

Congestion-driven Codelign of Power and Signal Networks *

Haihua Su [†]

Jiang Hu [†]

Sachin S. Sapatnekar *

Sani R. Nassif [†]

[†] IBM Corp.

11501 Burnet Rd.
Austin, TX 78758

{haihua,jianghu,nassif}@us.ibm.com

* ECE Dept, Univ. of Minnesota

200 Union St. SE
Minneapolis, MN 55455

sachin@ece.umn.edu

ABSTRACT

We present a global wire design methodology that simultaneously considers the performance needs for both signal lines and power grids under congestion considerations. An iterative procedure is employed in which the global routing is performed according to a congestion map that includes the resource utilization of the power grid, followed by a step in which the power grid is adjusted to relax the congestion in crowded regions. This adjustment is in the form of wire removal in noncritical regions, followed by a wire sizing step that overcomes the effects of wire removal. Experimental results show that the overall routability can be significantly improved while the power grid noise is maintained within the voltage droop constraint.

Categories and Subject Descriptors

B.8.2 [Performance and Reliability]: Performance Analysis and Design Aids

General Terms

Algorithms

Keywords

wire congestion, codelign, signal routing, power grid noise

1. INTRODUCTION

The role of interconnect has become increasingly critical in nanometer design and the need to meet stringent performance constraints has resulted in strong contention for scant routing resources. A major consumer of these resources is the power distribution network, which contains dense grids. On the other hand, global wires also compete for the same

routing resources, as they often require shortest-path routes to meet their own performance requirements. Traditionally, these two have been designed independently, with the routing needs for a regular power grid being determined first, after which the remaining resources are calculated to provide routing resource budgets for the signal nets.

As the number and criticality of global signal wires becomes more dominant, such a methodology becomes unsustainable as the initial budgets may often be entirely unreasonable. Therefore, in nanometer design, there is a strong need for a unified approach to the design of signal wires and power grids, with an integrated approach to routing resource management.

While it is convenient to build a regular power grid with a constant pitch (defined as the distance between adjacent wires in the grid), some degrees of freedom exist and it is desirable that they be exploited. For instance, in regions where the demand for routing resources from signal nets is high, a sparser power grid may be used as long as the performance constraints on the supply and ground lines can be met; likewise, signal nets are well advised to avoid the hot spots of the chip if possible, since these may need a locally dense power grid.

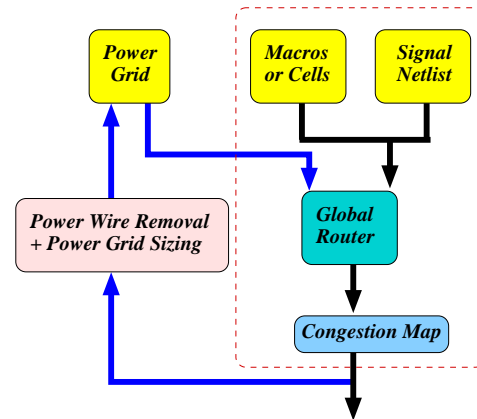


Figure 1: Congestion-driven power grid design and global routing.

The idea of managing wire congestion in signal routing has long been a significant objective in global routing. Various congestion-driven techniques include sequential routing (e.g., [6]), rip-up-and-reroute (e.g., [20]), and multicommod-

*This work was supported in part by the NSF under award CCR-0098117 and by the SRC under contract 99-TJ-714.

ity flow based methods (e.g., [16]) have been proposed. Most of these techniques aim at solving the problem solely at the routing stage, assuming that the total routing resources are fixed. Recent publications [1, 11] have presented techniques for simultaneous global routing and resource allocation under performance constraints.

Power grid optimization techniques have also been studied in [17, 19, 21]. All of these aim at minimizing total power wire area subject to the voltage droop and/or electromigration constraints, formulating the problem as a nonlinear program. The work in [12] presents a technique for shield insertion in a predesigned power grid to control inductive effects.

To the best of our knowledge, no published work performs a concurrent optimization of the power grid along with signal wires under routing congestion constraints, and this is the subject of the work presented in this paper. Our proposed congestion-driven flow is illustrated in Fig. 1. The dashed rectangle corresponds to a more conventional global routing flow. In essence, our approach presents a new flow that adds a feedback loop that permits the readjustment of the signal routing budgets by altering the power grid appropriately. Our approach incorporates a tight coupling between power grid adjustments and the routing of signal wires to exploit the altered congestions that result from these adjustments, and aims to solve problems with severe congestion constraints where conventional techniques are inadequate.

2. POWER GRID-AWARE SIGNAL ROUTING

As in global routing, we tessellate the entire chip into an array of grid cells, as shown in Fig. 2(a), and use the wiring information across the boundaries between neighboring grid cells to estimate the signal wire congestion distributions. We denote the width, in μm , of a boundary b between two neighboring grid cells as $W(b)$. This width represents the limited resources that must be shared on each layer by the supply lines and the signal lines that traverse the boundary, as shown in Fig. 2(b). In other words, the number of wires crossing b is inherently limited by the width $W(b)$, and $W(b)$ may partly or wholly be occupied by the crossing wires.

We represent the total width occupied by power grid wires on boundary b as $P(b)$. If a power grid wire p_i has a track width of $w(p_i)$, which includes its wire width and the required spacing from an adjacent wire, and there are a set of such wires, p_1, p_2, \dots, p_m , that cross b , then $P(b) = \sum_{i=1}^m w(p_i)$ represents the space that is unavailable for signal wires to cross the boundary. Therefore, we subtract this quantity from the boundary width to obtain the space available for signal wires as $W(b) - P(b)$. Typically, a uniform track width \bar{w} is applied to all the signal wires in the congestion estimation or global routing stage. Hence, signal wire congestion is often expressed in terms of the number of wiring tracks and the number of tracks available for signal wires is $T(b) = \lfloor \frac{W(b)-P(b)}{\bar{w}} \rfloor$. If there are $S(b)$ signal wires that cross a boundary b , then the overflow on b is $\max(0, S(b) - T(b))$. All tile boundaries with positive overflow values form a congestion map for a chip. The wire density at b is represented as $S(b)/T(b)$, measuring the congestion of b . A common objective for global routing is to ensure that there is no boundary with the wire density greater than one, i.e., $S(b) \leq T(b)$.

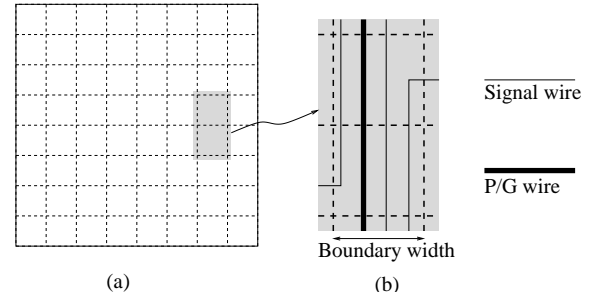


Figure 2: Wire congestion estimation based on tessellation on a chip.

In this work, the topology selection for the power grid is tightly coupled with the requirements of global routing for signal wires, and it is important to estimate the congestion contribution of the signal wires by implementing a fast global routing procedure. The technique used for this purpose is described in the remainder of this section, but it may be noted that this procedure may be replaced by any other computationally efficient method that shares a similar goal.

At the beginning of the algorithm, we perform a coarse global routing of all signal nets to obtain an estimate of the distribution of signal wire congestion at various boundaries, and feed this information to power/ground optimizer. The global routing technique used here is similar to [1], where a Steiner tree is initially constructed for each signal net using the AHHK algorithm [2] without considering congestion. After the initial Steiner trees have been constructed, an iterative rip-up-and-reroute procedure is applied to further reduce the wire congestion. The outer loop, consisting of signal routing followed by power grid optimization, is repeated iteratively until the constraints are satisfied or no further improvement is possible. In each iteration, the global routing solution from the previous iteration is used as a starting point for the rip-up-and-reroute procedure, with updated congestion values being used to direct the routing.

The rip-up-and-reroute procedure processes each net sequentially using the fixed net ordering heuristic proposed in [13], continuing until the maximum wire density is no greater than one, or after two complete iterations, since the congestion reduction after the second iteration is limited. Each net undergoing rip-up-and-reroute is entirely deleted and then rerouted using an algorithm similar to the min-max tree algorithm in [6].

3. POWER SUPPLY NOISE ANALYSIS

As is usually done in power grid analysis work [4, 5], we use the following linear circuit model to analyze the voltage droop noise of the power distribution network:

- The power grid is modeled as a resistive mesh.
- The cells/blocks are modeled as time-varying current sources connected between the power and ground plane.
- Decoupling capacitors are modeled as single lumped capacitors connected between power and ground.
- The top-level metal is connected to a package modeled as an inductance connected to an ideal constant voltage source.

This model leads to a large-scale linear circuit for power grid analysis, which can be efficiently analyzed using the technique described in [14, 18]. The wire width optimization procedure for the supply network, described in Section 4.2, also requires the computation of gradients, and this is performed using the simulation framework. Specifically, the transient adjoint sensitivity analysis technique is applied to calculate the sensitivity of the noise metric with respect to every tuning parameter, which, in our case, is the width of every power wire.

The noise and sensitivity analysis techniques used here are substantially similar to our work on decoupling capacitor (decap) placement [18] and have therefore been described only very briefly here. The only difference is that the tuning parameters in decap placement are the decap values, while in this work, the tuning parameters correspond to the width of each power grid wire. Consequently, current waveforms instead of voltage waveforms must be stored for adjoint sensitivity computations in this work.

4. POWER GRID DESIGN SCHEME

Starting with a dense grid that is guaranteed to meet the constraints on the supply network, our scheme iteratively sparsifies the grid to ease the wire congestion. In each iteration of the loop in Fig. 1, the power grid is adjusted using a two-step technique:

- In the first step, a wire removal heuristic is used to make the grid less dense in some regions, with due consideration paid to both the congestion information and the power grid noise.
- This is followed by a wire sizing step that readjusts the sizes of wires in the grid to compensate for the loss of this wire.

Therefore, our procedure ensures satisfactory performance of the power grid while utilizing just enough routing resources, so that the resources available to signal routing are maximized.

4.1 Power grid wire removal heuristic

As stated in Section 2, the congestion is measured in proportion to the overflow value on each tile edge. All power wires that lie in congested regions are potential candidates for removal, except those that lie within hot spots where the voltage droop is significant. The rationale behind this is that whenever a power wire is removed, the performance of the overall grid is compromised, and this is all the more noticeable if this wire lies within a hot spot. Therefore, we define a power wire as *critical* if the worst-case voltage droop on it is beyond some specified threshold VDP_{th} . Critical wires are not candidates for removal even if they lie in congested regions.

In addition, we define the *criticality* (Crt) of each non-critical power wire as the reciprocal of the RMS distance of that wire to critical nodes with nonzero noise within its neighborhood. Since the *criticality* of a power wire is only defined for non-critical wires, all these distances have non-zero values. For example, when a vertical non-critical power wire crosses a horizontal tile boundary located between $[x_L, x_R]$, the criticality of this wire to its closest noisy

region is

$$Crt_p = \frac{K_c}{\sqrt{\sum_{i \text{ critical and } |x_p - x_i| \leq \Delta} (x_p - x_i)^2}} \quad (1)$$

where x_p is the x-coordinate of the vertical power wire, x_i is the average of the two boundary terminals x_L and x_R , K_c is the total number of critical nodes within some distance Δ to the tile center and x_i is the x-coordinate of the critical node i . In our experiments, we take Δ as 1.5 to 2 tile edge lengths. It is clear to see that the larger the criticality of a power wire is, the closer this wire is to one of the hot spot region. Given several candidate power wires across one tile boundary with its overflow value larger than OV_{th} , the criticality of each wire determines an optimal order of removal for these wires.

The power wire removal heuristic proceeds as follows. First, the tile boundaries are sorted in decreasing order of their overflow values and all non-critical power wires are sorted according to their criticality values. Next, non-critical power wires are removed in the order of the sorted tile boundaries, provided the overflow value at the boundary is larger than OV_{th} . Several practical considerations must be taken into account. Firstly, the power wire removal process should be accompanied by dynamically updating the overflow value on every tile edge, so that subsequent wire removal is based upon the updated congestion information. Secondly, a reasonable number of power wires must be removed in each iteration, and this depends on the values of OV_{th} and VDP_{th} . Since the choice of these numbers is necessarily empirical and cannot be entirely relied on, we assert an upper bound for the number of power wires to be removed in any iteration to conserve the amount of computation required in the wire sizing step discussed in Section 4.2. In our experiments, this upper bound is chosen to be 6% of the total number of power wires. If a set of non-critical power wires are across one tile boundary, remove them in their increasing order of criticality. Through this rule, non-critical power wires with less criticality are removed first.

4.2 Power grid sizing

The second step in the adjustment of the power grid is related to sizing the wires in the grid to compensate for the increased voltage droop after the wire removal step. The problem is directly formulated as a nonlinearly constrained nonlinear programming problem as follows:

$$\begin{aligned} & \text{minimize} && Area(w_j) = \sum_{j=1}^{N_{wire}} l_j \times w_j \\ & \text{subject to} && w_{min} \leq w_j \leq w_{max}, j = 1 \cdots N_{wire} \\ & && \text{and} && Z(w_j) < \epsilon \end{aligned} \quad (2)$$

where ϵ is a very small number, and N_{wire} is the total number of wires in the grid.

The objective function that minimizes the total power wire area is consistent with the goal of congestion reduction. The first constraint restricts every power wire width to lie within a realistic range that is technology dependent. The second constraint requires the definition of the parameter Z , which is an effective metric for noise measurement that was proposed in [7]. This idea for one node is illustrated in Fig. 3, which shows the voltage waveform of one node on the V_{dd} grid. Z is the sum of each individual nonzero noise z_j . This metric finds the integral of the noise violation

and is zero if all constraints are satisfied. It punishes larger violations more severely than minor violations, and since it incorporates both the magnitude and time axes together, it is observed to be more practical than one that considers only the worst-case noise violation.

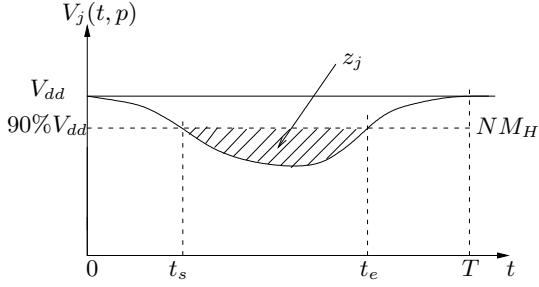


Figure 3: Illustration of the voltage droop at a given node in the V_{dd} power grid. The area of the shaded region corresponds to the integral z at that node.

The nonlinear constraint function Z can be obtained by transient analysis of the power grid circuit, and its sensitivity with respect to all the variables w_j can be calculated using the adjoint method discussed in [7, 18].

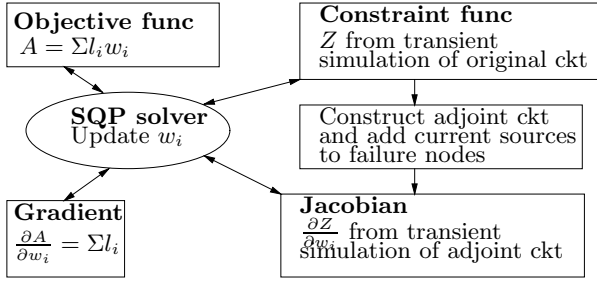


Figure 4: Power grid sizing procedure.

We use a standard Sequential Quadratic Programming (SQP) solver [3, 10] to solve the optimization problem. This solver requires users to provide subroutines to evaluate the objective and constraint functions and their derivatives with respect to each decision variable. The evaluations that are required for the SQP solver are illustrated in Fig. 4.

Practically, it is observed that the SQP solver converges slowly near the optimum. However, an approximate convergence is sufficiently good for our purposes, so that we require Z to be near-zero rather than exactly zero by choosing a sufficiently small ϵ .

5. OVERALL FLOW OF THE ALGORITHM

Having described all of the individual pieces of the algorithm, we now outline the overall flow of the procedure. The starting point is a given tessellation of a chip and a given initial power grid construction, and the sequence of steps can be summarized as follows:

- 1) An initial power-grid aware global routing step is carried out to route the signal nets. Each net is first routed without considering congestion, followed by an

iterative rip-up-and-reroute, described in Section 2, using the updated congestion information.

- 2) Based on the routes used by the power grid and the signal routes, a congestion map is generated and the overflow on each tile boundary is calculated. All tile boundaries whose overflow value exceeds the threshold OV_{th} are identified and are sorted in decreasing order of this value.
- 3) A transient simulation of the power grid is performed to identify critical power wires on which the worst-case voltage droop is above the threshold, VDP_{th} .
- 4) Sort the non-critical power wires in increasing order of the criticality.
- 5) Based on the congestion map generated in Step 2, non-critical power wires are removed according to the heuristic described in Section 4.1.
- 6) To compensate for the removal of these wires, the remaining power grid wires are resized using the nonlinear optimization procedure described in Section 4.2.
- 7) The congestion maps are updated, and the global routing is updated by performing rip-up-and-reroute based on the new congestions. At this point, the iterative loop is invoked so that the updated power grid and congestion map are fed back to step 3. The stopping criterion for the iteration is that the maximum overflow should be no greater than 1, or that no further improvement is possible. The latter is easily identified if it is detected that the changes in the congestion map after rip-up-and-reroute are insignificant, and that no further deletion of the power wires is possible.

The initial routing takes $O(MN^3)$ time, where N is the number of pins for each net and M is the total number of nets. The rip-up-and-reroute has a complexity of $E \log \log E$, where E in our case is the total number of tile edges. The complexity of power wire removal is $O(K \log K + P \log P + KPE)$, where K is the total number of tile edges with negative overflow values, P is total number of power wires and E is total number of tile edges. The first term stands for the complexity of sorting these tile edges, the second term is the worst-case complexity of sorting the criticality of non-critical power wires, and the third term comes from the power wire removal and dynamic tile edge overflow update process. The power grid analysis and sensitivity analysis has the complexity of one LU decomposition and two forward/backward substitutions. In practice, the cost of these computations is just over $O(n)$ for a sparse positive definite matrix, where n is the total number of nodes and inductance branches. For a nonlinear optimization problem with w decision variables, advanced implementations of the SQP solver have $O(w^2)$ cost. Therefore the worst-case complexity for our wire sizing procedure ends up to be $O(I(n + w^2))$, where I is the total number of iterations. The efficiency of the gradient-based SQP solver relies largely on the initial solution and its distance away from an optimum. Practically, it is seen that the optimal solution can be reached in a limited number of iterations.

6. EXPERIMENTAL RESULTS

The power grid analyzer and the global router have both been implemented in C++. The power grid removal and sizing scheme, and the overall congestion-driven power grid design and global routing flow has been written using Tcl. The wire size optimization is performed using an off-the-shelf SQP solver [10]. All experiments are performed on an Intel Pentium-IV 1.8GHz machine with 256M memory running Redhat Linux 7.0. The entire procedure is encapsulated in a flow called PSiCo (**P**ower-**S**ignal **C**ode**s**ign).

Circuit	B	N	Tile size
apte	9	77	31x36
ami33	33	112	30x28
ami49	49	368	33x34
playout	62	1294	30x26
ac3	27	200	38x36
hc7	77	430	34x39
a9c3	147	1148	33x31

Table 1: Test circuit parameters.

We have tested our flow on seven benchmark circuits obtained from the authors of [1]. All designs correspond to a $0.18\mu\text{m}$ technology and a supply voltage of 1.8V. The time-varying current sources modeling the behavior of each functional block was not originally available in these benchmark circuits. These waveforms are constructed by modifying current waveforms from several industrial circuits by adjusting their magnitude according to the area of each block. However, this is not a critical assumption since our method is applicable to the exact waveforms where available.

Circuit	Method	D_{max}	Overflow
apte	Traditional	2.50	27
	PSiCo	1.00	1
ami33	Traditional	3.00	122
	PSiCo	1.00	0
ami49	Traditional	2.14	192
	PSiCo	1.00	0
playout	Traditional	1.94	158
	PSiCo	1.05	2
ac3	Traditional	2.50	316
	PSiCo	1.00	0
hc7	Traditional	2.13	518
	PSiCo	1.00	0
a9c3	Traditional	1.33	229
	PSiCo	1.00	0

Table 2: A comparison of the congestion improvement after global routing using the conventional approach and our new method.

Initially, six layers of regularly distributed power grids with fixed wire widths are generated for all of these examples, with each layer containing only horizontal or vertical wires. Since a majority of the global routes are typically seen on the third and fourth metal layers, M3 and M4, we perform the global routing and power grid sizing on these two layers, assuming that the other layers are processed separately. The wire widths on these layers are assumed to be constrained within the range of $0.8\mu\text{m}$ and $4\mu\text{m}$ in our experiments. The initial power grid is constructed with a constant pitch in M3 and M4 such that under the given set of time-varying current sources that represent each block, the voltage droop on the entire power grid is within a threshold,

which in our experiments, is chosen to be 0.18V. Table 1 lists the characteristics of each circuit, in terms of the total number of blocks B and nets N and the tile sizes.

The performance of PSiCo can be described in terms of two components: the global routing solution, and the performance of the final power grid, and these results are shown in Tables 2 and 3, respectively. The wire congestion results of PSiCo are compared in Table 2 against the results of the traditional rip-up-and-reroute method, which corresponds to the result at the end of Step 1 in Section 5. For each circuit, the maximum density D_{max} , calculated as the maximum ratio of the utilization to the capacity across any tile boundary. Also shown is the overflow, i.e., the total amount by which the tile boundary capacities are violated, summed over all boundaries. It can be seen that for all the cases PSiCo gives much better congestion results than the conventional method.

Performance metrics for the initial power grid and the power grid obtained by PSiCo are listed in Table 3, in specific, the power wire area as a percentage of the area available on the two layers, the total number of nodes, the number of wires (W) in the power grid and the worst-case voltage droop. These numbers after optimization are shown in the table and may be compared with the corresponding numbers before optimization. It is seen that a well-designed power grid can save up to 12% of the total chip area while providing the optimal amount of resources needed for signal routing.

In each of the test cases, all grids whose worst-case voltage droop exceeds 0.14V (stricter than $10\%V_{dd}$ because layers M3 and M4 instead of M1 and M2 are considered) are marked as critical wires not to be removed. The voltage droop constraint (noise margin) for noise computation is chosen to be 0.17V, which is slightly stricter than $10\%V_{dd}$. To aid the rate of convergence, we heuristically terminate the SQP solver when we detect that the worst-case voltage droop in the circuit is below 0.18V, and it can be seen that the worst-case voltage droop always satisfies this requirement. The optimized noise integral, Z , for each circuit can be seen to be very small, and is nonzero due to the fact that voltage droops between 0.17V and 0.18V are tagged as “violations” by the SQP solver.

Lastly, Table 4 lists the total number of iterations and the CPU time, in minutes, required for each circuit. It can be seen that the number of iterations is typically small, and that the CPU times for these circuits are very reasonable.

7. CONCLUSION

We have proposed a new design flow for the codesign of signal routes and power grids. The technique is guided by congestion maps, and proceeds by removing non-critical power wires greedily in congested areas, and rerouting the signal wires according to the updated congestions. The effects of removing power wires are compensated for by a gradient-based wire sizing scheme. Experimental results for several benchmark circuits are presented in this paper. Future work includes applying this method to designs from industry to show the effectiveness of this method on larger industrial circuits.

8. REFERENCES

- [1] C. J. Alpert, J. Hu, S. S. Sapatneker, and P. G. Villarrubia. A Practical Methodology for Early Buffer

Circuit	Before Optimization				After Optimization				
	Area	Nodes	W	Droop	Area	Nodes	W	Droop	$Z (V \times ns)$
apte	16.8%	3411	133	0.178V	10.7%	3383	123	0.176V	2.84e-3
ami33	16.1%	11226	171	0.178V	8.90%	10245	156	0.179V	0.40e-3
ami49	19.3%	14986	198	0.178V	14.1%	13921	183	0.179V	3.65e-3
playout	19.3%	18981	223	0.170V	8.44%	16341	193	0.174V	0.10e-3
ac3	21.0%	4189	147	0.178V	9.38%	3985	128	0.180V	0.76e-3
hc7	22.4%	8882	215	0.180V	16.0%	8632	201	0.178V	7.05e-3
a9c3	19.3%	24168	251	0.175V	11.4%	22539	233	0.180V	0.85e-3

Table 3: Optimal power grid results.

	apte	ami33	ami49	playout	ac3	hc7	a9c3
CPU(min)	3.6	7.6	9.8	18.1	6.0	7.3	28.8
# Iter.	6	2	1	5	2	3	2

Table 4: CPU time statistics and number of iterations for the benchmarks.

- and Wire Resource Allocation. In *Proc. Design Automation Conference*, pages 189–194, Las Vegas, NV, June 2001.
- [2] C. J. Alpert, T. C. Hu, J. H. Huang, A. B. Kahng, and D. Karger. Prim-Dijkstra Tradeoffs for Improved Performance-Driven Routing Tree Design. *IEEE Transactions on Computer-Aided Design of ICs and Systems*, 14(7):890–896, July 1995.
- [3] P. T. Boggs and J. W. Tolle. *Sequential Quadratic Programming*. Cambridge Univ. Press, Cambridge, 1995.
- [4] H. H. Chen and D. D. Ling. Power Supply Noise Analysis Methodology for Deep-Submicron VLSI Chip Design. In *Proc. Design Automation Conference*, pages 638–643, Anaheim, CA, June 1997.
- [5] H. H. Chen and J. S. Neely. Interconnect and Circuit Modeling Techniques for Full-Chip Power Supply Noise Analysis. *IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part B*, 21(3):209–215, August 1998.
- [6] C. Chiang and M. Sarrafzadeh. Global Routing Based on Steiner Min-max Tree. *IEEE Transactions on Computer-Aided Design*, 9(12):1318–1325, December 1990.
- [7] A. R. Conn, R. A. Haring, and C. Visweswariah. Noise Considerations in Circuit Optimization. In *Proc. International Conference on Computer-Aided Design*, pages 220–227, San Jose, CA, November 1998.
- [8] S. W. Director and R. A. Rohrer. The Generalized Adjoint Network and Network Sensitivities. *IEEE Transactions on Circuit Theory*, 16(3):318–323, August 1969.
- [9] P. Feldmann, T. V. Nguyen, S. W. Director, and R. A. Rohrer. Sensitivity Computation in Piecewise Approximation Circuit Simulation. *IEEE Transactions on Computer-Aided Design of ICs and Systems*, 10(2):171–183, February 1991.
- [10] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. *User’s Guide for SOL/QPSOL: A Fortran Package for Quadratic Programming*. Department of Operations Research, Stanford University, July, 1983.
- [11] J. Hu and S. S. Sapatnekar. A Timing-constrained Algorithm for Simultaneous Routing of Multiple Nets. In *Proc. International Conference on Computer-Aided Design*, pages 99–103, San Jose, CA, November 2000.
- [12] J. D. Z. Ma and L. He. Simultaneous Signal and Power Routing under Keff Model. In *Proc. International Workshop on System-Level Interconnect Prediction*, pages 175–182, Sonoma, CA, April 2001.
- [13] R. Nair. A Simple Yet Effective Technique for Global Wiring. *IEEE Transactions on Computer-Aided Design*, 6(2):165–172, March 1987.
- [14] S. R. Nassif and J. N. Kozhaya. Fast Power Grid Simulation. In *Proc. Design Automation Conference*, pages 156–161, Los Angeles, CA, June 2000.
- [15] L. T. Pillage, R. A. Rohrer, and C. Visweswariah. *Electronic and System Simulation Methods*. McGraw-Hill, New York, NY, 1995.
- [16] E. Shragowitz and S. Keel. A Global Router Based on a Multicommodity Flow Model. *Integration: the VLSI Journal*, 5(1):3–16, March 1987.
- [17] H. Su, K. H. Gala, and S. S. Sapatnekar. Fast Analysis and Optimization of Power/Ground Networks. In *Proc. International Conference on Computer-Aided Design*, pages 477–480, San Jose, CA, November 2000.
- [18] H. Su, S. S. Sapatnekar, and S. R. Nassif. An Algorithm for Optimal Decoupling Capacitor Sizing and Placement for Standard Cell Layouts. To appear in *Proc. International Symposium on Physical Design*, San Diego, CA, April 2002.
- [19] X. Tan, C. J. R. Shi, D. Lungeanu, J. Lee, and L. Yuan. Reliability-Constrained Area Optimization of VLSI Power/Ground Networks Via Sequence of Linear Programings. In *Proc. Design Automation Conference*, pages 156–161, New Orleans, LA, June 1999.
- [20] B. S. Ting and B. N. Tien. Routing and Techniques for Gate Array. *IEEE Transactions on Computer-Aided Design*, 2(4):301–312, October 1983.
- [21] X. Wu, X. Hong, Y. Cai, C. K. Cheng, J. Gu, and W. Dai. Area Minimization of Power Distribution Network using Efficient Nonlinear Programming Techniques. In *Proc. International Conference on Computer-Aided Design*, pages 153–157, San Jose, CA 2001.
- [22] M. Zhao, R. V. Panda, S. S. Sapatnekar, T. Edwards, R. Chaudhry, and D. Blaauw. Hierarchical Analysis of Power Distribution Networks. In *Proc. Design Automation Conference*, pages 481–486, Los Angeles, CA, June 2000.