes and the extension wires can be determined through the resulting flow.

C. Motivation

In all the previous works [6]–[8], [14]–[18], the antenna violations are fixed by jumper or diode insertion alone, and the interaction between jumper and diode insertions is ignored. Considering the routing topology in Fig. 2(a) and the antenna bound of five unit length,[1] we need two jumpers for net 1, one jumper for net 2, and two jumpers for net 3 to fix the antenna violation. It requires totally five jumpers by jumper insertion alone [see Fig. 2(b)] or seven units of extension wire by diode insertion alone [see Fig. 2(c)] to fix the antenna violation. If we consider the interaction between diode and jumper insertions and fix the violations by simultaneous diode and jumper insertion (SDJI), however, the antenna effects can be fixed by merely one jumper and two units of extension wire [see Fig. 2(d)], which consumes much fewer resources than diode or jumper insertion alone.

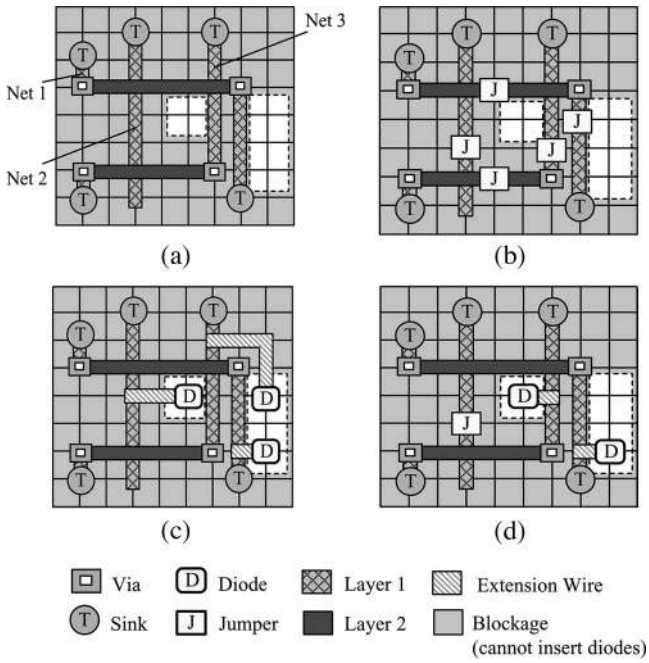

Fig. 2. Illustration of the consumed resources by jumpers and extension wires. Three violating wires, nets 1, 2, and 3, need to be fixed. (a) Three violating wires: Net 1 (2 jumpers needed), Net 2 (1 jumper needed), and Net 3 (2 jumpers needed). (b) Fix by jumper insertion: no. of jumpers $= 2 + 1 + 2 = 5$. (c) Fix by diode insertion: length of extension wire: $= 1 + 2 + 4 = 7$. (d) Fix by simultaneous diode/jumper insertion: no. of jumpers $= 1$, length of extension wire $= 2$.

In [5], [7], and [8], one inserted diode is assumed to protect all input ports that are connected to the same output port. This assumption is not always true in real circuits. Such as the tree representation of a given net in Fig. 3, both antenna weights (which could be wire-area-to-gate-size ratios, wire areas, or any other antenna measure) of segments s_1 and s_2 exceed $L_{\max}$, where $L_{\max}$ denotes the upper bound for antenna (i.e., any antenna measure larger than $L_{\max}$ will violate the antenna rule). If we insert only a diode on s_1 or s_2, after the metallization of metal layer 1, s_1 and s_2 are still two individual segments, and thus, the collected charges on the other segment will still cause damage to the connected input port. That means, in the case of Fig. 3, we must insert at least two diodes to fix the antenna violation. Thus, a more accurate algorithm is needed

[1]Note that the antenna bound could also be measured by wire-area-to-gate-size ratios, wire areas, or any other antenna measure.

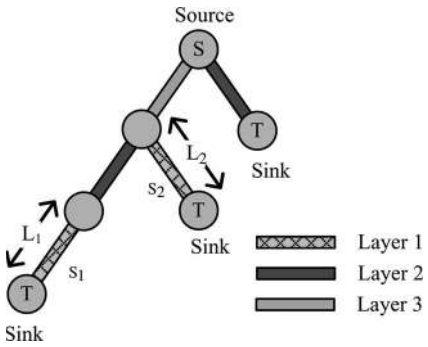

Fig. 3. Example that a net needs multiple diodes to fix the antenna violation. If both L_1 and L_2 exceed the antenna threshold $L_{\max}$, at least two diodes must be connected to s_1 and s_2 separately to fix the antenna violation.

to analyze the number of diodes needed to fix the antenna effect.

D. Our Contributions

In this paper, we propose a minimum-cost network-flow-based algorithm by SDJI to aviod/fix antenna violation. The proposed algorithm can find an optimal solution in polynomial time. In particular, it guarantees to fix the antenna violations if one feasible solution exists. We also present a more accurate model to analyze the exact number of diodes needed for antenna fixing. Experimental results show that our work achieves higher antenna fixing rates and incurs lower costs for antenna avoidance/fixing. Compared with the state-of-the-art jumper insertion algorithm, BUJIO, alone in [16] and [17], our algorithm achieves more than 99.6% fixing rate even with a dense 95% diode blockage rate, whereas BUJIO obtains only 63.4% fixing rate on average (due to the significant blockages for jumper insertion), based on a set of MCNC layouts obtained by the publicly available router MR [4], [10]. Our algorithm also consistently achieves higher antenna fixing rates than the diode insertion algorithm, DIRMCF, alone, under various diode blockage rates.

The remainder of this paper is organized as follows. Section II formulates the problem of detecting/fixing the antenna effects with SDJI. Section III presents an optimal algorithm for the proposed problem. Section IV reports the experimental results. Finally, the conclusions are given in Section V.

II. Problem Formulation

To detect/fix antenna violations, we have to check if the effective conductor connecting to a gate oxide exceeds a threshold $L_{\max}$. Here, $L_{\max}$ can be measured in wire-area-to-gate-size ratio, wire area, wirelength, or any model of the strength of antenna effect caused by conductors, same as that in [16] and [17]. To simplify the discussion, we assume that all sinks on a net are connected to a gate terminal, whereas the source is connected to diffusion (those sinks connecting to diffusion can be ignored since they will not cause any antenna violation for current technology). Aside from checking the existence of the antenna violation, we have to know where the diodes should be connected to protect the gate terminals. A violating-wire set (VWS) is defined as a group of connected wire segments, where exactly one diode needs to be connected to one of these wire segments to fix the antenna violation. Alternately, we can fix a VWS by one or more jumpers instead of one diode. Note that one net can be divided into several VWSs since a net may need multiple diodes to fix the antenna effect, as mentioned in Section I. Take Fig. 3 as an example. The given net contains two VWSs: One contains s_1, and the other s_2. Thus, exactly two diodes are needed for the given net.

Vias and metal wires can interplay with each other in many different ways. In this paper, we try to minimize the total delay induced by extra vias and metal wires. To evaluate the total induced delay when we fix the antenna violation, we define the cost function Φ composed of the total wirelength of extension wires (for diodes) and the total number of jumpers as follows:

$$\Phi = \mu \times (\beta \times m_J + l_E) \tag{1}$$

where m_J is the number of jumpers inserted to fix the antenna violations, l_E is the total wirelength of extension wires induced by diode insertion, β is a user-specified parameter for the ratio of the jumper induced delay to the unit-length extension-wire induced delay, and μ is the unit-length extension-wire induced delay. Note that the extension wire does not lie on a signal propagation path since it always connects to a diode. According to the Elmore delay model, only the capacitance of the extension wire is considered, and thus, the induced delay is linearly proportional to the length of the extension wire. This concept is similar to that in [7] and [8], which minimizes the total wirelength. It should be noted that (1) is merely an example modeling of the interplay of diode and jumper insertions; it will be clear that our algorithm also applies to the cases with different cost models.

With the aforementioned definitions, we can formulate the addressed problem as follows:

- **Problem ASDJI:** Given a routing topology T, an antenna threshold $L_{\max}$, and a set of diode insertion positions D, identify all the antenna violations in T and find a set of feasible jumper positions, a set of diode positions $D' \subset D$, and a set of paths P connecting some VWSs to the corresponding diode positions, such that the total induced cost is minimized, and all the VWSs are either broken into smaller antenna-safe segments by inserted jumpers or connected to inserted protection diodes.

III. Algorithms

We propose a two-phase method to solve the antenna effect detection/fixing with SDJI (ASDJI) problem. The first phase applies the wire violation detection (WVD) algorithm, and the second uses the SDJI algorithm. In the WVD algorithm, all VWSs in the given routing topology are identified, and then, in the SDJI algorithm, the identified VWSs are fixed by either diode or jumper insertion with the minimum delay cost. We explain the two algorithms in Sections III-A and III-B, respectively.

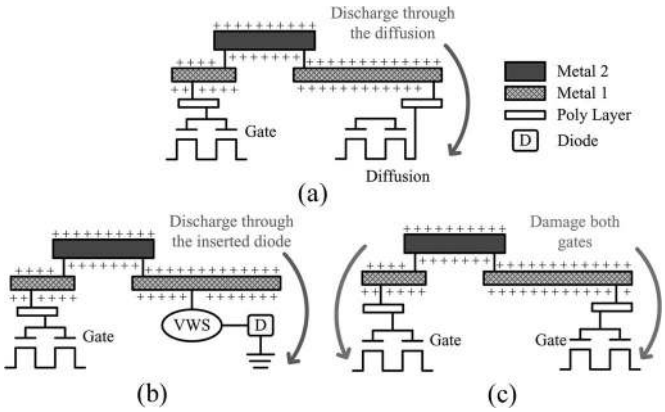

Fig. 4. Illustration of different cases of the connected component C_i in WVD algorithm. (a) C_i is connected to diffusion. The collected charges can be discharged through the diffusion. (b) C_i is connected to another VWS. The collected charges may be discharged through the inserted diode of the connected VWS. (c) C_i is not connected to any diffusion or VWSs. The collected charges will cause damage to connected gates.

ALGORITHM: WIRE VIOLATION DETECTION (WVD)
Input: Routing topology (T)
Antenna upper bound (L_{max})
Number of layers (n_{layer})
Output: Set of identified VWS's (S_{viol})
begin
1 Graph $G \leftarrow \varnothing$;
2 $S_{viol} \leftarrow \varnothing$;
3 **for** layer $i \leftarrow 1$ **to** n_{layer} **begin**
4 add segments in layer i into G;
5 **for** every connected component C in G which contains at least one sink **begin**
6 $W_C \leftarrow$ total weight of C;
7 **if** $W_C > L_{max}$ **and** C is not connected to any sources or any other VWS $\in S_{viol}$ **then**
8 $S_{viol} \leftarrow S_{viol} \cup C$;
9 **end**
10 **end**
11 **return** S_{viol};
end

Fig. 5. WVD algorithm.

A. WVD

We explain how to identify all the VWSs in this section. In our assumption, the antenna violation happens when the collected charges connected to a gate terminal exceed the antenna threshold during the metallization. Thus, the VWS should be identified by analyzing the intermediate topologies between the metallization of each metal layer. For example, after the metallization of metal layer 2, only segments in metal layers 1 and 2 are fabricated. At this intermediate stage, we should compute the collected charges on the segments in metal layers 1 and 2 and check whether the summation of the collected charges exceeds the antenna threshold. With the nature of metallization, the metal layers are fabricated from the bottom to the top layers. Thus, the proposed algorithm makes use of this nature and analyzes the intermediate topologies between the completeness of each metal layer.

The WVD algorithm is summarized in Fig. 5. The graph G is used to record the intermediate topologies between the metallization of each metal layer, and the set S_{viol} records the identified VWSs. For the main loop in lines 3–10, the segments in each metal layer are added into G in the increasing order of layers. In lines 5–8, since only the collected charges connected to a sink may cause the antenna violation, the connected components which contain at least one sink are extracted from G, and the total antenna weight W_{Ci} of each extracted connected component C_i is then computed. If $W_{\text{Ci}} > L_{\text{max}}$, the collected charges of C_i exceed the antenna threshold, and three cases need to be checked (see lines 7–8 and an illustration in Fig. 4).

Case 1) C_i is connected to a source node [Fig. 4(a)]. If the connected component C_i is connected to a source node, the collected charges of C_i can be discharged through the diffusion terminal, and thus, no antenna violation will occur.

Case 2) C_i is not connected to any source nodes but is connected to another VWS [Fig. 4(b)]. For this case, if the connected VWS is fixed by diode insertion, the collected charges of C_i can be discharged through the inserted diode and, thus, will not cause any antenna violations. However, if the connected VWS is fixed by jumper insertion, the collected charges may still cause the antenna violation, since jumper insertion will not create any discharging paths. In this phase, the case discussed here is treated as antenna-safe segments, and an enhanced technique is applied to solve this case in the second phase.

Case 3) C_i is not connected to any source nodes or any other VWSs [Fig. 4(c)]. In this case, the collected charges would damage the gate terminals, and thus, an antenna violation is identified. The connected component C_i is classified as a VWS and is added into S_{viol}.

We have the following theorem for the time complexity of our WVD algorithm.

Theorem 1: The time complexity of the WVD algorithm is $O(|T|^2 \cdot n_{\text{layer}})$, where $|T|$ is the total number of segments for the given topology T and n_{layer} is the number of layers.

Proof: As shown in Fig. 5, lines 1–2 take constant time. For the loop between lines 3 and 10, line 4 spends at most $O(|T|^2)$ time to add all segments into G. The inner loop (lines 5–9) needs $O(|T|)$ time since every segment is checked at most once to compute the corresponding antenna weight W_{Ci}. Since the loop between lines 3 and 10 executes n_{layer} times, we can conclude that the time complexity of the WVD algorithm is $O(|T|^2 \cdot n_{\text{layer}})$.

B. SDJI

In this phase, we fix every VWS identified in the first phase by SDJI with the minimum cost. Since the optimal jumper insertion solution for a VWS can be computed by the BUJIO algorithm [16], [17], we make use of the optimal solution of each VWS to minimize the cost induced by antenna fixing.

Inspired by the DIRMCF algorithm [7], [8], we also consider the jumper cost in the flow network, and thus, the jumper costs and the extension wire costs (for diodes) can be handled at the

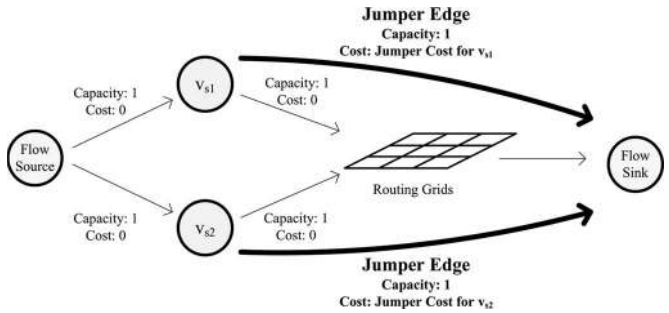

Fig. 6. Example to consider diodes and jumpers at the same time. A jumper edge is added for each VWS node, and the jumper cost is modeled as the edge cost.

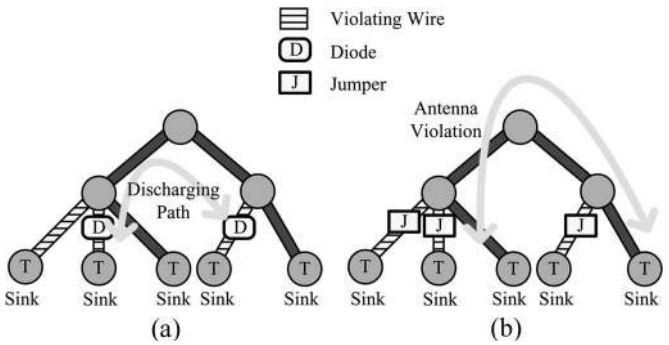

Fig. 7. Example to illustrate the interaction between diode and jumper insertions. (a) All VWSs are fixed by diode insertion. The charges on the remainder of the net can be discharged through the inserted diodes. (b) All VWSs are fixed by jumper insertion. The charges on the remainder of the net may still cause the antenna effect.

same time. For every VWS identified in the first phase, the BUJIO algorithm is applied to compute the number of jumpers m_J needed to fix the antenna violation. The jumper cost is calculated by $\beta \times m_J$. Then, we add a jumper edge for each VWS to model the jumper cost. Consider the example shown in Fig. 6 with two VWSs, which are represented by the VWS nodes v_{s1} and v_{s2}. The edges with unit capacity and zero cost are constructed from v_{s1} and v_{s2} to the routing grids, and thus, the resulting flow which goes through the routing grids determines the diode positions and the routing of extension wires connected to the protected VWS. Integrating the jumper costs into the flow network, one jumper edge with unit capacity is added from each VWS node to the sink of the network. The costs of the jumper edges are assigned to the optimal jumper costs computed by the BUJIO algorithm. Instead of going through the routing grids, the resulting flow now can alternately go through the jumper edge, which means that lower costs can be achieved if the corresponding VWS is fixed by jumper insertion.

However, even if the preceding algorithm is applied, some antenna violations may remain in the routing topology. Considering the example shown in Fig. 7, the tree representation of a net contains two identified VWSs. As mentioned in Case 2) of Section III-A, for a given net N, if at least one of the contained VWSs is fixed by diode insertion [see Fig. 7(a)], the collected charges of the remainder of N can be discharged through the inserted diodes, and thus, no antenna violation remains. In contrast, if all the contained VWSs of N are fixed by jumper insertion [see Fig. 7(b)], no discharging path is created, and thus, some antenna violation may remain on N if the collected charges of the remainder of N exceed the antenna threshold $L_{\max}$. Through this example, it is obvious that an extra jumper cost δ_N is needed for the remainder of N when all the contained VWSs are fixed by jumper insertion. Consider a net N with m identified VWSs. We define $c_J(N)$ as the optimal jumper cost for fixing net N, and $c_J(x)$ as that for fixing a VWS, x. The extra cost δ_N for net N can be computed by $\delta_N = c_J(N) - (\sum_{i=1}^{m} c_J(x_i))$.

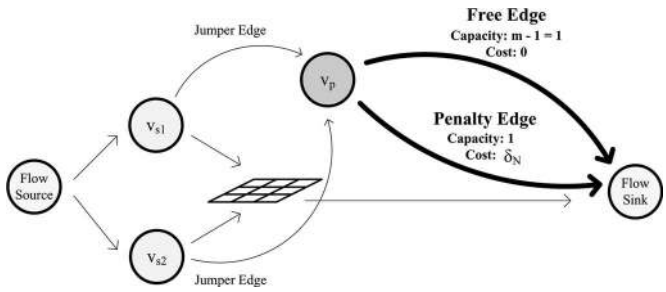

Fig. 8. Flow network to handle the extra jumper costs. A penalty node $v_{\rm p}$, a free edge, and a penalty edge are added for each net. The extra cost δ_N is modeled as the edge cost of the penalty edge.

In the SDJI algorithm, the extra cost δ_N should be added into the fixing cost when all the contained VWSs of net N are fixed by jumper insertion. To achieve this objective, a penalty node $v_{\rm p}$ is constructed for each net. Considering the example shown in Fig. 8, the flow network models a net N with $m = 2$ VWSs, represented by v_{s1} and v_{s2}. The jumper edges are connected to $v_{\rm p}$ instead of the sink of the flow network. Two edges, a free edge and a penalty edge, are connected from $v_{\rm p}$ to the sink of the network. For the free edge, the capacity is $m-1$ and the cost is zero. For the penalty edge, the capacity is one, and the cost is δ_N for net N. With this flow network, if the resulting flow finds fewer than m VWSs to be fixed by jumper insertion, no extra cost will be induced. If the resulting flow finds exactly m VWSs to be fixed by jumper insertion, however, the extra cost δ_N will be induced.

C. Overall Design Flow

Given the routing topology T, the antenna threshold $L_{\max}$, and a set of diode insertion positions D, the ASDJI problem can be solved by the design flow shown in Fig. 9. First, for the given T and $L_{\max}$, the VWSs can be identified by the WVD algorithm proposed in Section III-A. Second, the optimal jumper positions and costs to fix each VWS and the extra costs δ_N for each net N are computed by the BUJIO algorithm. Then, the flow network $G(V, E)$ is constructed as follows.

1) Construct a flow source, a flow sink, a representing node v_s for each VWS, and a grid node for each routing grid point. The grid nodes can be categorized into three types: v_x represents the grid point occupied by a violating wire; v_d represents the grid point feasible for diode insertion; and v_f represents the other grid point not occupied by the routed segments or routing blockages. The capacity of each grid node is equal to one.

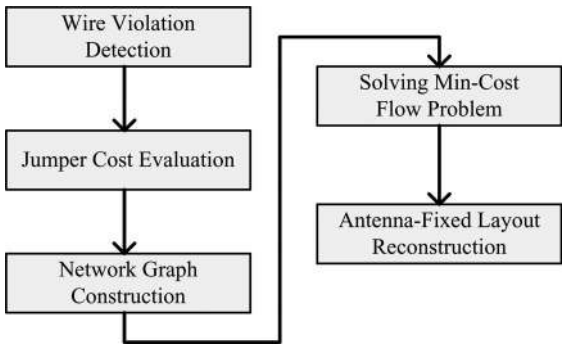

Fig. 9. Overall design flow.

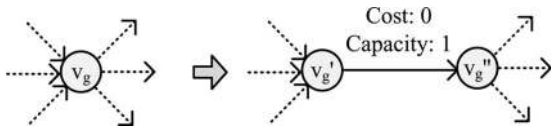

Fig. 10. Modeling the grid node capacity.

2) For each net containing at least one VWS, construct a penalty node v_{p}.
3) Construct the grid edges (v_{x_i},v_{f_j}), (v_{f_i}, v_{f_j}), and (v_{f_i}, v_{d_j}) between neighboring grid points. These edges represent all the possible routing directions of extension wires. All the grid edge capacities are equal to one, and all the costs are equal to the distance between the two grid points.
4) Construct the edges (source, v_{s_i}), (v_{s_i}, v_{s_j}), and (v_{d_i}, sink). All the edge capacities are equal to one, and all the costs are equal to zero.
5) Construct the jumper edges from each v_{s_i} to the corresponding v_{p_i} with unit capacity and corresponding jumper cost. The free edge and the penalty edge from v_{p_i} to the flow sink are constructed as described in Section III-B.

In particular, the capacity of a grid node (in step 1) can be modeled as in the example shown in Fig. 10. For a grid node v_g, we decompose it into two nodes, v'_g and v''_g, and connect from v'_g to v''_g. All incoming edges of v_g are now connected to v'_g, and all outgoing edges are connected from v''_g. The cost and capacity of the edge between v'_g and v''_g are set to zero and one, respectively. By this model, we can ensure that no more than one extension wire will go through v_g, and thus, the extension wires will not cross each other.

After constructing the flow network G, the optimal antenna fixing result can be determined by the minimum-cost network-flow algorithm. The diode and jumper positions can be extracted by checking the resulting flows on the edges (v_{s_i}, v_{p_i}) and (v_{d_i}, sink). The extension wire routing can be extracted by checking the flows on the grid edges. The antenna fixing result with SDJI can be concluded in the following theorem.

Theorem 2: For a routing topology T with m identified VWSs, if the value of the resulting flow f of the SDJI algorithm is equal to m, all the antenna violations can be fixed with the minimum cost. In contrast, if the value of the resulting flow f is less than m, no feasible solution exists to completely fix the antenna effect in T by SDJI.

For modern VLSI designs, there are two commonly used design rules for antenna violation checking, namely, partial antenna ratio (PAR) and cumulative antenna ratio (CAR) [2], [18]. The antenna ratio is defined as the antenna area divided by the connected gate area. The difference between PAR and CAR is that PAR considers antenna damage to gates on only one layer, whereas CAR accumulates the effect to gates from each metallization step [2]. Since the proposed WVD algorithm analyzes the intermediate topologies between the metallization steps, our algorithm can easily handle the CAR by treating it as the antenna upper bound. Furthermore, our algorithm can also handle the PAR problem with minor modifications for our design flow. For example, if we consider PAR on metal layer 3, we first group the gates and the connected metal wires on metal layers 1 and 2 as pseudogates. Then, applying the WVD algorithm, we set the antenna upper bound to the given PAR and perform the loop between lines 3 and 10, as shown in Fig. 5, only once on metal layer 3. Therefore, the VWSs are detected according to the given PAR. We can finally ungroup all pseudogates and apply the SDJI algorithm to fix all antenna violations with the PAR consideration as well.

We have the following theorem for the time complexity of our SDJI algorithm.

Theorem 3: The time complexity of the SDJI algorithm is $O(VE\ \mathtt{lg}(V^2/E)\mathtt{lg}(V))$, where V is the number of grid points and E is the number of edges among grid points.

Proof: The time complexity for BUJIO to compute the jumper positions for each VWS is $O((V'+D)\mathtt{lg}D)$ in [16] and [17], where D is the number of jumper obstacles and V' is the number of tree nodes in the tree representation of the VWS. In a grid-based routing model, the number of tree nodes and the number of obstacle points are smaller than that of grid points. Let V be the number of grid points in the given routing topology. The time complexity can be rewritten as $O(V\mathtt{lg}V)$ since $V'=O(V)$ and $D=O(V)$. The construction steps of the flow network can be performed in $O(E)$ time, and solving the minimum-cost network-flow problem requires $O(VE\ \mathtt{lg}(V^2/E)\mathtt{lg}(V))$ time [3]. Therefore, the time complexity of the SDJI algorithm is dominated by the minimum-cost network-flow algorithm.

D. Flow Network Pruning Technique

In the constructed flow network, the jumper cost is the cost upper bound to fix a VWS, and thus, it can be used to reduce the search space of the minimum-cost network-flow problem. Evaluating the worst case of jumper insertion, the cost upper bound $\hat{\delta}_s$ for a VWS s of net N can be computed by

$$\hat{\delta}_s = \delta_N + c_J(s) \tag{2}$$

where δ_N is the extra jumper cost of the net N and $c_J(s)$ is the optimal jumper cost for fixing s. With this upper bound, we have the following theorem.

Theorem 4: For every grid node v in the flow network $G(V,E)$ of SDJI, if the distance from v to every VWS s is larger than the corresponding $\hat{\delta}_s$, v can be pruned without loss of the solution optimality for the ASDJI problem.

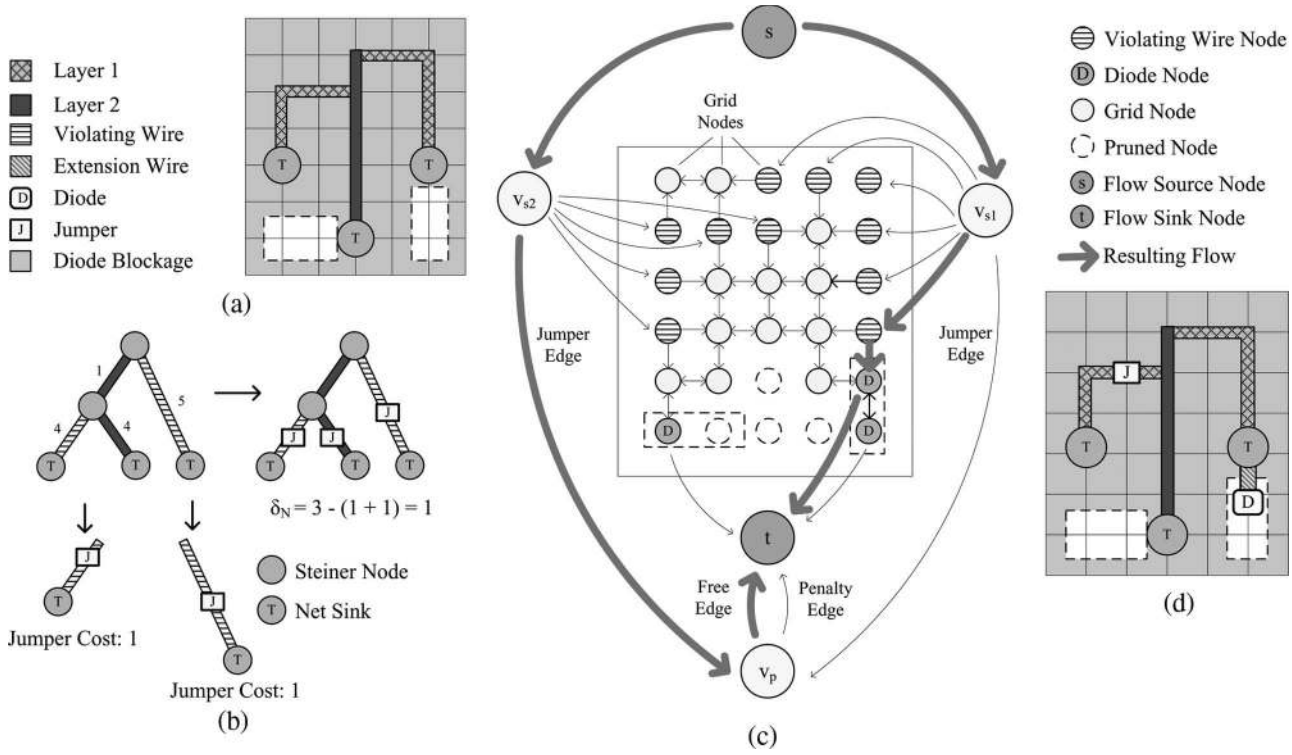

Fig. 11. Illustration of the proposed algorithm: (a) The given routing topology. (b) Calculation of the jumper cost for each VWS and the extra jumper penalty. (c) The constructed network graph and the resulting flow. The grid nodes are extracted from the grid points in layer 1 of figure (a). (d) The resulting layout by SDJI.

Proof: For a VWS s, with its representing node v_s, of the net N, we will connect v_s to a penalty node through a jumper edge and grid nodes through grid edges during the construction of the flow network G. Thus, for a resulting flow that enters v_s, it must leave v_s through either the connected jumper edge or grid edges. If the resulting flow goes through the jumper edge, the largest possible cost is $\hat{\delta}_s = \delta_N + c_J(s)$ since the flow may go through both the jumper edge (cost $c_J(s)$) and the penalty edge (cost δ_N). That means, for a grid node v, if the distance from v_s to v is larger than $\hat{\delta}_s$, the flow that enters v_s will result in a larger cost if it goes to the grid node v rather than leave v_s through the jumper edge. Since the SDJI algorithm finds the minimum-cost network-flow on G, the optimal flow that enters v_s will not go through v since it can easily choose to go through the jumper edge and get a smaller cost. In other words, if the distance from v to every VWS s is larger than the corresponding $\hat{\delta}_s$, we can remove v and all its connected edges from G since no optimal flow will go through v.

E. Complete Example

Fig. 11 shows an example to illustrate the overall design. We assume that both a jumper and a unit-length extension wire induce one unit delay. Consider the given routing topology with exactly one net in Fig. 11(a) and the tree representation in Fig. 11(b). Applying the WVD algorithm, two VWSs are identified. By the BUJIO algorithm, each VWS needs one jumper to fix the antenna violation, and thus, both the costs of the jumper edges are set to one. The number of jumpers needed to fix the whole routing tree is three, and the extra jumper cost δ_N is equal to one. In Fig. 11(c), to construct the flow network, the grid nodes and edges are first extracted from the grid points in layer 1 of Fig. 11(a). Then, the jumper edges are constructed for each VWS, and the penalty nodes, the penalty edges, and the free edges are constructed for each net. Since the number of VWSs in the given net is two, both the capacities of the penalty edge and the free edge are set to one, and the cost of the penalty edge is set to $\delta_N = 1$. After we construct the flow network, the minimum-cost network-flow algorithm is applied, and both the value and the cost of the resulting flow are equal to two. The optimal fixing solution is finally shown in Fig. 11(d).

TABLE I
MCNC BENCHMARK STATISTICS

Circuit	Size (μm^2)	# Layers	# Nets	# Pins
s5378	435×239	3	1693	4818
s9234	404×225	3	1476	4260
s13207	660×365	3	3777	10776
s15850	705×389	3	4470	12793
s38417	1144×619	3	11308	32344
s38584	1295×672	3	14753	42931

IV. EXPERIMENTAL RESULTS

The proposed algorithm was implemented in the C++ language on a 1.2-GHz SUN Blade 2000 machine with 8-GB memory.

The statistics of the benchmark circuits are listed in Table I. Six test cases are chosen from the MCNC benchmarks since only these test cases record the source and sink information for each net. The column "Circuit" denotes the circuit name, "Size" denotes the circuit dimension, "# Layers" denotes the number of routing layers, "# Nets" denotes the number of nets, and "# Pins" denotes the number of pins.

TABLE II
COMPARISON WITH BUJIO

Circuit Name	L_{max} (μm)	Total # Viol.	BUJIO [16]			Our Work											
						Blockage Rate: 80			Blockage Rate: 85			Blockage Rate: 90			Blockage Rate: 95		
			# Fixed Viol.	Fixing Rate (%)	Jumper Cost	# Fixed Viol.	Fixing Rate (%)	Total Cost	# Fixed Viol.	Fixing Rate (%)	Total Cost	# Fixed Viol.	Fixing Rate (%)	Total Cost	# Fixed Viol.	Fixing Rate (%)	Total Cost
s5378	50	95	65	68.42	810	95	100	182.76	95	100	236.28	95	100	516.72	95	100	624.24
	100	49	44	89.80	1005	49	100	20.88	49	100	37.92	49	100	140.64	49	100	120.00
s9234	50	56	34	60.71	435	56	100	153.48	56	100	81.00	56	100	485.16	56	100	231.24
	100	22	17	77.27	450	22	100	15.00	22	100	30.72	22	100	93.36	22	100	118.92
s13207	50	164	86	52.44	900	164	100	271.92	164	100	453.36	164	100	976.20	164	100	863.40
	100	83	51	61.45	975	83	100	29.40	83	100	215.64	83	100	203.52	83	100	444.00
s15850	50	182	93	51.10	1395	182	100	628.92	182	100	550.56	182	100	1007.76	182	100	1188.72
	100	98	54	55.10	1365	98	100	106.80	98	100	10.80	98	100	153.36	98	100	68.76
s38417	50	406	231	56.90	2295	405	99.75	1008.36	403	99.26	1226.04	401	98.77	3190.56	396	97.54	3158.76
	100	184	122	66.30	2210	184	100	97.68	183	99.46	95.40	183	99.46	486.12	182	98.91	643.92
s38584	50	550	341	62.00	2265	550	100	1379.04	550	100	1332.36	550	100	4008.48	550	100	3990.00
	100	283	167	59.01	4515	283	100	176.76	283	100	172.32	283	100	753.96	283	100	1027.56
			Avg.	63.38		Avg.	99.98		Avg.	99.89		Avg.	99.85		Avg.	99.69	

TABLE III
COMPARISON WITH DIRMCF FOR THE 80% DIODE BLOCKAGE RATE

Circuit Name	L_{max} (μm)	Total # Viol.	DIRMCF [7]					Our Work						
			# Fixed Viol.	Fixing Rate (%)	# Diodes	E. Wire Cost (μm)	CPU Time (s)	# Fixed Viol.	Fixing Rate (%)	Jumper Cost	# Diodes	E. Wire Cost (μm)	Total Cost	CPU Time (s)
s5378	50	95	92	96.84	92	154.80	2.6	95	100	105	88	77.76	182.76	2.2
	100	49	49	100	49	20.88	1.4	49	100	0	49	20.88	20.88	2.9
s9234	50	56	55	98.21	55	241.92	3.3	56	100	75	51	78.48	153.48	2
	100	22	22	100	22	18.00	0.9	22	100	15	21	0	15.00	0.7
s13207	50	164	158	96.34	158	212.40	47.2	164	100	120	156	151.92	271.92	34.9
	100	83	83	100	83	56.88	11.1	83	100	15	82	14.40	29.40	7.7
s15850	50	182	181	99.45	181	419.76	41	182	100	315	162	313.92	628.92	53.4
	100	98	98	100	98	38.08	29.5	98	100	60	94	46.80	106.80	32.5
s38417	50	406	396	97.54	296	887.04	54.7	405	99.75	495	374	513.36	1008.36	123.1
	100	184	183	99.46	183	95.76	52.6	184	100	30	182	67.68	97.68	84.8
s38584	50	550	541	98.36	541	1581.12	389.3	550	100	780	501	599.04	1379.04	253.9
	100	283	283	100	283	223.20	481.1	283	100	45	280	131.76	176.76	88.8
			Avg.	98.85				Avg.	99.98					

The minimum-cost network-flow solver used is LEDA 4.1 [1]. The input routing results of the test cases were taken from the multilevel routing results [10]. According to the TSMC 0.25-μm technology file, the jumper-to-wire ratio β in (1) was set to 15 for all the experiments. The antenna threshold $L_{\max}$ set in [6] is 100 μm, and in our experiments, 50 and 100 μm were both tested. To reflect modern design complexity, we randomly increase the diode blockage rate of each circuit to 80%, 85%, 90%, and 95%. We compared our work with the jumper insertion algorithm BUJIO [16], [17] and the diode insertion algorithm DIRMCF [7], [8]. We integrated both works with our WVD algorithm to identify the antenna VWSs. The experimental results show that our work achieves very high antenna violation fixing rates even in high-density circuits. We also performed several experiments to analyze the empirical runtime and induced delays of our algorithm and the effectiveness of the pruning technique proposed in Theorem 4.

A. Effectiveness of SDJI Algorithm

Table II gives the comparison of the antenna violation fixing rates between BUJIO and our work. Columns 1, 2, and 3 give the circuit name of each test case, the antenna threshold $L_{\max}$, and the numbers of antenna violations, respectively. Columns 4, 7, 10, 13, and 16 give the numbers of fixed antenna violation; columns 5, 8, 11, 14, and 17 give the fixing rates; and columns 6, 9, 12, 15, and 18 give the fixing costs of BUJIO and our work with different diode blockage rates. Note that for jumper insertion alone, the diode blockage rate would not influence the fixing result since jumper insertion only consumes the free spaces in the routing layers above the violating wires. The fixing rate is calculated by (# fixed antenna violations)/(# antenna violations). It is not surprising that BUJIO achieves only 63.38% fixing rate on average, since the routing layouts are usually too dense to find feasible jumper positions. In contrast, our work achieves more than 99.6% fixing rate even with the 95% diode blockage rate. It should also be noted that the fixing costs of BUJIO and our work cannot be compared directly since BIJIO usually fixes significantly fewer antenna violations than our work.

Tables III and IV give the comparison of the antenna-fixing results between DIRMCF and our work in 80% and 90% diode blockage rates, respectively. In the tables, column "# diodes" gives the numbers of diodes used to fix the antenna violations, column "E. Wire Cost" gives the total length of extension wires, and column "Jumper Cost" gives the jumper cost to fix the antenna violations, which is calculated by $\beta\times$ (number of jumpers used). Column "Total Cost" gives the cost to fix the antenna violations, which is the summation of the jumper cost and the extension wire cost. Note that the total cost in DIRMCF is equal to the extension wire cost. Column "CPU Time" gives the runtime for both algorithms.

As shown in the table, our work completely fixes all antenna violations for all test cases except for "s38417," whereas

TABLE IV
COMPARISON WITH DIRMCF FOR THE 90% DIODE BLOCKAGE RATE

Circuit Name	L_{max} (μm)	Total # Viol.	DIRMCF [7] # Fixed Viol.	DIRMCF [7] Fixing Rate (%)	DIRMCF [7] # Diodes	DIRMCF [7] E. Wire Cost (μm)	DIRMCF [7] CPU Time (s)	Our Work # Fixed Viol.	Our Work Fixing Rate (%)	Our Work Jumper Cost	Our Work # Diodes	Our Work E. Wire Cost (μm)	Our Work Total Cost	Our Work CPU Time (s)
s5378	50	95	87	91.58	87	543.60	2.8	95	100	210	81	306.72	516.72	2.1
	100	49	48	97.96	48	266.40	2.2	49	100	60	46	80.64	140.64	2.7
s9234	50	56	52	92.86	52	560.16	2.1	56	100	195	45	290.16	485.16	1.4
	100	22	22	100	22	190.08	0.9	22	100	30	20	63.36	93.36	0.8
s13207	50	164	159	96.95	159	1271.52	33	164	100	465	134	511.20	976.20	28.5
	100	83	82	98.80	82	200.16	11.9	83	100	120	75	83.52	203.52	9.4
s15850	50	182	181	99.45	181	1450.80	76.5	182	100	390	156	617.76	1007.76	56.9
	100	98	98	100	98	175.68	29	98	100	90	92	63.36	153.36	20.8
s38417	50	406	381	93.84	381	4007.52	260.8	401	98.77	1320	316	1870.56	3190.56	265
	100	184	183	99.46	183	543.60	169.1	183	99.46	255	167	231.12	486.12	118.2
s38584	50	550	519	94.36	519	6348.96	320.2	550	100	2040	428	1968.48	4008.48	184.9
	100	283	281	99.29	281	1356.48	102.4	283	100	345	261	408.96	753.96	201.6
			Avg.	97.05				Avg.	99.85					

TABLE V
AVERAGE FIXING RATE COMPARISON WITH DIRMCF

Algorithms	Fixing Rate 80	Fixing Rate 85	Fixing Rate 90	Fixing Rate 95
DIRMCF [7]	98.85%	98.45%	97.05%	94.04%
Ours	99.98%	99.89%	99.85%	99.69%

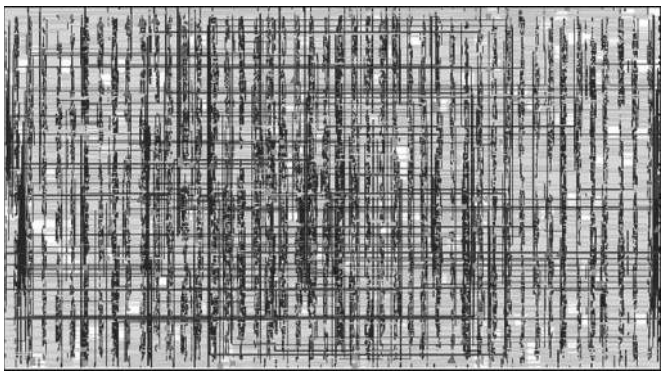

Fig. 12. Diode and jumper insertion result of our algorithm for "s15850." The inserted diodes and jumpers are highlighted by orange squares and pink triangles, respectively.

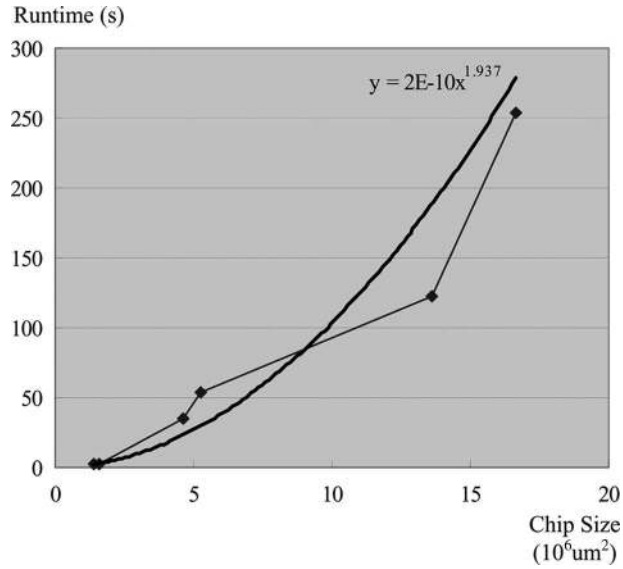

Fig. 13. Runtime is plotted as a function of chip sizes for the 80% blockage rate.

DIRMCF cannot for most cases. For those cases with the 100% fixing rate, our work always achieves lower fixing cost than DIRMCF. Table V summarizes the average fixing rates of DIRMCF and our work for 80%, 85%, 90%, and 95% diode blockage rates. Column "Fixing Rate 80" gives the average fixing rates with the 80% diode blockage rate, and so on. It is natural that the fixing rate of both works decreases as the diode blockage rate increases since less space is available for diode insertion. The results show that our work consistently achieves very high fixing rates at more than 99.69% even for 95% diode blockage rate, whereas the average fixing rate of DIRMCF decreases to 94.04% at the same blockage rate. Fig. 12 shows the diode and jumper insertion result of our algorithm for "s15850."

B. Empirical Runtime Analysis

Fig. 13 shows the empirical runtime trend of our program. First, the runtimes were derived from all test cases with the 80% blockage rate and the 50-μm antenna upper bound. Then, we applied the regression analysis based on the least-square method to derive the relationship between the runtimes and chip sizes. We found that the empirical time complexity of our program is about $O(n^{1.937})$ to the chip size n, which is much lower than the theoretical bound shown in Theorem 3.

C. Net Delay Impact

Table VI summarizes the delay penalty rates of antenna violating nets due to the antenna fixing. The delay penalty rate is computed by the increased delay over the original net delay. Columns 2, 3, and 4 give the maximum, minimum, and average delay penalty rates of all test cases with the 80% blockage rate and the 50-μm antenna upper bound, whereas columns 5, 6, and 7 give those with the 90% blockage rate. As shown in the table, fixing antenna violations by the SDJI algorithm resulted in only about 1% average delay penalty with the 80% blockage rates. If the blockage rate increases to 90%, the average delay penalty also increases to about 3% since the resources for antenna fixing are reduced. It should be noted that the minimum delay penalty of all test cases is zero. The reason is that, in most situations,

TABLE VI
DELAY PENALTIES FOR ANTENNA FIXING

Circuit Name	Delay Penalty 80			Delay Penalty Rate 90		
	Max.	Min.	Avg.	Max.	Min.	Avg.
s5378	21.15%	0%	1.10%	34.55%	0%	2.44%
s9234	13.06%	0%	0.93%	54.98%	0%	4.83%
s13207	34.32%	0%	1.10%	51.43%	0%	2.90%
s15850	30.91%	0%	1.13%	36.15%	0%	2.19%
s38417	44.32%	0%	1.06%	58.04%	0%	2.79%
s38584	69.58%	0%	1.60%	65.35%	0%	2.91%

TABLE VII
EXPERIMENT ON THE EFFECTIVENESS OF THE PRUNING TECHNIQUE

Circuit Name	w/o Pruning		w/ Pruning	
	CPU (s)	Mem (MB)	CPU (s)	Mem (MB)
s5378	88.4	491	2.2	31
s9234	58.2	406	2.0	27
s13207	144.6	2100	34.9	66
s15850	1072.6	2600	53.4	83
s38417	N/A	> 4096	123.1	207
s38584	N/A	> 4096	253.9	402
Avg.	1	1	8.8%	4.9%

the inserted diode or jumper might be located on a noncritical path, and thus, it will not increase the critical-path delay, or it has only an insignificant impact.

D. Effectiveness of the Pruning Technique

In Table VII, we compare the CPU times and memory usages of the SDJI algorithm without and with the pruning technique proposed in Theorem 4. The experimental data were obtained from all test cases with the 80% blockage rate and the 50-μm antenna upper bound. Note that the CPU times and memory usages for "s38417" and "s38584" without the pruning are not available since they need more than 4-GB memory, which exceeds the limitation of our 32-b machine. As shown in the table, pruning unnecessary nodes and edges in the flow network can reduce about 95% memory of the SDJI algorithm on average. Furthermore, due to the decrease of nodes and edges, the SDJI algorithm can also save about 91% CPU time on average to fix antenna violations. This experiment has shown that the proposed pruning technique is effective and necessary for solving the ASDJI problem.

V. CONCLUSION

We have proposed an optimal algorithm to solve the ASDJI problem. Our algorithm guarantees to find the optimal antenna fixing solution with diode/jumper insertion if such a solution exists. The experimental results have shown that our work achieves higher fixing rates and lower delay costs even for high-density circuits compared with the state-of-the-art previous works.

REFERENCES

[1] *The LEDA Package*. [Online]. Available: http://www.mpi-sb.mpg.de/LEDA/

[2] *LEF/DEF 5.5 language reference*.

[3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[4] Y.-W. Chang and S.-P. Lin, "MR: A new framework for multilevel full-chip routing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 5, pp. 793–800, May 2004.

[5] P. H. Chen, S. Malkani, C.-M. Peng, and J. Lin, "Fixing antenna problem by dynamic diode dropping and jumper insertion," in *Proc. IEEE Int. Symp. Quality Electron. Des.*, Mar. 2000, pp. 275–282.

[6] T.-Y. Ho, Y.-W. Chang, and S.-J. Chen, "Multilevel routing with antenna avoidance," in *Proc. ACM Int. Symp. Phys. Des.*, Apr. 2004, pp. 34–40.

[7] L.-D. Huang, X. Tang, H. Xiang, D. F. Wong, and I.-M. Liu, "A polynomial time optimal diode insertion/routing algorithm for fixing antenna problem," in *Proc. ACM/IEEE Des. Autom. Test Eur.*, Mar. 2002, pp. 470–475.

[8] L.-D. Huang, X. Tang, H. Xiang, D. F. Wong, and I.-M. Liu, "A polynomial time-optimal diode insertion/routing algorithm for fixing antenna problem," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 1, pp. 141–147, Jan. 2004.

[9] Z.-W. Jiang and Y.-W. Chang, "An optimal simultaneous diode/jumper insertion algorithm for antenna fixing," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2006, pp. 669–674.

[10] S.-P. Lin and Y.-W. Chang, "A novel framework for multilevel routing considering routability and performance," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2002, pp. 44–50.

[11] W. Maly, C. Ouyang, S. Ghosh, and S. Maturi, "Detection of an antenna effect in VLSI designs," in *Proc. IEEE Int. Defect Fault Tolerance VLSI Syst.*, Nov. 1996, pp. 86–94.

[12] H. Shirota, T. Sadakane, and M. Terai, "A new rip-up and reroute algorithm for very large scale gate arrays," in *Proc. IEEE Custom Integr. Circuit Conf.*, May 1996, pp. 171–174.

[13] H. Shirota, T. Sadakane, M. Terai, and K. Okazaki, "A new router for reducing 'antenna effect' in ASIC design," in *Proc. IEEE Custom Integr. Circuit Conf.*, May 1998, pp. 601–604.

[14] B.-Y. Su and Y.-W. Chang, "An exact jumper insertion algorithm for antenna effect avoidance/fixing," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jun. 2005, pp. 597–602.

[15] B.-Y. Su and Y.-W. Chang, "An optimal jumper-insertion algorithm for antenna avoidance/fixing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 10, pp. 1818–1829, Oct. 2007.

[16] B.-Y. Su, Y.-W. Chang, and J. Hu, "An optimal jumper insertion algorithm for antenna avoidance/fixing on general routing trees with obstacles," in *Proc. ACM Int. Symp. Phys. Des.*, Apr. 2006, pp. 56–63.

[17] B.-Y. Su, Y.-W. Chang, and J. Hu, "An exact jumper-insertion algorithm for antenna violation avoidance/fixing considering routing obstacles," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 4, pp. 719–734, Apr. 2007.

[18] D. Wu, J. Hu, and R. Mahapatra, "Antenna avoidance in layer assignment," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 4, pp. 734–738, Apr. 2006.

Zhe-Wei Jiang (S'05) received the B.S. degree in electronics engineering from the National Chiao Tung University, Hsinchu, Taiwan, in 2003. He is currently working toward the Ph.D. degree at the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan.

His current research interests focus on large-scale mixed-size placement and design for manufacturability.

Yao-Wen Chang (S'94–M'96) received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1988, and the M.S. and Ph.D. degrees from the University of Texas at Austin in 1993 and 1996, respectively, all in computer science.

He is a Professor in the Department of Electrical Engineering and the Graduate Institute of Electronics Engineering, National Taiwan University. He is currently also a Visiting Professor at Waseda University, Kitakyushu, Japan. He was with the IBM T. J. Watson Research Center, Yorktown Heights, NY, in the summer of 1994. From 1996 to 2001, he was on the faculty of National Chiao Tung University, Taiwan. His current research interests include VLSI physical design, design for manufacturability and reliability, design automation for biochips, and FPGA. He has been working closely with industry on projects in these areas. He has coauthored one book on routing and over 130 ACM/IEEE conference/journal papers in these areas.

Dr. Chang received an award at the 2006 ACM ISPD Placement Contest, Best Paper Awards at ICCD-95 and the 2007 VLSI Design/CAD Symposium, and 11 Best Paper Award Nominations from DAC (2000, 2005, 2007, 2008), ICCAD (2002, 2007), ISPD (two in 2007), ACM TODAES (2003), ASP-DAC (2004), and ICCD (2001). He has received many awards for research, such as the 2007 Distinguished Research Award, the inaugural 2005 First-Class Principal Investigator Award, and the 2004 Dr.Wu Ta You Memorial Award from the National Science Council of Taiwan, the 2004 MXIC Young Chair Professorship from the MXIC Corporation, and for excellent teaching from National Taiwan University (2004, 2006, 2007) and National Chiao Tung University (2000). He is currently an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS (TCAD) and an editor of the *Journal of Information Science and Engineering* (JISE). He currently serves on the ICCAD Executive Committee, the ACM/SIGDA Physical Design Technical Committee, and the ISPD Organizing Committee and has served on the technical program committees of ASP-DAC (topic chair), DAC, DATE, FPL, FPT (program co-chair), GLSVLSI, ICCAD, ICCD, IECON (topic chair), ISPD, SOCC (topic chair), TENCON, and VLSI-DAT (topic chair). He is currently an independent board director of Genesys Logic, Inc., a member of the board of governors of the Taiwan IC Design Society, and a member of the IEEE Circuits and Systems Society, ACM, and ACM/SIGDA.