

Vertical Via Design Techniques for Multi-layered P/G Networks*

Shuai Li, Jin Shi, Yici Cai, Xianlong Hong

Department of Computer Science and Technology, Tsinghua University, Beijing 100084, P.R. China
Email: {shuai-li05,shi-j03}@mails.tsinghua.edu.cn, {caiyc,hxl-dcs}@mail.tsinghua.edu.cn

Abstract - In multi-layered power/ground (P/G) networks, to connect the whole network together, vertical vias are usually placed at intersections between metal wires of adjoining layers. In this paper, a deep study about the design of vertical vias is presented. First we present an efficient heuristic algorithm based on sensitivity analysis to optimize via allocation in early design stage. Compared with even allocation, averagely our algorithm is capable of reducing worst voltage drop by 8.43% while using the same or even less number of vias. Also, adjoint network method is utilized and significantly improves the efficiency of our algorithm. Next, we demonstrate that by linking metal wires of nonadjacent layers, cross-layer vias are powerful in eliminating “hot” areas which suffer from large voltage drop on bottom layer. A similar heuristic algorithm is also developed for the addition of cross-layer vias.

I Introduction

As technology scales in VLSI chips, power distribution becomes a critical issue. The lower voltage supply allows a smaller margin of voltage drop on power/ground (P/G) network. At the same time, due to the increase in chip density, the currents drawn from P/G networks has increased greatly, while the resistances of interconnects has also been growing rapidly as a result of decreasing wire width. Thus, the IR drop in supply grid wires has become significant. Together with excessive inductive effects, this can lead to severe transistor performance degradation and logical failures. To overcome challenges mentioned above, a number of specialized techniques for the design and optimization of P/G networks are developed, including: 1) wire sizing [1-6]; 2) adding decoupling capacitors to keep dynamic fluctuations under control [7-10]; 3) topology optimization [11][12][13].

Among these techniques, wire sizing is one typical approach to solve the problem of excessive IR drop. In [1], an optimization engine for wire sizing is presented that first finds an initial feasible solution, and then iteratively looks for the optimal solution along a feasible direction using a sensitivity analysis method. [2] proposes another wire sizing algorithm based on the conjugate gradient method and circuit sensitivity analysis. Great progress is made in the algorithm in that only conductance is used as the variable and the adjoint network method is used to calculate the gradient. An optimization scheme for calculating both the lengths and widths of P/G networks using a sequential network simplex method is developed in [3]. In [4], wire widths are optimized by transforming the constrained nonlinear programming problem into a sequence of linear programs. Other heuristic

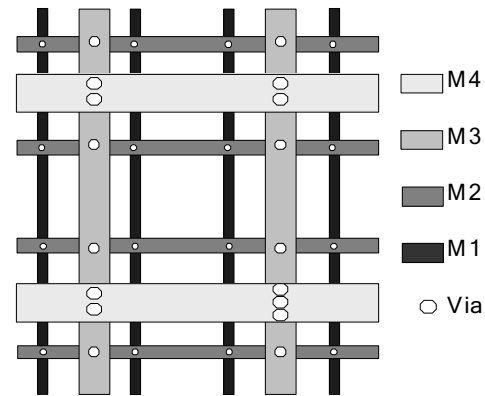


Fig. 1 An illustration of a four-layered P/G grid.

methods aiming to minimize wire area are also proposed in [5] and [6].

However, until now, little work has been done about vertical vias, which have an unneglectable influence on voltage drop, especially in multi-layered P/G networks. Fig.1 is an illustration of a four-layered power grid, where as usual, both the wire width and pitch of upper layers are larger than those of lower ones. Metal wires in adjoining layers are perpendicular to each other and usually vias are placed at intersections.

With technology scaling, the resistance of vias are also increasing rapidly and by now, in most cases, one via per intersection will result in significant voltage drop. Placing more vias at intersections is a typical solution for the problem. In the simplest way, all intersections between the same two adjoining layers could be allocated with the same number of vias. But definitely this is not the best idea for all circuits. And it remains an open problem how to allocate vias to intersections.

In this paper, we present two different design techniques for vertical vias.

Firstly, a heuristic algorithm for via allocation is proposed. In the algorithm, the sensitivity of objective function with respect to via number at every intersection is utilized to guide the optimization. Like in [2][7], adjoint network method is applied to calculate the sensitivity, as a result of which we can get sensitivities rapidly even in the circuit with up to 106929 nodes. According to experiment results, compared with equally allocating vias to all intersections, our method manages to reduce the worst voltage drop on the bottom layer by 8.43% averagely, with the same or even less number of vias.

In another point of view, as devices aiming to connect the

* This work supported by the National Natural Science Foundation of China No.60776026

P/G network together, vias should not be constrained to linking metal wires of adjacent layers only. The second technique proposed in our paper is just about linking metal wires of nonadjacent layers by means of cross-layer vias. In fact, in the case of mesh-like P/G network in Fig. 1, with little expense of wire congestion, cross-layer vias could be added to the network. And experiment results in IIIA shows that cross-layer vias are quite powerful in eliminating “hot” areas which suffer from large voltage drop on bottom layer. Furthermore, a heuristic algorithm is developed in IIIB for the addition of cross-layer vias. Also based on sensitivity analysis, the algorithm proves to be effective in determining where and how many cross-layer vias should be added to the network.

The rest of the paper is organized as follows. In Section II, the heuristic algorithm for via allocation is presented: part A presents the modeling for multi-layered P/G networks while part B formulates the problem; part C describes the solution techniques in detail and part D demonstrates the experiment results. Next, in Section III, we present our study on cross-layer vias: the advantage of cross-layer vias is demonstrated in part A and part B proposes the heuristic algorithm for the addition of cross-layer vias. Finally, we conclude the paper in Section IV.

II. Via Allocation

A. Modeling for Multi-layered P/G network with vias

For the similarity of power and ground grid, we will only consider power grid in this paper. In general, the power grid is modeled as an equivalent resistance circuit network G with n node $N=\{1, \dots, n\}$, as is shown in Fig. 2. Specifically, all intersections between metal wires of adjoining layers are extracted as nodes in the network. According to grid dimensions, metal wires are broken up into detailed metal segments, and the segments are modeled as resistances in the network. Vias at the same intersection are merged into one *Equivalent Via* (EV). Every EV is also modeled as a resistance and corresponds to one edge in G . The resistance of EV equals the resistance of one via divided by the number of vias in it. At last, given the worst case currents drawn by transistors, constant current sources to ground (not shown in the figure) are placed at leaf nodes of bottom layer.

We will use the following notations:

(i, j)	edge between nodes i and j ;
$g_{i,j}$	conductance of edge (i, j) ;
V_i	voltage of node i ;
VIO	set of violation nodes on bottom layer;
M_p	set of EVs between layers p and $p+1$;
vg_p	conductance of a via between layers p and $p+1$;
$vn_{a,b}$	number of vias at EV (a, b) ;

B. Problem Formulation

Generally, we are to optimize the number of vias at each intersection so that lower voltage degradation and less violation nodes could be achieved with less number of vias. Therefore, via allocation can be formulated into a constraint optimization problem:

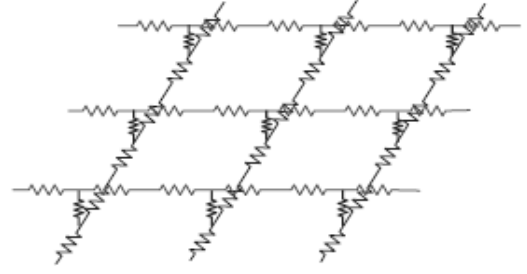


Fig. 2 Resistance network with vias merged into EV.

● Objective Function

In order to take all the violation nodes into consideration, the objective function is set to be:

$$S = \sum_{i \in VIO} (V_i - V_{min}) \quad (1)$$

where V_{min} is the constraint voltage by which we identify violation nodes on bottom layer. S is always nonpositive and larger S means lower power degradation.

● Vias Per Intersection Constraints

Between two certain adjoining layers, the number of vias placed at each intersection is restricted by wire widths and other factors:

$$vn_{a,b} \leq UpB_p \quad \forall (a,b) \in M_p \quad (2)$$

where UpB_p is the upper limit of the number of vias to be placed at each intersection between layers p and $p+1$.

From the objective function and the constraints, we can see the similarity between wire sizing and via allocation. Both are aiming to make full use of resources to lower IR drop. However, unlike wire sizing, the problem of via allocation is discrete. Techniques for constraint optimization problem in wire sizing are not applicable here. Hence, a heuristic algorithm is proposed in our paper for the solution of via allocation.

C. Solution Technique

1. Sensitivity calculation

In our algorithm, the sensitivity considering the number of vias at each EV, $\partial S / \partial vn_{a,b}$, is utilized to guide the optimization. Apparently, adding vias to EVs with larger sensitivity will result in a larger increase in the objective function and the sensitivity will be helpful in determining how many vias should be allocated to each EV.

The techniques for sensitivity calculation are as follows.

First, according to the definition of $vn_{a,b}$ and $g_{a,b}$, we get:

$$\frac{\partial g_{a,b}}{\partial vn_{a,b}} = vg_p \quad \forall (a,b) \in M_p \quad (3)$$

Then, using the chain rule, we can make further calculations as below:

$$\frac{\partial S}{\partial vn_{a,b}} = \frac{\partial S}{\partial g_{a,b}} \frac{\partial g_{a,b}}{\partial vn_{a,b}} = vg_p \frac{\partial S}{\partial g_{a,b}} \quad \forall (a,b) \in M_p \quad (4)$$

$$\frac{\partial S}{\partial g_{a,b}} = \sum_{i \in VIO} \frac{\partial V_i}{\partial g_{a,b}} \quad (5)$$

To get $\partial V_i / \partial g_{a,b}$, we can use the adjoint network method based on the Tellegen's method [15]. But using it directly would be very time-consuming, because for each violation node, an adjoint network must be built and simulated. Therefore, the merge adjoint network method proposed in [2] is adopted here.

Specifically, in the case of calculating $\partial S / \partial g_{a,b}$, which is simply the addition of $\partial V_i / \partial g_{a,b}$ of all the violation nodes, the method can be used in this way. First we construct an adjoint network G' with the same topology as the original network G , except that all voltage sources are shorted and current sources are open in the adjoint network. Next, different from the original adjoint network method in [15], a current source to ground valued $1A$ is added to each violation node. Then, using the technique proposed in [2], we get:

$$\frac{\partial S}{\partial g_{a,b}} = (V_a - V_b)(V'_a - V'_b) \quad (6)$$

where V'_a and V'_b are the voltage of nodes a and b in the adjoint network G' respectively.

By (4) and (6), we can obtain the sensitivity of all the EVs. Only two circuit simulations are necessary; one is for G and another for G' . Also, Cholesky decomposition conjugate gradient (ICCG) [14] is used as the solver for circuit simulation, which makes our algorithm even less time-consuming.

2. Algorithm flow

Fig.3 shows the flowchart of our algorithm.

At the beginning of the algorithm, all the v_n are set to be 1, i.e. only one via is placed at each intersection. Then, iteratively, vias are added to intersections, with sensitivity helping determine to which EVs vias are to be added.

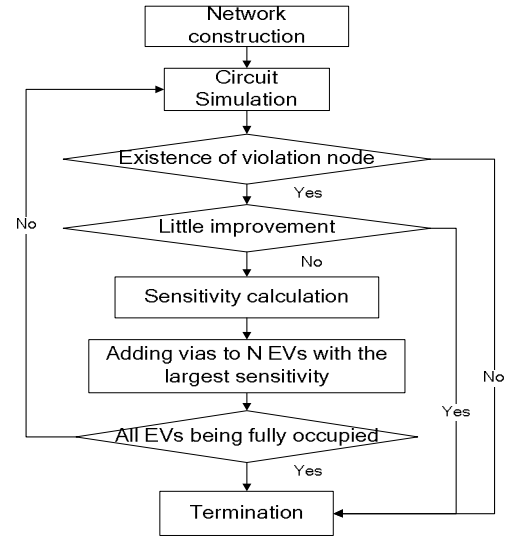


Fig. 3 Flowchart for the heuristic algorithm.

In every iteration, first we analyze the network for node voltage and identify the violation nodes on bottom layer; then, the adjoint network is built and the sensitivity of every EV is calculated; among all the EVs where addition of via will not violate the constraint, N EVs with the highest sensitivities are chosen and one via is added to each of them. The iteration will continue until all violation nodes are eliminated or no more vias can be added to intersections.

Generally speaking, with the addition of vias, the objective function will be increasing gradually. But since the number of violation nodes is becoming smaller, the sensitivity of EVs are decreasing; the improvement every added via contributes to the objective function is also weakening.

To make sure that the addition of vias is worthwhile, we set an *improvement threshold* α . When the improvement of every added via, the increase of the objective function divided by N , is less than α , the algorithm will stop as well.

TABLE I Experiment results of every circuit

Ckt	Num of Nodes	Num of EVs	N	Running Time (s)		Sum of vias	avg(VN)					MAX (V)	VIO NUM	AVG (V)
							7-6	6-5	5-4	4-3	3-2			
1	19078	2127	100	186.08	Opt	3627	4.79	3.06	3.71	2.04	1.13	0.0586	0	0.04223
					Comp1	5376	5	4	4	3	2	0.06378	89	0.0386
					Comp2	7503	6	5	5	4	3	0.06154	14	0.03687
					Comp3	9630	7	6	6	5	4	0.06014	1	0.03579
2	36562	4044	150	611.99	Opt	6294	3.91	2.64	3.00	1.86	1.11	0.06726	69	0.04458
					Comp	8872	4	3	3	2	2	0.07573	709	0.04025
3	58287	8394	200	957.75	Opt	12594	4.95	2.44	2.15	1.51	1.14	0.0652	177	0.04146
					Comp	18724	5	3	3	2	2	0.07035	756	0.03883
4	106929	11695	300	3021	Opt	17995	4.03	2.70	3.01	1.78	1.11	0.05998	0	0.03613
					Comp	26858	5	3	4	2	2	0.0646	446	0.0329

D. Experiment Results

The proposed algorithm has been implemented in C++. And a number of experiments have been performed on SUN UltraSparc workstation V880 with 750MHz CPU and 2GB memory.

Table I shows the experiment results of 4 circuits. All circuits are equipped with 7-layered power grid and the allocation of vias above layer 2 is optimized using our algorithm. Due to the unavailability of benchmark circuits for multi-layered P/G network, the circuits are randomly generated but with real circuit parameter values. The supply voltage is 1.2V, and the voltage constraint V_{min} is set to be 95% of 1.2V.

Table I also contains the comparison between optimized allocation and even allocation, in terms of the worst (MAX) and average (AVG) voltage drop on bottom layer, as well as the number of violation nodes (VIO NUM).

In the table, avg(VN) is the average of all the vn between two adjoining layers. In even allocation, all vn are set to be the ceiling integer of corresponding avg(VN) in optimized allocation. Thus, in even allocation, the sum of vias between any two adjoining layer is no less than that in optimized allocation. This explains why the AVG of even allocation is relatively higher. But apparently, in all circuits, our algorithm outperforms even allocation in MAX, by 8.43% averagely.

Also, we observe that the VIO NUM after optimized allocation are much less than even allocation, which means that our method is more effective in eliminating violation nodes. This is also evident in the comparison in Fig. 4, the bottom-layer voltage drop contours for Circuit 2. Furthermore, in Circuit 1, we also compare the optimized allocation with other cases of even allocation with more vias, as is shown in Table I. It is apparent that simply adding vias averagely to all intersections is not a good idea to lower the worst voltage drop. Finally, the results of our algorithm is dependent on the selection of N and α . In general, with less N, the program will become more time-consuming but we

TABLE II Comparison results with different N and same α

	N=10	N=20	N=50	N=200	N=300
Iteration Step	143	72	29	8	7
Running Time (s)	1713.99	774.72	336.74	106.12	92.46
Sum of vias	3557	3567	3577	3727	4227

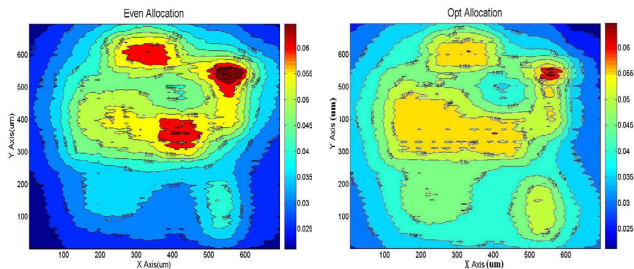


Fig. 4 Contours of voltage drop (V) on bottom layer for Circuit 2

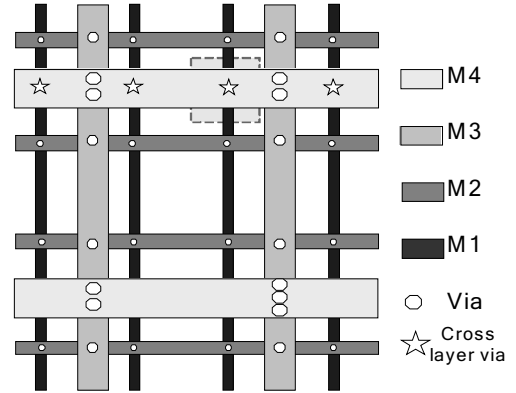


Fig. 5 The illusion of the same four-layered P/G grid, with cross-layer vias placed at intersections between metal wires of layer 4 and layer 1.

can get sounder results. Table II summarizes the comparison results with different N and same α when Circuit 1 is optimized.

III. Cross-layer Vias

A. Advantage of cross-layer vias

By now, only vias placed at intersections between adjacent layers are discussed in the paper. However, as is shown in Fig.1, perpendicular intersections also exist between metal wires of layer 1 and layer 4. At these intersections, the linkage between metal wires of the two layers could also be realized by three vias connected head to end. To differentiate it from ordinary vias, this kind of combination of vias is called *cross-layer via* in our paper, and it is marked as a star in Fig.5. Cross-layer vias only have a little influence on the wire congestion of middle layers. In some cases, they may also conflict with metal wires in middle layers, but the problem could be solved by slightly changing via position and wire winding, the impact of which is almost negligible compared with resistance of vias. Thus, in our model, cross-layer vias are simply abstracted as ordinary vias in series.

In multi-layered P/G network, IR drop is composed of voltage drop on vertical vias and voltage drop on metal wires. But apparently, the latter part does not exist on cross-layer vias. Thus, in some sense, a cross-layer via could be considered as a “short cut” for currents on the network and supposedly, it will help decrease the voltage drop of its nearby area. As is shown in Fig.5, the area enclosed by the dotted rectangle may benefit from the cross-layer via.

Our supposition is confirmed in experiments. One of our experiments is implemented on a 7-layered power grid. The contour of voltage drop on bottom layer for the original circuit is shown in Fig.6(a). One “hot” area which suffers from large voltage drop is enclosed by dotted rectangle in the figure. The worst voltage drop in the area is up to 0.08179V, which is also the worst voltage drop of the whole circuit. In the rectangle, there are in all 12 intersections between metal wires of layers 7 and 4. In our experiment, cross-layer vias are added to the 12 intersections, with each intersection allocated equal number of cross-layer vias.

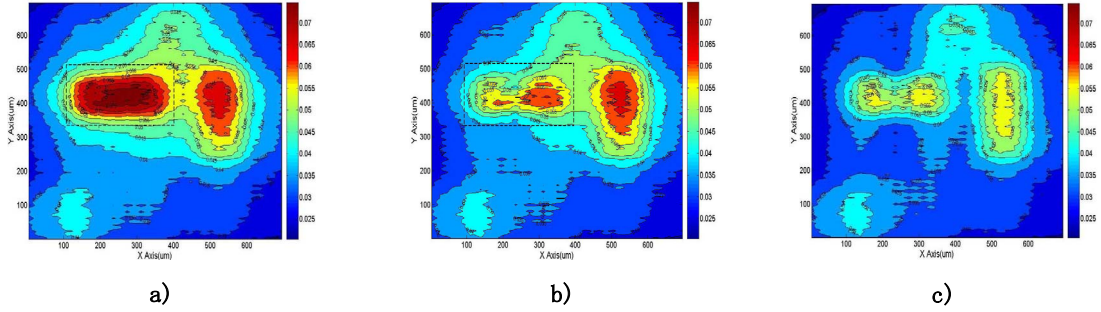


Fig. 6 Contours of voltage drop (V) on bottom layer for another circuit: a) original; b) after 72 cross-layer vias linking metal wires of layer 7 and layer 4 are added, with 6 added to each intersection in the dotted rectangle; c) after the sensitivity-based algorithm described in IIIB is applied to the circuit

Table III demonstrates the experiment results after different number of cross-layer vias are added. We observe that gradually, both AVG and MAX in the area decrease with the addition of cross-layer vias. By comparing MAX in the area with MAX of the circuit, we find that after 48 cross-layer vias are added above it, MAX in the area is no longer equal to MAX of the circuit. The “hot” area in the rectangle is no longer the “hottest” on the chip. We can also see the elimination of the “hot” area in Fig.6(b), the voltage drop contour of the same circuit after the addition of 72 cross-layer vias. According to Table III, the number of violation nodes on the chip is reduced from 1466 to 690 by the 72 cross-layer vias.

B. Heuristic algorithm for the addition of cross-layer vias

In the example in IIIA, equal number of cross-layer vias is added to each intersection in a fixed area. This method is not applicable to all circuits and may result in a waste of resources. Besides, in the example, only cross-layer vias between layers 7 and 4 are used, while we can also link metal wires of other nonadjacent layers. A wiser approach is needed to determine where, how many and what kind of cross-layer vias should be added to the network.

To solve the problem about the addition of cross-layer vias, a sensitivity-based heuristic algorithm is developed. The algorithm is similar with the algorithm for via allocation.

In the algorithm, the objective function is the same as (1) and constraints also exist as for the number of cross-layer vias at each intersection. A resistance network G^+ similar with Fig.2 is constructed for the circuit. In the network, besides intersections between metal wires of adjacent layers, candidate intersections between metal wires of nonadjacent

TABLE III Experiment results after different number of cross-layer vias is added

Num of cross layer vias	AVG in hot area (V)	MAX in hot area (V)	MAX (V)	VIO NUM
0	0.07234	0.08179	0.08179	1466
12	0.06416	0.07286	0.07286	1108
24	0.06114	0.06993	0.06993	945
48	0.05849	0.06790	0.06909	765
72	0.05763	0.06695	0.06903	690

layers are also extracted as nodes. Thus, like ordinary vias, cross-layer vias could also be modeled as resistances in G^+ .

In our algorithm, however, the fact is that at every candidate intersection place for cross-layer vias, one edge is added in G^+ . Like EV, the edge is defined as *virtual equivalent via* (VEV). VEV is virtual because at the beginning of the algorithm it contains no via. Its conductance is also 0 at the beginning but will change once cross-layer vias are added to it. The following notations related to VEV are used:

- $M_{p,q}$ set of VEVs between layers p and q , with $p > q$;
- $vg_{p,q}$ conductance of one cross-layer via between layers p and q , with $p > q$;
- $vn_{c,d}$ number of ordinary vias in VEV (c, d);
- $vvn_{c,d}$ number of cross-layer vias in VEV (c, d);

And the following relation exists:

$$vn_{c,d} = (p - q)vvn_{c,d} \quad \forall (c, d) \in M_{p,q} \quad (7)$$

The sensitivity of objective function with respect to all VEVs, $\partial S / \partial vn$, is also helpful in determining the identity of VEV where the addition of cross-layer vias would have the most favorable impact on objective function. In fact, according to (7) and the chain rule, easily we can obtain:

$$\frac{\partial S}{\partial vvn_{c,d}} = \frac{\partial S}{\partial g_{c,d}} \frac{\partial g_{c,d}}{\partial vvn_{c,d}} \frac{\partial vvn_{c,d}}{\partial vn_{c,d}} = \frac{vg_{p,q}}{(p - q)} \frac{\partial S}{\partial g_{c,d}} \quad (8)$$

where $vg_{p,q}$ could be calculated given the conductance of ordinary vias it contains and $\partial S / \partial g_{c,d}$ can be got by the same method described in IIC.

With the guide of sensitivity, cross-layer vias can be added to the network iteratively. In each iteration, after identifying violation nodes and calculating sensitivity of all NEVs, N NEVs with the largest sensitivity are chosen and one cross-layer via is added to each of them. The addition will continue until no violation node exists or added vias have little impact on the objective function. The flow of the algorithm is similar with Fig.3.

The algorithm has also been implemented in C++ and two experiments have been performed. In experiments, the upper limit of cross-layer vias at one intersection between layers p and q , $UpB_{p,q}$, is set to be $\min(UpB_q, UpB_{q+1}, \dots, UpB_p)$.

First, cross-layer vias are added to the original circuits in Table I. Two kinds of cross-layer vias, $M_{7,4}$, $M_{6,3}$, are available. Both are composed of 3 ordinary vias in series. For the same circuit, the improvement threshold α is the

TABLE IV Experiment results of the heuristic algorithm for adding cross-layer vias

Ckt	Num of VEV	N	Time (s)	Ordinary vias added	MAX (V)	VIO NUM
1	359	10	150.46	240	0.0587	0
2	672	20	340.77	540	0.0576	0
3	1369	30	929.96	1080	0.062	27
4	1899	40	607.54	360	0.05948	0

same as before. The results are demonstrated in Table IV. Apparently, in eliminating violation nodes, adding cross-layer vias is more effective than simply adding vias to intersections of adjacent layers. In Circuit 1, for example, all violation nodes are eliminated after merely 80 cross-layer vias, or 240 ordinary vias, are added, while in Table I, 3627 vias are needed even in the optimized allocation.

The second experiment is implemented on the circuit mentioned in IIIA and also, only cross-layer vias between layers 7 and 4 are available. It turns out that with the heuristic algorithm, only 85 cross-layer vias are enough to eliminate all the violation nodes on the chip, as is shown in Fig. 6(c).

IV. Conclusion

In conclusion, we have presented the first deep study on the design of vertical vias in multi-layered P/G networks. We first proposed an efficient heuristic method for the problem of via allocation. In the method, sensitivity analysis considering the number of vias at each intersection is utilized to guide the optimization. This proves to be effective and our algorithm outperforms even allocation in both the worst voltage drop and the number of violation nodes. Furthermore, due to the adoption of the adjoint network method and ICCG, sensitivity calculation is much less time-consuming and this greatly improves the efficiency of our method.

In the study of cross-layer vias, we showed that cross-layer vias are powerful in eliminating violation nodes on bottom layer and a sensitivity based heuristic algorithm was also proposed for the addition of cross-layer vias.

References

- [1] R. Dutta and M. Marck-Sadowska, "Automatic sizing of Power Ground (P/G) networks in VLSI," *Proc. CM/IEEE Design Automation Conf.*, pp. 783–786, 1989.
- [2] X. Wu, X. Hon, Y. Ca, C. K. Cheng, J. Gu, and W. Dai, "Area minimization of power distribution network using efficient nonlinear programming techniques," *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, pp. 153–157, 2001.
- [3] T. Wang and C. C. Chen, "Optimization of the power/ground network wire-sizing and spacing based on sequential network simplex algorithm," *Proc. IEEE Int. Symp. Quality Electron. Design*, pp. 157–162, 2002.
- [4] X. D. S. Tan, C. J. R. Shi, and J. C. Lee, "Reliability-constrained area optimization of VLSI power/ground networks via sequence of linear programmings," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst. Conf.*, vol. 22, pp. 156–161, 2003.
- [5] S. Boyd, L. Vandenberghe, A. E. Gamal, and S. Yun, "Design of robust global power and ground networks," *Proc. ACM Int. Symp. Phys. Design*, pp. 60–65, 2001.
- [6] X. Wu, C. Qiao, X. Hong, "Design and Optimization of Power/Ground Network for Cell-Based VLSIs with Macro Cells," *Proc. of ASP-DAC'99, HongKong*, pp. 21–24, 1999.
- [7] J. Fu, Z., X. Hong, Y. Cai, S. X. D. Tan, Z. Pan, "A Fast Decoupling Capacitor Budgeting Algorithm for Robust On-Chip Power Delivery," *Proc. ASPDAC*, pp. 505–510, 2004.
- [8] S.-Y. Zhao, K.K. Roy, C.-K. Koh. "Decoupling capacitance allocation for power supply noise suppression." *Proc. IEEE/ACM International Symp. on Physical Design*, pp.66-71,2001.
- [9] H.-H. Su, K.K. Roy, S. S. Sapatnekar. S. R. Nassif. "An algorithm for optimal decoupling capacitor sizing and placement for standard cell layouts." *Proc. IEEE/ACM International Symp. on Physical Design*, pp.68-75, 2002.
- [10] J. Fu, Z. Luo, X. Hong, Y. Cai, S. X.-D. Tan, Z. Pan, "VLSI On-Chip Power/Ground Network Optimization Considering Decap Leakage Currents", *Proc. ASPDAC*, p735-738, 2005.
- [11] T. Mitsuhashi and E. S. Kuh, "Power and ground network topology optimization for cell based VLSIs," *Proc. ACM/IEEE Design Automation Conf.*, pp. 524–529., 1992.
- [12] J. Oh and M. Pedram, "Multi-pad power/ground network design for uniform distribution of ground bounce," *Proc. ACM/IEEE Design Automation Conf.*, pp. 287–290, 1998.
- [13] J. Singh and S. Sapatnekar, "Congestion-aware topology optimization of structured power/ground networks," *IEEE TCAD*, pp. 683-695, 2005.
- [14] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1983.
- [15] L. T. Pillage, R. A. Rohrer, and C. Visweswariah, *Electronic Circuit and System Simulation Methods*. New York: McGraw-Hill, 1995.