Full-Chip Routing Optimization With RLC Crosstalk Budgeting

Jinjun Xiong and Lei He, *Member, IEEE*

Abstract—**Existing layout-optimization methods for both capacitive and inductive (RLC) crosstalk reduction assume a set of interconnects with *a priori* given crosstalk bounds in a routing region. RLC crosstalk budgeting is critical for effectively applying these methods at the full-chip level. In this paper, we formulate a full-chip routing optimization problem with RLC crosstalk budgeting, and solve this problem with a multiphase algorithm. In phase I, we solve an optimal RLC crosstalk budgeting based on linear programming to partition crosstalk bounds at sinks into bounds for net segments in routing regions. In phase II, we perform simultaneous shield insertion and net ordering to meet the partitioned crosstalk bounds in each region. In phase III, we carry out a local refinement procedure to reduce the total number of shields. Compared with the best alternative approach in experiments, the proposed algorithm reduces the total routing area by up to 5.71% and uses less runtime. To the best of our knowledge, this work is the first in-depth study on full-chip routing optimization with RLC crosstalk budgeting.**

Index Terms—**Budgeting, crosstalk, inductance, interconnect optimization, routing, shielding.**

I. INTRODUCTION

AS THE CLOCK frequency continues to increase, while the minimum feature size continues to shrink, signal integrity becomes one of the primary design constraints for high-performance very large scale integrated (VLSI) chip design [1]. Because RLC crosstalk is of growing importance for gigahertz+ range IC design [2], net ordering [3], [4], and spacing [5] under *RC* crosstalk model are no longer sufficient to satisfy signal integrity constraints, due to the long range inductive coupling. Several recent studies on interconnect synthesis have considered RLC crosstalk reduction, utilizing uniform shielding [6], simultaneous shield insertion and net ordering (SINO) [7], staggered buffers [8], twisted bundle wires [9], and differential signaling [10]. However, all these methods assume a set of parallel interconnects with *a priori* given crosstalk bounds, and can only be applied within a routing region. In practice, the crosstalk bounds are usually specified at sinks. In order to apply the above region-based interconnect synthesis techniques to the full-chip level optimization, a crosstalk budgeting problem should be solved to distribute the crosstalk bounds at sinks into crosstalk bounds

for net segments in routing regions. A good crosstalk budgeting algorithm may reduce the routing congestion and routing area.

The crosstalk budgeting problem has been studied for net ordering and shield insertion under capacitive crosstalk constraints in [11]. The algorithm is based on iterations between the following two procedures: crosstalk risk estimation and crosstalk bound partitioning. Crosstalk risk estimation computes the number of shields needed to meet the partitioned crosstalk bounds for a given region with consideration of net ordering. It is formulated and solved approximately as an NP-hard graph optimization problem. Crosstalk-bound partitioning is a two-phase integer linear programming (ILP) optimization problem, minimizing the number of shields for the current global routing solution. Rip-up and reroute can be carried out to adjust the global routing to further reduce the total number of shields. However, the assumption that coupling exists only between adjacent wires no longer holds for inductive coupling, which exists between both adjacent and nonadjacent wires. Furthermore, the algorithm complexity is high as ILP is used.

In this paper, we study the full-chip routing optimization problem considering SINO with RLC crosstalk constraints. We propose a simple yet effective LSK model for the long-range RLC crosstalk at the full-chip level, develop a closed-form formula to estimate the number of shields needed by the min-area SINO solution, and formulate the crosstalk budgeting as a linear programming (LP) problem that is more efficient than ILP formulation in [11]. We finally solve the full-chip routing optimization problem by a multiphase algorithm. In phase I, we solve the full-chip crosstalk budgeting problem. In phase II, we perform SINO to meet the partitioned crosstalk bounds in each region. In phase III, we carry out a local refinement (LR) procedure to reduce the total number of shields.

The rest of the paper is organized as follows. Section II presents the background knowledge. Section III formulates the full-chip routing optimization problem, and presents our multiphase routing optimization algorithm including LP-based crosstalk budgeting. Section IV reports experiment results using Microelectronics Center of North Carolina (MCNC) benchmarks, and also presents further tuning of the crosstalk budgeting formulation. Section V concludes the paper with discussions of future work.

II. BACKGROUND

A. Preliminaries

To make the presentation simple, we assume two routing layers, one for horizontal wires and the other for vertical wires.

The authors are with the Electrical Engineering Department, University of California, Los Angeles, CA 90095 USA (e-mail: jinjun@ee.ucla.edu; lhe@ee.ucla.edu).

TABLE I
NOTATIONS THAT ARE FREQUENTLY USED IN THIS PAPER

R_t	routing regions in a chip
$\mathcal{R}$	set of all routing regions
$|\mathcal{R}|$	total number of routing regions
CLM	set of horizontal routing regions in a column
$\mathcal{CLM}$	set of all CLM's
ROW	set of vertical routing regions in a row
$\mathcal{ROW}$	set of all ROW's
l_t	length of region R_t
C_t	total number of tracks in R_t
O_t	total number of tracks occupied by obstacles in R_t
G_t	set of net segments in R_t
$|G_t|$	total number of net segments in R_t
S_t	set of shield segments in R_t
$|S_t|$	total number of shield segments in region R_t
N_i	signal net
$\mathcal{N}$	set of all signal nets
$|\mathcal{N}|$	total number of signal nets
p_{i0}	source pin of net N_i
p_{ij}	j^{th} sink of net N_i
len_{ij}	routing length from p_{i0} to p_{ij}
P_i	set of all sinks for net N_i
$|P_i|$	total number of sinks for net N_i
N_{it}	net segment of net N_i in R_t
n_i	total number of net segments in the route for net N_i
r_{it}	physical sensitivity rate of N_{it} in region R_t
H_{ij}	set of regions containing the route for sink p_{ij}
K_{it}	total inductive coupling for net segment N_{it}
$\overline{K_{it}}$	bound of K_{it}
LSK_{ij}	LSK value of sink p_{ij} of net N_i
$\overline{LSK_{ij}}$	bound of LSK at sink p_{ij} of net N_i

The routing layers are divided by prerouted power/ground (P/G) networks into routing *regions*. A route for a net contains a sequence of net *segments* in different routing regions. A *shield* is a wire directly connected (without through devices) to P/G networks. We also assume that all signal and shield wires (except for P/G wires, which are often wider) have the same width and spacing. We summarize the notations frequently used in this paper in Table I, and they will be explained in detail when they are first used.

According to [7], two signal nets are *logically sensitive* (or, in short, *sensitive*) to each other if, through logic synthesis or timing analysis [12], a switching event on one signal net causes the other to malfunction due to an excessive crosstalk noise, and vice versa. We illustrate this by using three signal nets (N_i, N_j, and N_k) in Fig. 1, where only N_k is switching, while N_i and N_j are quiet. The *sampling window*, the time period when a net's logic value is evaluated, is represented by the small dotted box. Because of the coupling effect between nets, a switching event on N_k induces coupling noises on both N_i and N_j, and both noise voltage levels are above the logic threshold (V_{th}). According to Fig. 1, the induced noise occurs during net N_i's sampling window, but not for net N_j. Therefore, N_i is sensitive to N_k, (or equivalently, N_k is an *aggressor* for N_i and N_i is a *victim* of N_k), but N_j is not sensitive to N_k. The *logic sensitivity rate* (or in short, *sensitivity rate*) of N_i is defined as the ratio of the number of aggressors for N_i to the total number of signal nets. During the global routing stage, however, two logically sensitive net segments are considered to be *physically sensitive* to each other only if they are routed within the same region, because we assume no crosstalk (coupling) between different
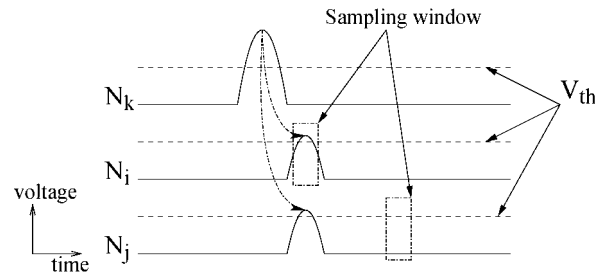

Fig. 1. Illustration of net sensitivity. The horizontal axis is time, and the vertical axis is signal–voltage level.
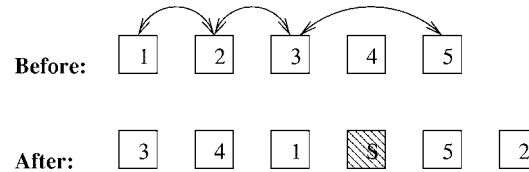

Fig. 2. Illustration of simultaneous SINO (before and after) for a routing region with five net segments, where the arrows above the net segments indicate that the two net segments are physically sensitive to each other, and the shaded square is a shield.

regions separated by P/G wires.[1] Therefore, the *physical sensitivity rate* of net segment N_{it} in region R_t is defined as the ratio of the number of aggressors in R_t for N_{it} to the total number of net segments in R_t. Because the optimization technique to be presented is a postglobal routing procedure and the routing paths for all nets are known, it is reasonable to assume that within a routing region the logic sensitivity information between nets is known and will not change during SINO. The same assumption has been employed in [13] and [14].

B. SINO Problem

Given a set of parallel interconnects with a uniform wire length, the SINO problem [7] finds a min-area SINO solution, such that all interconnects are capacitive-crosstalk-free (i.e., no physically sensitive net segments are adjacent to each other) and have inductive crosstalk less than the given bounds. In a SINO solution, a *block* includes all tracks that are sandwiched by adjacent shields. Fig. 2(a) shows an initial routing solution for five parallel nets in a routing region and Fig. 2(b)shows the final solution after SINO, where no two sensitive nets are adjacent to each other.[2] It has been shown in [7] that the SINO problem is NP-hard, therefore, a simulated annealing algorithm is developed to obtain a high-quality solution.

Because the set of parallel interconnects is equivalent to net segments in a routing region, SINO problem assumes that a global routing solution and RLC crosstalk bounds for net segments in routing regions are given *a priori*. In order to utilize SINO in the context of global routing, the following problems must be solved: 1) how to model the long-range RLC crosstalk at the full-chip level in an effective and efficient fashion and 2) how to partition RLC crosstalk bounds specified at sinks into bounds for net segments so that the area overhead due to shield

---

[1]This assumption is compatible with the K_{eff} model to be presented in Section II-C.

[2]No two sensitive nets in this example are within a same block sandwiched by adjacent shields (the leftmost and rightmost P/G networks are not drawn).
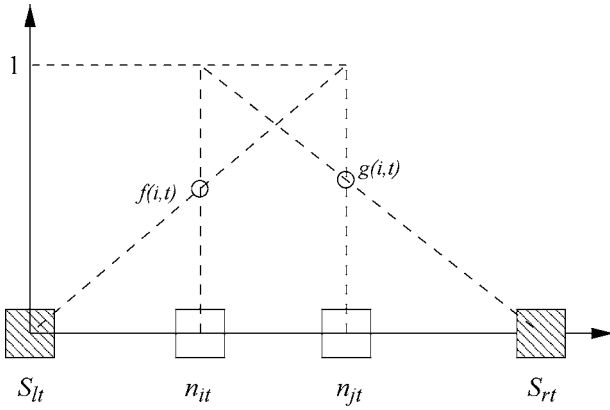
Fig. 3. Illustration of $K_{it,jt}$ computation in a given block. n_{it} and n_{jt} are track ordering numbers for net segment N_{it} and N_{jt}, and s_{lt} and s_{rt} are track ordering numbers for the edge shield segments of the block. $f(i,t)$ and $g(j,t)$ are two linear interpolation functions as shown by the dotted slope lines. The mutual inductive coupling is given by the mean of $f(i,t)$ and $g(j,t)$. Subscript t is used to indicate the routing region R_t.

insertion is minimized. We propose an efficient RLC crosstalk model to address the first problem in Section II-C, and formulate an LP-based RLC crosstalk budgeting problem to address the second problem. The LP-based budgeting formulation is made possible by a simple yet accurate formula to estimate the number of shields needed by the min-area SINO solution without actually carrying out the SINO algorithm. We discuss shield estimation in Section II-D and budgeting formulation in Section III.

C. LSK Model for RLC Crosstalk

It has been proposed in [15] that the coupling coefficient between two net segments N_{it} and N_{jt} can be used to characterize the inductive coupling effect between them. The coefficient is defined as

$$K_{it,jt} = \frac{M_{it,jt}}{\sqrt{L_{it} \cdot L_{jt}}} \tag{1}$$

where $M_{it,jt}$ is the mutual inductance between N_{it} and N_{jt}, and L_{it} and L_{jt} are self inductance for N_{it} and N_{jt} under the loop inductance model, respectively. Extensive experiments have shown that $K_{it,jt}$ is relatively independent of such technology parameters as wire width, thickness, length, spacing, and frequency, and a formula-based K model has been developed to compute $K_{it,jt}$ [15]. As shown in Fig. 3, n_{it} and n_{jt} are track-ordering numbers for net segments N_{it} and N_{jt}, respectively, and s_{lt} and s_{rt} are track ordering numbers for the two edge shield segments. When N_{it} and N_{jt} are in different "blocks" separated by shield segments, $K_{it,jt}$ equals to zero or a small constant. When the two net segments are in the same block, we consider the following two special cases first.

1) When $n_{it} = n_{jt}$, the mutual inductance is reduced to self inductance and $K_{it,it} = 1$ by definition.
2) When n_{it} (or n_{jt}) becomes s_{lt} (or s_{rt}), $K_{it,jt} = 0$ because it is now defined for two segments of the same current loop and should be zero under the loop inductance model.

In general, $K_{it,jt} (= K_{jt,it})$ should be between 0 and 1, and can be approximated by a linear interpolation of the above two special cases. Therefore, we have

$$K_{it,jt} = \frac{\left(f(i,t) + g(j,t)\right)}{2} \tag{2}$$

where $f(i,t) = (n_{it} - s_{lt})/(n_{jt} - s_{lt})$ and $g(j,t) = (s_{rt} - n_{jt})/(s_{rt} - n_{it})$ are two linear interpolations of 0 and 1.

The K model is reasonably accurate—within a $+20\%$ to -10% error range compared to the three-dimensional field solver FastHenry [16]—and tends to be conservative. Furthermore, an effective K model (or in short, K_{eff} model) is proposed to use the weighted sum of coupling coefficients (K_{it}) as the figure of merit for the total amount of inductive noise induced on the net segment N_{it}. The K_{it} can be calculated by

$$K_{it} = \sum_{j \neq i} S_{ij} \cdot K_{it,jt} \tag{3}$$

where $S_{ij} = 1$ for all net segments N_{jt} that are sensitive to N_{it}, otherwise $S_{ij} = 0$.

It has been shown in [7] that the K_{eff} model has a high fidelity versus SPICE calculated RLC noise for a SINO solution with a fixed wire length. That is, a signal net in a SINO solution with a higher K value given by the K_{eff} model also has a higher SPICE-computed voltage under the distributed RLC circuit model. Such fidelity holds under the assumption that no sensitive nets are adjacent to each other in a SINO solution and, therefore, there is no capacitive noise. The K_{eff} model is computationally simple and convenient to use in early design stages.

Note that the K_{eff} model is proposed for wires with a fixed length. To consider the effect of interconnect length and the general case where the total coupling is not uniform among routing regions, we propose a length-scaled K_{eff} (LSK) model, where the LSK value for a net N_i at its jth sink is defined as

$$LSK_{ij} = \sum_{t \in H_{ij}} l_t \cdot K_{it} \tag{4}$$

where l_t is the length of R_t, K_{it} is the total coupling for N_{it}, and the sum is over the path from the source to the sink. This model can be justified by the following experiments. We randomly choose four SINO solutions, and assign different wire lengths to each SINO solution. A distributed full RLC circuit model is generated for each SINO solution with the assigned wire length according to [17]. Specifically, we first divide each signal wire into 100 μm-long segments, and then connect each shield to P/G wires for every 100 μm-long segment. Each signal segment is modeled by an RLC π-model with a coupling capacitance to its adjacent segment that does not belong to the same wire. There is a coupling inductance between any two segments, including those that belong to the same wire. The capacitance and inductance are calculated based on [18] and [17], respectively. We find the worst case noise via SPICE simulation for each distributed RLC circuit by using the worst-case noise algorithm proposed in [19]. Fig. 4 plots the relation between the worst case noise and the wire length for all four SINO solutions.
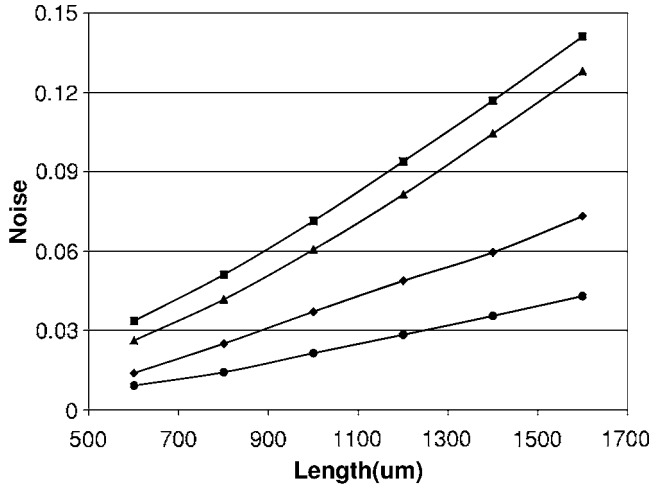
Fig. 4. Linear scalability of SPICE computed worst case noise versus the length for four min-area SINO solutions.

We observe that the worst-case noise is nearly proportional to the interconnect length.

A similar length scaled approach has been adopted in the early work by [11], [14] to model the capacitive crosstalk by the product of the coupling length and the coupling capacitance. However, no SPICE simulation results were shown in [11], [14] to verify the linear relationship between the length and the *RC* crosstalk coupling.

D. Shield Estimation

In this paper, we consider the SINO formulation under the K_{eff} model, i.e., the inductive noise bounds are given as K_{th} bounds. According to [7], the number of shields needed for a min-area SINO solution depends on both the sensitivity rates (r_{it}) and the noise bounds (K_{th}) of all net segments in a routing region. In [7], a routing region with 32 net segments is examined with a uniform sensitivity rate and a uniform K_{th} bound for each net segment. The examined sensitivity rates are 0.1, 0.3, and 0.6, and the examined K_{th} bounds are 0.5, 1.0, 1.5, and 2.0. For each combination of sensitive rates and K_{th} bounds, the average shield number of five runs of the SINO algorithm is recorded. Based on the data from [7], we plot the relationships between the number of shields and the sensitive rate and noise bound in Fig. 5(a) and (b), respectively.

According to Fig. 5, the higher the sensitivity rate, the greater the number of shields and the higher the noise bound, the fewer the number of shields. Furthermore, a rough linear relationship can be observed from both figures. We believe that these observations would still be true even if we allow different net segments to have different sensitivity rates and K_{th} bounds. Therefore, a complete linear shield estimation template can be used to capture these linear characteristics

$$|S_t| = a_1 \cdot \sum_{N_{it} \in G_t} \overline{K_{it}} \cdot r_{it} + a_2 \cdot \sum_{N_{it} \in G_t} r_{it} + a_3 \cdot \sum_{N_{it} \in G_t} \overline{K_{it}} + a_4 \tag{5}$$

where notations from Table I are used, and $a_1 \sim a_4$ are constant coefficients. There are four linear terms in (5). However, not all terms are statistically necessary (or significant) for an accurate

estimation. Statistical analysis techniques, such as analysis of variance (ANOVA) [20] can be used to identify which terms are statistically significant in (5). This is equivalent to testing whether the null hypothesis of a coefficient a_i is actually zero [20].

We randomly generate 10 000 routing solutions with different combinations of number of net segments, sensitivity rate r_{it} (ranging from 20% to 80%), and $\overline{K_{it}}$ values. For each routing solution, we run the SINO algorithm and collect the number of shields in each SINO solution. The 10 000 point data set is then used to perform ANOVA in a commercial statistical analysis package [21]. The p-value is used to test whether the null hypothesis is true for each coefficient [20], i.e., the smaller the p-value, the less likely the parameter is actually zero, or vice versa. The obtained p-values are as follows: $p(a_1) = 0.000\,01$, $p(a_2) = 0.000\,01$, $p(a_3) = 0.936\,71$, and $p(a_4) = 0.344\,21$, which indicate that the last two linear terms in (5) are not statistically significant to improve the accuracy of shield estimation.[3] Therefore, the final shield estimation formula considered in this paper is as follows:

$$|S_t| = a_1 \cdot \sum_{N_{it} \in G_t} \overline{K_{it}} \cdot r_{it} + a_2 \cdot \sum_{N_{it} \in G_t} r_{it}. \tag{6}$$

In order to obtain the coefficients in (6) for a given routing region, we evenly divide the collected 10 000 data sets into five groups, and employ a multivariable curve-fitting process that minimizes the least square error [22]–[24] to obtain the coefficients in (6) under different groups. The results are shown in Table II, where the last column δ are absolute values of the standard deviation divided by the mean among five groups. δ value can be used to measure the variation of the parameters. The larger the δ, the bigger the variation.

According to Table II, the values of the coefficient of determination (R^2) under all groups show a very satisfactory goodness-of-fit for (6) in terms of estimation accuracy.[4] Moreover, the coefficients obtained from different groups are very consistent as indicated by the smaller δ for a_1 and a_2. The consistency of a_1 and a_2 implies that (6) is also very robust even in the presence of variation of sensitivity rates and K_{th} bounds. Another observation from Table II is that the coefficient of a_1 are negative across all groups. This property ensures that all coefficients of $\overline{K_{it}}$ in (6) are negative as well. The physical meaning is that increasing the crosstalk bound reduces the number of shields needed for a min-area SINO solution, which is in accordance with the observation from Fig. 5(b). Therefore, having negative signs for $\overline{K_{it}}$'s coefficients is very important to maintain shield estimation (6) physically correct. The coefficients finally used in this paper are obtained by merging all data from the above five groups together and redoing the curve-fitting process, which leads to $a_1 = -0.104\,91$ and $a_2 = 0.493\,92$ with $R^2 = 0.87$.

---

[3]For example, "$p(a_1) = 0.000\,01$" means that there is a 0.001% chance that the actual coefficient is zero; and "$p(a_3) = 0.936\,71$" means there is a 93.671% chance that the actual parameter value is zero. Therefore, in cases where p-values of coefficients are large, they can usually be removed from the model without affecting the regression accuracy [20].

[4]The higher the coefficient of determination (R^2), the better the goodness-of-fit [20]. A perfect match has $R^2 = 1$. The relative high R^2 values further confirm that the selected two linear terms are able to achieve high estimation accuracy.
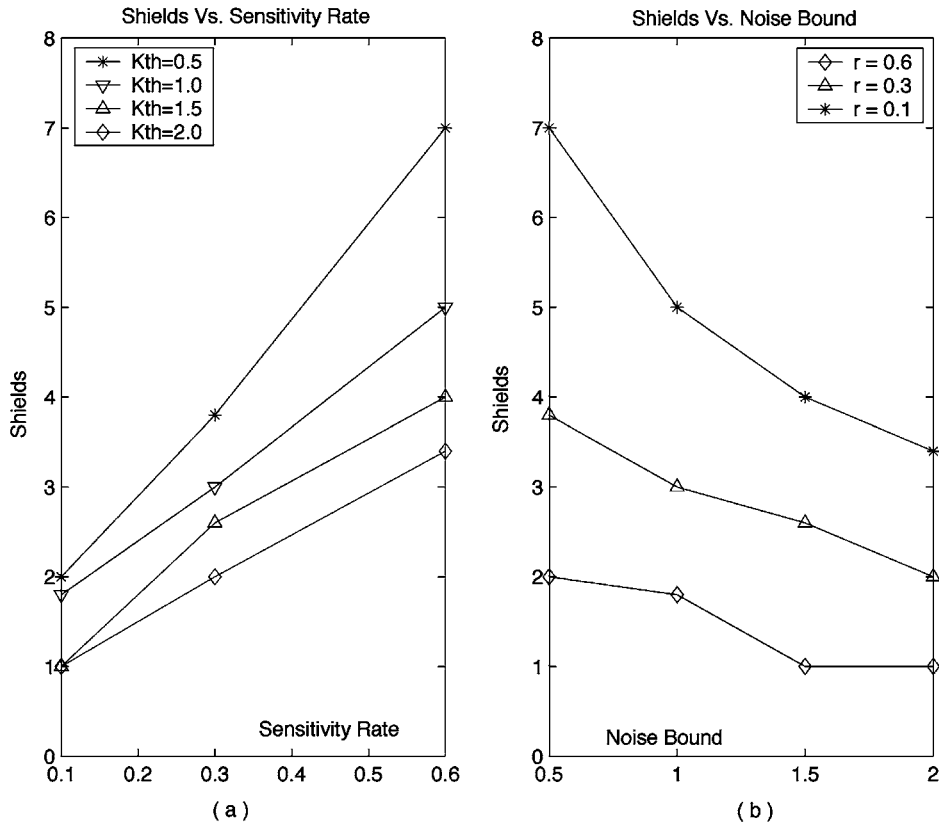
Fig. 5.   Illustration of the relationships (a) between shield numbers and sensitivity rates and (b) between shield numbers and noise bounds.

TABLE II
COEFFICIENTS FOR SHIELD ESTIMATION IN (6)

	I	II	III	IV	V	δ
a_1	-0.10956	-0.10408	-0.09781	-0.10605	-0.10795	0.04337
a_2	0.50339	0.47515	0.47025	0.49420	0.51500	0.03832
R^2	0.8186	0.8071	0.8419	0.8711	0.8972	-

Note that the number of net segments $|G_t|$ and physical sensitivity rates r_{it}s are fixed in a region R_t for the given global routing solution. Hence, (6) can be further simplified as a linear combination of the given noise bounds $\overline{K_{it}}$

$$|S_t| = \sum_{N_{it} \in G_t} \alpha_{it} \cdot \overline{K_{it}} + \beta_t \qquad (7)$$

where $\alpha_{it} = a_1 \cdot r_{it}$ and $\beta_t = a_2 \cdot \sum_{N_{it} \in G_t} r_{it}$, according to (6). Because a_1 is negative, so are all coefficients (α_{it}s) of $\overline{K_{it}}$s.

III. PROBLEM FORMULATIONS AND ALGORITHMS

A. Full-Chip Routing Optimization Problem

Formulation 1: (**Full-chip routing optimization**) Given a global routing solution and RLC crosstalk bounds for all sinks, the full-chip routing optimization problem determines the RLC coupling bound for each net segment and finds a min-area SINO solution within each region, such that the RLC crosstalk bound is satisfied at each sink and the total routing area is minimized.

The full-chip routing optimization problem has a high complexity, as its subproblem to find a SINO solution within a region

Full-chip routing optimization algorithm overview
Given global routing solution and RLC crosstalk bound
for each sink
Phase I: Crosstalk budgeting at the full-chip level.
Phase II: SINO within each region.
Phase III: Local refinement.

Fig. 6.   Overview of the three-phase algorithm for full-chip routing optimization.

is already NP-hard [7]. Therefore, we develop the following three-phase heuristic algorithm (see Fig. 6). Phase I finds the crosstalk bound for each net segment in all regions, and is called crosstalk budgeting problem. The input to the crosstalk budgeting problem is the crosstalk bound (in LSK value in this paper) for each sink and the output is the noise bound for each net segment (in K value). The output of noise bounds is the input to the Phase II algorithm. Phase II performs SINO in each region by using the algorithm developed in [7]. Phase III is a LR procedure that completely eliminates the remaining (but very limited) RLC crosstalk violations and further reduces the number of shields.[5] Below, we discuss Phase I and Phase III algorithms in detail.

---

[5]The proposed full-chip routing optimization fits in between the existing global routing stage and detailed routing stage. The output of our algorithm is track assignment solutions for all net-segments, and we still need a detailed routing algorithm to determine the connections of the same net in different routing regions. As the required detailed routing algorithm should not change our track assignment solutions, therefore, minimizing routing area in this stage is of paramount importance, and it will be chosen as a major design goal in this work. We do not consider detailed routing in this paper.
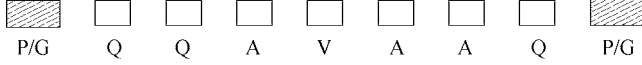
Fig. 7. Illustration on how to obtain the maximum $\overline{K_{it}^{\max}}$ for a victim net segment N_{it} (V) with three aggressors (A) and three quiet signals (Q) or free tracks for a given routing region R_t. The leftmost and the rightmost are P/G networks.

B. RLC Crosstalk Budgeting

1) Common Constraints: In this work, we present an *optimal RLC crosstalk budgeting* scheme based on LP. There are three common design constraints that must be satisfied: i) crosstalk bound constraint—the LSK value should be less than the given crosstalk bound $\overline{LSK}$ at each sink; ii) positive shield number constraints—the number of estimated shields should be positive at each region; and iii) worst-case upper bound—for net segment N_{it} in region R_t, the budgeted bound $\overline{K_{it}}$ should not exceed a maximum value $\overline{K_{it}^{\max}}$ that it may suffer under the worst case. $\overline{K_{it}^{\max}}$ can be obtained as follows. Assuming there is no shield in R_t, the victim N_{it} is placed at the center of the region, and all N_{it}'s aggressors are placed as close to it as possible. Fig. 7 illustrates this scenario for a victim (V) with three aggressors (A) in a region of seven tracks. The victim N_{it}'s K_{it}-value obtained in this case is $\overline{K_{it}^{\max}}$.

The three constraints can be expressed formally as follows:

$$\sum_{R_t \in H_{ij}} l_t \cdot \overline{K_{it}} \leq \overline{LSK_{ij}} \quad \forall p_{ij} \in P_i \ and \ \forall N_i \in \mathcal{N} \quad (8)$$

$$\sum_{N_{it} \in G_t} \alpha_{it} \cdot \overline{K_{it}} + \beta_t \geq 0 \quad \forall R_t \in \mathcal{R} \quad (9)$$

$$\overline{K_{it}} \leq \overline{K_{it}^{\max}} \quad \forall N_{it} \in R_t \ and \ \forall R_t \in \mathcal{R}. \quad (10)$$

Because all of the above constraints should be considered for any LP-based budgeting scheme to be presented, we will not repeat them explicitly later on.

2) One-Dimensional (1-D) Optimal Budgeting: Without loss of generality, we call the global routing within a row of regions that allow only horizontal wires as 1-D routing (e.g., bus structures). The total number of tracks occupied by net segments, shields and/or obstacles in region R_t is defined as its height h_t, and the maximum height $h_{\max}$ among all regions is defined as the height of the routing solution. *Critical regions* are routing regions that define $h_{\max}$. We then formulate the following 1-D problem:

Formulation 2: (**1-D crosstalk budgeting problem**) For a given 1-D routing solution, the 1-D problem partitions crosstalk bounds among regions such that the maximum height $h_{\max}$ is minimized. The 1-D problem can be mathematically stated as

$$\textbf{minimize } h_{\max}$$
$$\textbf{s.t.}$$
$$\sum_{N_{it} \in G_t} \alpha_{it} \cdot \overline{K_{it}} + \beta_t + |G_t| + O_t \leq h_{\max} \quad \forall R_t \in \mathcal{R} \quad (11)$$

where $|G_t|$ represents the total number of net segments in R_t, and O_t represents the total number tracks occupied by obstacles in R_t. The lefthand side of new constraint (11) computes the estimated height of region R_t, and the $h_{\max}$ is the maximum height of the whole row. All constraints in (11) together enforce
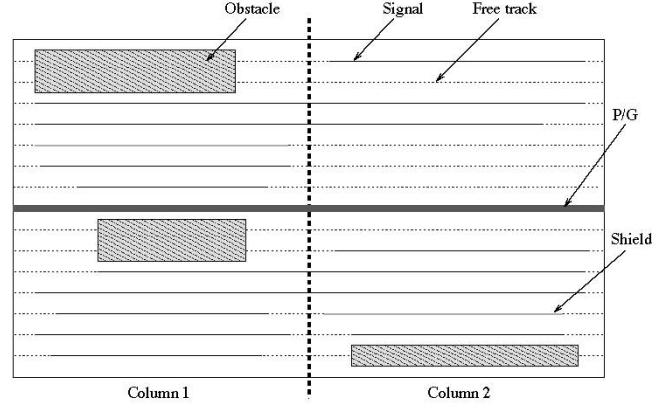

Fig. 8. Horizontal routing layer with two columns and four routing regions. Free tracks, signals, shields, obstacles, and prelayed P/G network are illustrated as well.

that the height of any routing region should be less than the maximum height of the whole row. Furthermore, $\overline{K_{it}}$ are the unknowns that we need to solve for the 1-D problem and for the two-dimensional (2-D) p problem to be presented as well.

3) Pseudo 2-D Optimal Budgeting: For a 2-D global routing consisting of an array of routing regions, let CLM (ROW) be the set of routing regions in a column (row) for horizontal (vertical) wires, and $\mathcal{CLM}$ ($\mathcal{ROW}$) be the set of all $CLMs$ ($ROWs$). The height h for CLM is defined as the total number of tracks occupied by net segments, shields and obstacles in CLM, and the height $h_{\max}$ of the total routing area is defined as the maximum h among all $CLM \in \mathcal{CLM}$. The width w for ROW and $w_{\max}$ for the total routing area can be defined similarly. Consistent with the 1-D problem, *critical regions* are routing regions that define $h_{\max}$ or $w_{\max}$. In Fig. 8, we show an example of two columns in the horizontal routing layer with two columns and four routing regions. Because the maximum height of the two columns is decided by column 1, regions in column 1 are called critical regions. Similar to [25], we will minimize $h_{\max}$ and $w_{\max}$ in our problem formulation. The pseudo 2-D optimal budgeting (2-Dp) problem is defined as follows.

Formulation 3: (**Pseudo 2-D crosstalk budgeting (2-Dp) problem**) For a given global routing solution, the 2-Dp problem partitions crosstalk bounds among all routing regions, such that the weighted sum $\gamma \cdot h_{\max} + \theta \cdot w_{\max}$ is minimized, where γ and θ are two positive constants. The 2-Dp problem can be mathematically stated as

$$\textbf{minimize } \gamma \cdot h_{\max} + \theta \cdot w_{\max}$$
$$\textbf{s.t.}$$
$$\sum_{R_t \in CLM} \left(\sum_{N_{it} \in G_t} \alpha_{it} \cdot \overline{K_{it}} + \beta_t + |G_t| + O_t \right) \leq h_{\max}$$
$$\forall CLM \in \mathcal{CLM} \quad (12)$$
$$\sum_{R_t \in ROW} \left(\sum_{N_{it} \in G_t} \alpha_{it} \cdot \overline{K_{it}} + \beta_t + |G_t| + O_t \right) \leq w_{\max}$$
$$\forall ROW \in \mathcal{ROW} \quad (13)$$

where the left hand sides of constraints (12) and (13) compute the height and width of an entire column and row, respectively. We approximate the objective of minimizing the total routing

LR-I: Eliminate crosstalk violations
While(there has crosstalk violation){
  Find N_i's j^{th} sink p_{ij} with most severe crosstalk violation;
  Find the region R_t containing N_{it} with highest K_{it};
  Insert a shield into R_t;
  Simultaneous ordering of shields and net segments;
  Update *LSK slacks* for all affected paths;
}

LR-II: Reduce shield number
While(removing shields is possible){
  Find net N_i whose j^{th} sink p_{ij} has largest *LSK slack*;
  Find the region R_t containing N_{it} with least K_{it};
  Remove a shield from R_t;
  Simultaneous ordering of shields and net segments;
  If(no violation is found){
    Accept the new solution for R_t;
    Update *LSK slacks* for all affected paths;
  }**else**
    Restore the old solution for R_t;
}

Fig. 9.   Phase III LR algorithm.

area ($h_{\max} \cdot w_{\max}$) by minimizing the weighted sum of $h_{\max}$ and $w_{\max}$. Because $h_{\max}$ and $w_{\max}$ often have similar values in practice, minimizing their weighted sum provides a good solution for minimizing their product but with a much reduced complexity.[6]

4) Main Theorem: According to the RLC crosstalk budgeting formulations, we have the following theorem.

Theorem 1: Both 1- and 2-Dp problems are LP problems.

Sketch of proof: It is easy to verify that all constraints (8)–(12) are linear, and the objective functions of 1- and 2-Dp are linear too, hence both 1- and 2-Dp are LP problems [26].   □

There are well-developed LP solvers available from both the commercial world (like [27]) and the academia (like [28]). In order to solve our crosstalk budgeting problem, we can utilize any of these solvers. For the rest of the paper, we use LP to represent either 1- or 2-Dp whenever there is no ambiguity.

C. LR

As shown in Fig. 9, there are two loops (denoted as LR-I and LR-II, respectively) in Phase III.

The SINO algorithm from [7] is based on simulated annealing, and the crosstalk and area constraints are implemented as two components of the cost function. Therefore, a very limited number of crosstalk violations may exist after Phase II. To implement a "better" SINO algorithm such that all net segments satisfy the partitioned crosstalk bounds within *each* and *every* region may lead to over-design with shields more than needed. Instead, we choose to eliminate the remaining crosstalk violations through LR-I. Let *LSK slack* of a sink be the gap between $\overline{LSK}$ and LSK at the sink, therefore, the crosstalk violation at each sink is indicated by a negative

---

TABLE III
TEST CIRCUIT CHARACTERISTICS

Test Ckts	Number of nets	Number of pins	Regions (row × col)	Obstacle segments
bus.1	64	128	1 × 10	16
bus.2	64	128	15 × 10	32
MCNC-1	607	1835	8 × 16	460
MCNC-2	677	2155	16 × 16	643
MCNC-3	814	2713	128 × 16	5758

LSK slack value. In LR-I, we first find a net N_i with the most negative LSK slack (i.e., the most severe crosstalk violation) at sink p_{ij}, and locate a routing region R_t with the highest K_{it} for segment N_{it}. We then insert exactly *one* more shield into R_t, and carry out simultaneous ordering of both shields and net segments to obtain the minimum crosstalk for N_{it}, but still satisfy crosstalk bounds for all other net segments in R_t. Such a net ordering is implemented as a simpler case of the SINO algorithm [7] without shield insertion or deletion during the simulated annealing process. Inserting a shield in R_t may reduce K_{it} as well as K_{jt} for other segments N_{jt} in R_t. Hence, we need to update the LSK slacks for all nets passing R_t. The iteration stops when there is no crosstalk violation for any net.

LR-II reduces the total shield number by exploiting the remaining LSK slacks to remove unnecessary shield segments in each region. We first find a route that has the largest LSK slack at sink p_{ij} for net N_i, then we find a region R_t with the least K_{it} value for net segment N_{it}. Exactly one shield will be removed from R_t and then simultaneous ordering of both signals and net segments is performed to obtain a solution with the minimum sum of K values for all net segments in R_t. Removing a shield may cause some net segments' K values to increase. If these increments can be compensated by their LSK slacks respectively, no crosstalk violation occurs. Therefore, we accept the new solution for R_t and update the affected LSK slacks for all nets passing R_t. Otherwise, we restore our original solution for R_t. The iteration stops when removing shields is no longer possible in any region.

IV. EXPERIMENT RESULTS AND ALGORITHM TUNING

We have implemented our algorithm in C/C++ on UNIX/Linux platforms. A simplex-based LP engine, *lp-solver* [8] is used to solve the LP-based budgeting in Phase I. We present the experiment results by using two bus structures and three MCNC benchmarks. The MCNC benchmarks are placed by DRAGON [29], and routed by our own global router implementing the iterative deletion algorithm [25]. In all experiments, we assume that buffers are inserted so that no wires are longer than 1000 μm. We consider two average logic sensitivity rates of 50% and 70% over the chip, and report the ranges of physical sensitivity rates for all benchmarks under the first column in Tables IV–IX. We also assume that all sinks have the same noise bound ($\overline{LSK}$) as 1000. Nevertheless, our algorithm and implementation can handle nonuniform noise bounds as well. Obstacles are randomly generated in each region. Table III summarizes the characteristics of our test circuits, where the net number and pin number do not include the nets and pins within the same routing region. In the following,

---

[6]Minimizing the product of $h_{\max}$ and $w_{\max}$ is a quadric programming problem, but minimizing the sum is an LP problem. Moreover, minimizing $h_{\max}$ and $w_{\max}$ is closely related to minimizing routing congestion in *critical regions*. Note that in our 1- and 2-Dp budgeting formulations, we do not explicitly express the limited routing resource constraints. But minimizing $h_{\max}$ and $w_{\max}$ can indirectly enforce our budgeting algorithms to utilize the limited resources economically, hence, we can satisfy the same design constraints implicitly.

TABLE IV
COMPARISON OF THE TOTAL NUMBER OF SHIELDS, ROUTING AREAS IN $h_{\max}$, AND THE MAXIMUM/AVERAGE LSK VALUES UNDER
UD+LR AND LP+LR BUDGETING SCHEMES AFTER EACH PHASE ALGORITHM FOR TWO 64-BIT BUS STRUCTURES.
THE VALUES IN PARENTHESIS ARE LP+LR'S ROUTING AREA REDUCTION OVER UD+LRs IN PERCENTAGE

1	2	3	4	5	6	7	8	9
Sensitive Rate	Algorithm	Phase I		Phase II		Phase III		$\overline{LSK}$ 1000
		Shield	h_{max}	Shield	h_{max}	Shield	h_{max}	
bus.1								
45~61%	UD+LR	114.1	92.7 (0.00%)	73	88 (0.00%)	45	85 (0.00%)	950.0/553.9
	LP+LR	128.0	80.0 (-13.70%)	108	88 (0.00%)	72	86 (1.18%)	994.4/520.6
67~77%	UD+LR	158.8	97.6 (0.00%)	103	92 (0.00%)	76	88 (0.00%)	981.8/701.5
	LP+LR	158.8	83.4 (-14.55%)	220	99 (7.61%)	187	94 (6.82%)	992.5/321.2
bus.2								
45~61%	UD+LR	114.1	108.7 (0.00%)	72	104 (0.00%)	48	101 (0.00%)	958.1/556.2
	LP+LR	132.7	96.0 (-11.68%)	103	102 (-1.92%)	70	100 (-0.99%)	981.6/451.8
67~77%	UD+LR	158.8	113.6 (0.00%)	104	108 (0.00%)	79	105 (0.00%)	964.2/695.4
	LP+LR	184.7	96.0 (-15.49%)	178	104 (-3.70%)	141	100 (-4.76%)	995.0/652.6

TABLE V
COMPARISON OF THE TOTAL NUMBER OF SHIELDS, ROUTING AREAS IN $(h_{\max} + w_{\max})$, AND THE MAXIMUM/AVERAGE LSK VALUES
UNDER UD+LR AND LP+LR BUDGETING SCHEMES FOR THREE MCNC BENCHMARK CIRCUITS. THE VALUES IN
PARENTHESIS ARE LP+LR'S ROUTING AREA REDUCTION OVER UD+LRs IN PERCENTAGE

1	2	3	4	5	6	7	8	9
Sensitive Rate	Algorithm	Phase I		Phase II		Phase III		$\overline{LSK}$ 1000
		Shield	$h_{max} + w_{max}$	Shield	$h_{max} + w_{max}$	Shield	$h_{max} + w_{max}$	
MCNC-1								
45~78%	UD+LR	108.6	230.4 + 147.5 (0.00%)	168	232 + 151 (0.00%)	98	220 + 149 (0.00%)	995.6/283.4
	LP+LR	444.0	218.0 + 159.6 (-0.08%)	405	246 + 161 (6.27%)	227	231 + 154 (4.34%)	997.9/201.1
67~87%	UD+LR	159.0	236.8 + 149.6 (0.00%)	244	237 + 154 (0.00%)	161	228 + 150 (0.00%)	999.4/311.0
	LP+LR	533.1	218.0 + 167.9 (-0.13%)	640	255 + 177 (10.49%)	444	241 + 169 (8.47%)	986.1/209.4
MCNC-2								
44~79%	UD+LR	261.0	193.6 + 194.8 (0.00%)	258	194 + 194 (0.00%)	129	188 + 192 (0.00%)	995.5/269.9
	LP+LR	655.8	182.0 + 202.7 (-0.95%)	607	197 + 213 (5.67%)	264	187 + 198 (1.32%)	998.2/198.6
66~87%	UD+LR	368.5	199.2 + 199.1 (0.00%)	395	200 + 199 (0.00%)	228	191 + 193 (0.00%)	998.4/310.8
	LP+LR	676.4	182.0 + 207.8 (-2.13%)	1215	225 + 235 (15.29%)	790	210 + 219 (11.72%)	905.5/102.1
MCNC-3								
44~78%	UD+LR	2749.4	214.7 + 939.7 (0.00%)	1894	210 + 895 (0.00%)	656	200 + 885 (0.00%)	998.8/109.1
	LP+LR	3574.3	195.0 + 886.0 (-6.36%)	3425	214 + 1037 (13.21%)	1132	201 + 946 (5.71%)	1000.0/37.8
66~88%	UD+LR	3859.1	222.3 + 963.8 (0.00%)	2872	217 + 920 (0.00%)	1224	206 + 889 (0.00%)	999.7/254.8
	LP+LR	4857.6	195.0 + 894.5 (-8.14%)	5725	223 + 1112 (17.41%)	2953	209 + 1017 (11.96%)	998.5/67.6

we present the initial experiment results in Section IV-A, and discuss the tuning of our budgeting formulation and improved experiment results in Section IV-B.

A. Initial Experiment Results and Discussions

1) Initial Comparison Between Uniform Budgeting (UD)- and LP-Based Algorithms: In order to show the effectiveness of our LP-based budgeting scheme, we further propose an alternative budgeting scheme called UD scheme, which distributes the crosstalk bound uniformly along the route. Let $\overline{LSK_{ij}}$ be the crosstalk bound at sink p_{ij} for net N_i, len_{ij} be the total routing length from source p_{i0} to sink p_{ij}, then each routing region R_t on the path is assigned a *uniform* crosstalk budget

$$\overline{K_{it}} = \frac{\overline{LSK_{ij}}}{len_{ij}}. \qquad (14)$$

If segment N_{it} is shared by multiple paths starting from the same source to different sinks, we use the minimum value computed for these paths according to (14). However, we point out that the uniform distribution is for an individual net only, and different nets have different values according to their own crosstalk budgets. UD is unable to consider the nonuniform routing congestion among different regions.

For complete and fair comparison, Phases II and III will be applied to both UD-based and LP-based budgeting schemes. We denote the full-chip routing optimization algorithm with UD in Phase I as UD+LR, and the one with LP in Phase I as LP+LR. Since UD+LR provides an alternative way to do full-chip routing optimization, it will be used to compare our proposed LP+LR optimization algorithm.

Tables IV and V present the initial experiment results for two bus structures and three MCNC benchmark circuits, respectively. According to column 9 in Tables IV and V, the maximum LSK values among all sinks are all smaller than the given bound $\overline{LSK}$, i.e., both UD+LR and LP+LR algorithms completely eliminate crosstalk violations. Furthermore, when the sensitivity rate and number of obstacles increase, the routing area and number of shields increase as well for both algorithms. The same as the objective in our LP formulation, the shield and area in Phase I for both UD+LR and LP+LR are based on our shield-estimation formula (7). As shown in column 4 of Tables IV and V, LP-based budgeting does achieve smaller area compared to UD, and the area reduction can be as high as 15.49%. All the above observations are as expected, and indicate the correctness of our problem formulation and program implementation.

Furthermore, we compare results of Phases II and III between UD+LR and LP+LR. For bus structures from Table IV, we observe that when the routing resources get more restrictive (because of more obstacles for *bus.2*), LP+LR is better than UD+LR after Phases II and III. When the routing resources are not that restrictive (e.g., *bus.1*), LP+LR is not necessarily better than UD+LR after Phases II and III. For MCNC circuits, Table V shows that LP+LR performs worse than UD+LR after Phases II and III. Below, we discuss why LP+LR may be worse and present the improved LP formulation in order for LP+LR to perform better than UD+LR.

2) Limitation of Initial LP-Based Budgeting: Our shield estimation formula (7) has the following limitations. First, the formula results in a continuous value, but the number of shields is an integer in reality. Even though we could round the estimated shield number to an integer, it might still be different from the number of shields obtained by the detailed SINO algorithm.[7] Second, the formula (7) only reflects the total effects of all net segments in a given region, and can not differentiate the individual contribution of each net segment clearly. In contrast, our LP formulation treats the contribution of each net segment as an individual optimization variable. Knowing the difference between different net segments is the key for the LP formulation to balance the tradeoff among all net segments and, therefore, to achieve the global optimal budgeting solution. Because of this discrepancy between what our estimation formula can provide and what our LP+LR formulation requires, LP+LR may be worse than UD+LR in Phases II and III.

The discrepancy between our shield estimation and LP formulation can be further illustrated by a simple example. Let us assume that in a given routing region, all net segments have the same sensitivity rate and the sum of $\overline{K_{it}}$ over all net segments is fixed as $\overline{K_t^{\text{sum}}}$. In this case, evenly distributing $\overline{K_t^{\text{sum}}}$ among all net segments or giving $\overline{K_t^{\text{max}}}$ to only one net segment does not make much difference in terms of our LP solution, but it may make a difference in reality. For example, if a net segment has a high coupling bound and the rest have low bounds, the SINO solution may need a large number of shields in order to meet these low coupling bounds. In contrast, the SINO solution under more balanced coupling bounds for all net segments may have fewer shields.

B. Improved Budgeting Formulation and Experiment Results

1) New LP-Based Budgeting Formulations: To avoid the above discrepancy between our budgeting formulation and shield estimation, we can either develop a new shield estimation formula that captures the precise contribution of each individual net segment to the shield estimation in terms of the crosstalk bound, or impose more constraints to guide the budgeting formulation to better use our current shield estimation. The choice between the two options reflects the tradeoff between estimation accuracy and solution efficiency. The first approach of a more sophisticated shield estimation may lead to an intractable budgeting formulation. We believe that the second approach may be better if we can keep all

new constraints linear and, therefore, maintain the budgeting problem as a more tractable LP problem.

Based upon the discussion in Section IV-A-2, we propose the following two heuristic constraints. The first one is the universal upper bound constraint given by

$$\overline{K_{it}} \leq -\frac{a_2}{a_1} \quad \forall N_{it} \in R_t, \tag{15}$$

where a_1 and a_2 are constant coefficients in formula (6). The universal upper bound constraint prevents a budgeting solution that favors one net segment too much. In fact, constraint (15) could be derived from the positive shield constraints (9) if we assume that all net segments in one region have the same bound as $\overline{K_{it}}$. Experiments show that (15) often provides a tighter upper bound for $\overline{K_{it}}$ than (10) does.

The second heuristic constraint is based on the following intuition. For a given routing region, a good budgeting should give a higher $\overline{K_{it}}$ to a net segment N_{it} with a higher sensitivity rate r_{it}, as N_{it} is likely to suffer higher RLC crosstalk. That is

$$\overline{K_{jt}} \leq \overline{K_{it}} \quad \forall r_{jt} \leq r_{it}. \tag{16}$$

Theoretically, the above constraint is valid only if the congestion differences between different routing regions is ignored. However, it leads to nice results in our experiments with the presence of nonuniform congestion distribution.

Because both (15) and (16) are linear, our 1-D and 2-D*p* budgeting formulations considering the two new constraints are still LP problems. For the rest of the paper, we call the LP budgeting without (15) and (16) as LP, the LP budgeting with (15) but not (9) as LP(1), and the LP budgeting with both (15) and (16) but not (9) as LP(2).[8]

2) Comparison Between UD- and LP-Based Algorithms: We report the new experiments for both the UD- and LP-based algorithms in Tables VI and VII, respectively. As shown in column 9, all crosstalk constraints are still satisfied. Furthermore, the new full-chip routing optimization algorithms LP(1)+LR and LP(2)+LR become better than UD+LR in terms of the routing area after almost each and every phase. As shown in column 8, LP(1)+LR and LP(2)+LR reduce the area by up to 5.71% for bus structures and up to 4.57% for MCNC benchmarks compared to UD+LR. And there is no all-time winner between LP(1)+LR and LP(2)+LR.

To better appreciate our LP based algorithms, we further compare these algorithms to the uniform budgeting without LR.[9] Ignoring the limited crosstalk violations that may exist after Phase II, results shown in **bold** in columns 5 and 6 of Tables VI and VII represent the lower bounds for shields and area in a uniform budgeting solution with no crosstalk violation. Compared with the lower-bound results of uniform budgeting, LP(1)+LR and LP(2)+LR reduce the area by up to 9.78% for bus structures

---

[7]Note that rounding up the estimated number of shields to an integer will theoretically end up with an ILP problem. Such an ILP problem would be much less efficient compared with the LP problem.

[8]We also considered the LP budgeting with (16) but not (15). However, our experiment results showed that constraint (16) alone cannot prevent an unbalanced budgeting solution as discussed in Section IV-A-2, because constraint (16) alone only provides a localized guidance for budgeting within a region and lacks a global view.

[9]UD+LR starts with a uniform budgeting, but adjusts the budgeting because of the Pass II algorithm in LR. Therefore, the final solution from UD+LR is no longer uniform budgeting.

TABLE VI

COMPARISON OF THE TOTAL NUMBER OF SHIELDS, ROUTING AREAS IN $h_{\max}$, AND THE MAXIMUM/AVERAGE LSK VALUES UNDER UD+LR, LP(1)+LR AND LP(2)+LR FOR TWO 64-BIT BUS STRUCTURES. THE VALUES IN PARENTHESIS ARE LP(1)+LR AND LP(2)+LR'S ROUTING AREA REDUCTION OVER UD+LR'S IN PERCENTAGE. THE VALUES IN **bold** ARE THE RESULTS FOR UNIFORM BUDGETING WITHOUT LOCAL REFINEMENT

1	2	3	4	5	6	7	8	9
Sensitive Rate	Algorithm	Phase I		Phase II		Phase III		LSK
		Shield	h_{max}	Shield	h_{max}	Shield	h_{max}	1000
				bus.1				
45~61%	UD+LR	114.1	92.7 (0.00%)	**73**	**88 (0.00%)**	45	85 (0.00%)	950.0/553.9
	LP(1)+LR	128.0	80.0 (-13.70%)	98	83 (-5.68%)	62	82 (-3.53%)	998.5/624.0
	LP(2)+LR	128.0	80.0 (-13.70%)	81	83 (-5.68%)	58	82 (-3.53%)	980.2/623.8
67~77%	UD+LR	158.8	97.6 (0.00%)	**103**	**92 (0.00%)**	76	88 (0.00%)	981.8/701.5
	LP(1)+LR	158.8	83.4 (-14.55%)	195	89 (-3.26%)	176	87 (-1.14%)	977.7/355.3
	LP(2)+LR	158.8	83.4 (-14.55%)	123	85 (-7.61%)	96	83 (-5.68%)	980.1/705.0
				bus.2				
45~61%	UD+LR	114.1	108.7 (0.00%)	**72**	**104 (0.00%)**	48	101 (0.00%)	958.1/556.2
	LP(1)+LR	132.7	96.0 (-11.68%)	82	99 (-4.81%)	61	98 (-2.97%)	982.7/579.5
	LP(2)+LR	132.7	96.0 (-11.68%)	80	99 (-4.81%)	58	98 (-2.97%)	970.0/547.2
67~77%	UD+LR	158.8	113.6 (0.00%)	**104**	**108 (0.00%)**	79	105 (0.00%)	964.2/695.4
	LP(1)+LR	184.7	96.0 (-15.49%)	120	100 (-7.41%)	104	99 (-5.71%)	999.9/732.1
	LP(2)+LR	184.7	96.0 (-15.49%)	119	100 (-7.41%)	99	99 (-5.71%)	980.1/751.0

TABLE VII

COMPARISON OF THE TOTAL NUMBER OF SHIELDS, ROUTING AREAS IN $(h_{\max} + w_{\max})$, AND THE MAXIMUM/AVERAGE LSK VALUES UNDER UD+LR, LP(1)+LR AND LP(2)+LR FOR THREE MCNC BENCHMARK CIRCUITS. THE VALUES IN PARENTHESIS ARE LP(1)+LR AND LP(2)+LR'S ROUTING AREA REDUCTION OVER UD+LRs IN PERCENTAGE. THE VALUES IN **bold** ARE THE RESULTS FOR THE UNIFORM BUDGETING WITHOUT LOCAL REFINEMENT

1	2	3	4	5	6	7	8	9
Sensitive Rate	Algorithm	Phase I		Phase II		Phase III		LSK
		Shield	$h_{max} + w_{max}$	Shield	$h_{max} + w_{max}$	Shield	$h_{max} + w_{max}$	1000
				MCNC-1				
45~78%	UD+LR	108.6	230.4 + 147.5 (0.00%)	**168**	**232 + 151 (0.00%)**	98	220 + 149 (0.00%)	995.6/283.4
	LP(1)+LR	423.6	226.5 + 145.5 (-1.56%)	283	228 + 145 (-2.61%)	157	218 + 143 (-2.17%)	996.2/262.1
	LP(2)+LR	422.1	226.9 + 145.7 (-1.40%)	274	222 + 145 (-4.18%)	156	218 + 143 (-2.17%)	998.5/259.8
67~87%	UD+LR	159.0	236.8 + 149.6 (0.00%)	**244**	**237 + 154 (0.00%)**	161	228 + 150 (0.00%)	999.4/311.0
	LP(1)+LR	509.2	225.8 + 143.8 (-4.35%)	442	234 + 150 (-1.79%)	315	225 + 148 (-1.32%)	996.1/289.2
	LP(2)+LR	548.7	231.9 + 146.8 (-1.99%)	428	229 + 149 (-3.32%)	304	222 + 148 (-2.12%)	998.9/308.6
				MCNC-2				
44~79%	UD+LR	261.0	193.6 + 194.8 (0.00%)	**258**	**194 + 194 (0.00%)**	129	188 + 192 (0.00%)	995.5/269.9
	LP(1)+LR	682.4	190.7 + 194.8 (-0.75%)	392	182 + 192 (-3.61%)	167	182 + 185 (-3.42%)	997.0/268.9
	LP(2)+LR	688.9	190.7 + 194.8 (-0.75%)	378	182 + 191 (-3.87%)	170	182 + 185 (-3.42%)	997.5/266.5
66~87%	UD+LR	368.5	199.2 + 199.1 (0.00%)	**395**	**200 + 199 (0.00%)**	228	191 + 193 (0.00%)	998.4/310.8
	LP(1)+LR	878.7	190.1 + 193.2 (-3.77%)	709	187 + 198 (-3.51%)	397	183 + 185 (-4.17%)	997.3/270.8
	LP(2)+LR	879.0	190.6 + 193.3 (-3.62%)	692	182 + 196 (-5.26%)	406	183 + 187 (-3.65%)	999.0/272.5
				MCNC-3				
44~78%	UD+LR	2749.4	214.7 + 939.7 (0.00%)	**1894**	**210 + 895 (0.00%)**	656	200 + 885 (0.00%)	998.8/109.1
	LP(1)+LR	3740.8	206.2 + 911.7 (-3.16%)	2561	197 + 929 (1.90%)	857	195 + 854 (-3.32%)	999.4/75.0
	LP(2)+LR	3743.5	206.2 + 911.7 (-3.16%)	2553	195 + 929 (1.72%)	885	195 + 858 (-2.95%)	999.3/68.6
66~88%	UD+LR	3859.1	222.3 + 963.8 (0.00%)	**2872**	**217 + 920 (0.00%)**	1224	206 + 889 (0.00%)	999.7/254.8
	LP(1)+LR	5143.5	206.6 + 914.5 (-5.48%)	3895	201 + 951 (1.32%)	1720	195 + 850 (-4.57%)	999.6/176.9
	LP(2)+LR	5143.5	206.6 + 914.5 (-5.48%)	3900	199 + 951 (1.14%)	1719	195 + 850 (-4.57%)	999.9/174.2

and up to 8.09% for MCNC benchmarks (See comparisons between column 8 from LP(1)+LR and LP(2)+LR and column 6 from UD+LR in **bold**, respectively).

An interesting observation from Tables VI and VII is that although LP-based full-chip routing optimization algorithms consume far more shields than UD+LR, the routing areas are indeed smaller for LP(1)+LR and LP(2)+LR. These extra shields must belong to the noncritical routing regions. This observation indicates that LP-based budgeting schemes meet our expectation—reducing congestion in critical regions by allocating higher crosstalk bounds to the critical regions. This leads to more shields in the noncritical regions without increasing routing area as defined in Section III-B.3.

Moreover, LR is effective to reduce the total number of shields. As illustrated by MCNC-3 under an average logic sensitivity 50%, the physical sensitivity rate of which ranges from 44~78%, LR reduces shields from 1894 to 656 for UD+LR and from 2561 to 857 for LP(1)+LR (see comparison between columns 5 and 7). The relative reduction is up to 65.4% and 66.9%, respectively. As a by-product, LR also reduces the routing area. For the same experiment example, the area reduction is 1.8% for UD+LR and 8.31% for LP(1)+LR (see comparison between columns 6 and 8).

In order to examine the impact of shield insertion on the total routing area, we compare the routing areas of global routing without shield insertion (denoted as GR) with those of UD+LR,

TABLE VIII
COMPARISON OF THE ROUTING AREAS IN $(h_{\max} + w_{\max})$ BETWEEN
GR, UD+LR, LP(1)+LR, AND LP(2)+LR FOR THREE MCNC
BENCHMARK CIRCUITS. THE VALUES IN PARENTHESIS ARE
ROUTING AREA OVERHEAD OVER GR

Ckts	GR	UD+LR	LP(1)+LR	LP(2)+LR
MCNC-1	218+143	220+149(2.2%)	218+143(0.0%)	218+143(0.0%)
MCNC-2	182+185	188+192(3.5%)	182+185(0.0%)	182+185(0.0%)
MCNC-3	195+848	200+885(4.0%)	195+854(0.6%)	195+858(0.9%)

TABLE IX
RUNNING TIME (IN SECONDS) FOR UD AND LP(2) BUDGETING SCHEMES, AS
WELL AS THE TOTAL RUNNING TIME FOR FULL-CHIP ROUTING OPTIMIZATION
ALGORITHMS OF UD+LR AND LP(2)+LR, RESPECTIVELY

Sensitivity Rate	Test Ckts	UD+LR		LP(2)+LR	
		Budget	Total	Budget	Total
45~78%	MCNC-1	0.23	3002.26	10.54	2378.53
44~79%	MCNC-2	0.40	2709.08	5.75	2638.90
44~78%	MCNC-3	3.07	10878.41	140.06	11255.80
67~87%	MCNC-1	0.22	3010.56	31.36	2700.38
66~87%	MCNC-2	0.40	2935.31	12.04	2703.23
66~88%	MCNC-3	3.10	11648.60	113.56	13106.94

LP(1)+LR and LP(2)+LR in Table VIII under an average logic sensitivity rate 50%. According to Table VIII, shield insertion in UD+LR may cause up to 4.0% routing area overhead compared to GR, but LP(1)+LR and LP(2)+LR have much lower routing area overhead than UD+LR. For example, while UD+LR increases the routing area by 4.0% for MCNC–3, LP(1)+LR and LP(2)+LR increase the routing area only by 0.6% and 0.9%, respectively. Clearly, a good budgeting scheme is able to allocate higher noise budgets to more congested regions and, thus, keep the routing area overhead due to shield insertion to the minimum.

We report the running time for different RLC crosstalk budgeting schemes as well as the total running time including Phases II and III in Table IX. Only the running time for LP(2)+LR is reported as it is the slowest one among all LP-based algorithms. The running time is based on three MCNC benchmark circuits. As shown in Table IX, even though the LP budgeting consumes more time than UD, the total running time for LP(2)+LR is not necessary higher. In fact, LP(2)+LR has a much less runtime than UD+LR for the first two benchmark circuits. Because the runtime for budgeting is just a small fraction of the total runtime, we conclude that future work on runtime reduction should focus on SINO and LR, but not on budgeting.

V. CONCLUSION AND DISCUSSIONS

Existing layout optimization methods for RLC crosstalk reduction assume a set of interconnects with *a priori* given crosstalk bounds in a routing region. RLC crosstalk budgeting is critical for effectively applying these methods at the full-chip level. In this paper, we have formulated a full-chip routing optimization problem with RLC crosstalk budgeting, and solved this problem by a multiphase algorithm. Compared with uniform budgeting without LR, our full-chip routing optimization algorithm using LP budgeting can reduce the

routing area for bus structures and MCNC benchmarks by up to 9.78% and 8.09%, respectively. The uniform budgeting can be improved by LR and results in the best alternative (UD+LR) in this paper. Compared with UD+LR, our full-chip routing optimization algorithm using LP budgeting can use less runtime, and reduce the total routing area by up to 5.71% and 4.57% for bus structures and MCNC benchmarks, respectively.

An ILP-based noise budgeting has been developed for *RC* crosstalk in [11]. Compared with the ILP-based method, our LP-based approach is more efficient and able to handle larger design examples. Nevertheless, benchmarks presented in this paper are small compared with the gigascale integration. This is mainly because of the limitation of the LP engine, *lp-solver* [28], used in our current implementation. *lp-solver* is a public domain software, and may suffer numerical instability if the LP has more than tens of thousands of independent variables and constraints. In our budgeting formulation, the number of independent variables $(\overline{K_{it}})$ equals to the number of net segments (N_{it}), hence even for a small benchmark, the number of independent variables can easily go beyond tens of thousands. (For example, there are around 13 000 independent variables and 35 000 constraints for MCNC-3). Therefore, our current implementation is not able to handle very large design cases. For industrial real designs, we suggest that a more robust commercial LP engine, like CPLEX from [27], should be used.

In this paper, a very efficient LSK model has been developed to model the RLC crosstalk at the full-chip level. The proposed LSK model is developed based on the worst-case noise algorithm from [19]. Because [19] assumes that all parallel interconnects have the same routing directions, the same assumption is inherited by our LSK model implicitly. We will extend the worst-case noise algorithm and the LSK model to consider arbitrary routing directions. Note that although the proposed budgeting technique itself is not limited to two routing layers, the K_{eff} crosstalk model is developed under the assumption of two routing layers [7]. Therefore, our future work will develop more general crosstalk models considering multilayer routing structures and extend SINO algorithm, shield estimation, and noise budgeting, correspondingly. Shields are naturally a part of the P/G network. In this work, we only consider signal integrity, but not power integrity. In the future, we plan to also study the codesign of P/G nets (including shields) and signal nets, with consideration of both power and signal integrity and routing area/congestion reduction.

The proposed crosstalk budgeting is a postglobal routing optimization technique, and we assume that the global routing solution is fixed and cannot be changed during the budgeting procedures. To explore a larger design space, we are working on a global router that is able to simultaneously consider noise budgeting and shield insertion.

References

[1] K. L. Shepard and V. Narayanan, "Noise in deep submicron digital design," in *Proc. Int. Conf. Computer-Aided Design*, 1996, pp. 524–531.

[2] L. He, *Interconnect Modeling and Design With Consideration of On-Chip Inductance, Chapter 5 in Layout Optimization in VLSI Designs*, D. Z. Du and S. Sapatnekar, Eds. Norwell, MA: Kluwer, 2001.

[3] T. Gao and C. L. Liu, "Minimum crosstalk channel routing," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1993, pp. 692–696.

[4] D. A. Kirkpatrick and A. L. Sangiovanni-Vincentelli, "Techniques for crosstalk avoidance in the physical design of high-performance digital systems," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1994, pp. 616–619.

[5] K. Chaudhary, A. Onozawa, and E. S. Kuh, "A spacing algorithm for performance enhancement and cross-talk reduction," in *Proc. Int. Conf. Computer-Aided Design*, 1993, pp. 697–702.

[6] L. He, N. Chang, S. Lin, and O. S. Nakagawa, "An efficient inductance modeling for on-chip interconnects," in *Proc. CICC*, 1999, pp. 457–460.

[7] L. He and K. M. Lepak, "Simultaneous shielding insertion and net ordering for capacitive and inductive coupling minimization," in *Proc. Int. Symp. Phys. Design*, 2000, pp. 55–60.

[8] Y. Cao, C. M. Hu, X. Huang, A. B. Kahng, S. Muddu, D. Stroobandt, and D. Sylvester, "Effects of global interconnect optimizations on performance estimation of deep submicron design," in *Proc. Int. Conf. Computer-Aided Design*, 2000, pp. 56–61.

[9] G. Zhong, H. Wang, C.-K. Koh, and K. Roy, "A twisted bundle layout structure for minimizing inductive coupling noise," in *Proc. Int. Conf. Computer-Aided Design*, 2000, pp. 406–411.

[10] Y. Massoud, J. Kawa, D. MacMillen, and J. White, "Modeling and analysis of differential signaling for minimizing inductive cross-talk," in *Proc. Design Automation Conf.*, 2001, pp. 804–809.

[11] T. Xue, E. S. Kuh, and D. Wang, "Post global routing crosstalk risk estimation and reduction," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1996, pp. 302–309.

[12] P. Chen, Y. Kukimoto, C.-C. Teng, and K. Keutzer, "On convergence of switching windows computation in presence of crosstalk noise," in *Proc. Int. Symp. Phys. Design*, 2002, pp. 84–89.

[13] T. Xue and E. S. Kuh, "Post global routing crosstalk synthesis," *IEEE Trans. Computer-Aided Design*, vol. 16, pp. 1418–1430, Dec. 1997.

[14] H. Zhou and D. F. Wong, "Global routing with crosstalk constraints," *IEEE Trans. Computer-Aided Design*, vol. 18, pp. 1683–1688, Nov. 1999.

[15] K. M. Lepak, J. D. Ma, M. Xu, and L. He, "Simultaneous Shield Insertion and Net Ordering for Capacitive and Inductive Coupling Minimization," Univ. of Wisconsin, Madison, Tech. Rep. ECE-00-002, 2000.

[16] M. Kamon, M. Tsuk, and J. White, "Fasthenry: A multipole-accelerated 3D inductance extraction program," *IEEE Trans. MTT*, p. 1758, 1994.

[17] M. Xu and L. He, "An efficient model for frequency-based on-chip inductance," in *Proc. Great Lake Symp. VLSI*, 2001, pp. 115–120.

[18] J. Cong, L. He, A. B. Kahng, D. Noice, N. Shirali, and S. H.-C. Yen, "Analysis and justification of a simple, practical 2 1/2-D capacitance extraction methodology," in *Proc. Design Automation Conf.*, 1997, pp. 627–632.

[19] J. Chen and L. He, "Determination of worst-case crosstalk noise for nonswitching victims in GHz+ interconnects," in *ASP-DAC*, 2003, pp. 162–167.

[20] D. C. Montgomery, *Design and Analysis of Experiments*. New York: Wiley, 2001.

[21] G. Levine, *A Guide to SPSS for Analysis of Variance*. Hillsdale, N.J.: Lawrence Erlbaum, 1991.

[22] G. E. P. Box, W. G. Hunter, and J. S. Hunter, *Statistics for Experimenters: An Introduction to Design, Analysis, and Model Building*. New York: Wiley, 1978.

[23] J. E. Dennis, D. M. Gay, and R. E. Welsch, "An adaptive nonlinear least-squares algorithm," *ACM Trans. Math. Software*, pp. 348–368, 1981.

[24] P. H. Sherrod, NLREG 4.0: A nonlinear regression algorithm shareware, 1998.

[25] J. Cong and B. Preas, "A new algorithm for standard cell global routing," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1988, pp. 176–179.

[26] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*. New York: McGraw-Hill, 1996.

[27] ILOG CPLEX Optimizers *http://www.ilog.com/products/cplex/* [Online]

[28] lp-solver 3.2: A Public Domain (MI)LP Solver, M. Berkelaar *.ftp://ftp.ics.ele.tue.nl/pub/lp_solve/* [Online]

[29] M. Wang, X. Yang, and M. Sarrafzadeh, "DRAGON2000: Standard-cell placement tool for large industry circuits," in *Proc. Int. Conf. Computer-Aided Design*, 2000, pp. 260–263.

Jinjun Xiong received the B.E. degree (with honors) in precision measurement and control, the B.E. degree in industrial engineering, and the M.E. degree in microelectronic engineering from Tsinghua University, Beijing, China, in 1998, 1998, and 2000, respectively, and the M.S. degree in electrical and computer engineering in the University of Wisconsin, Madison, in 2001. He is currently pursuing the Ph.D. degree at the University of California, Los Angeles.

He was part-time with the Tsinghua Tongfang Corporation from 1998 to 2000. He was with the Mechanical and Industrial Engineering Department, University of Illinois, Urbana-Champaign, from August 2000 to August 2001. His current research interests include design automation for VLSI circuits and systems, large-scale optimization, and combinatorial mathematics.

Mr. Xiong received the Distinguished Graduate Fellowship from the University of Wisconsin, Madison, in 2001, and from the University of California, Los Angeles, in 2002.

Lei He (S'94–M'99) received the B.S. degree in electrical engineering from Fudan University, Shanghai, China, in 1990 and the Ph.D. degree in computer science from University of California, Los Angeles (UCLA), in 1999.

He is currently an Assistant Professor in the Electrical Engineering Department, UCLA. From 1999 to 2001, he was a Faculty Member at the University of Wisconsin, Madison. He held industrial positions with Cadence, Hewlett Packard, the Intel Corporation, and Synopsys. His research interests include computer-aided design of VLSI circuits and systems, interconnect modeling and design, programmable logic and interconnect, and power-efficient circuits and systems.

Prof. He received the Dimitris N. Chorafas Foundation Prize for Engineering and Technology in 1997, the Distinguished Ph.D. Award from the UCLA Henry Samueli School of Engineering and Applied Science in 2000, the NSF CAREER Award in 2000, the UCLA Chancellor's Faculty Development Award in 2003, and the IBM Faculty Award in 2003.