Full-chip Routing Optimization with RLC Crosstalk Budgeting*

Jinjun Xiong Lei He
EE Department, University of California at Los Angeles
Los Angeles, CA, 90095

Abstract

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 formulate a full-chip routing optimization problem with RLC crosstalk budgeting, and solve this problem with a multi-phase 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 to 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 constraints.

1 Introduction

As the clock frequency continues to increase and the minimum feature size keeps shrinking, signal integrity becomes one of the primary design constraints for high performance VLSI chip design [1]. Because RLC crosstalk gains a growing importance for GHz+ IC design [2], net ordering [3, 4] and spacing [5] under RC crosstalk model are no longer sufficient to reduce the long range inductive coupling. Several recent studies

*This research is partially supported by NSF CAREER Award CCR-0093273 and a grant from Intel Design Science and Technology Committee. Address comments to lhe@ee.ucla.edu.

on interconnect synthesis have considered RLC crosstalk reduction, utilizing uniform shielding [6], simultaneous shield insertion and net ordering [7], staggered buffers [8], twisted bundle layout structure [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 shielding 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 re-route 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 non-adjacent wires. Furthermore, the algorithm complexity is high as ILP is used.

In this paper, we study the full-chip routing optimization problem considering simultaneous shield insertion and net ordering (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 multi-phase 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 procedure to reduce the total number of shields.

The rest of the paper is organized as follows: Section 2 presents the background knowledge. Section 3 formulates the full-chip routing optimization problem, and presents our multi-phase routing optimization algorithm including LP-based crosstalk budgeting. Section 4 reports experiment results using MCNC benchmarks, and also presents further tuning of the crosstalk budgeting formulation. Section 5 concludes the paper with discussions of the future work.

2 Background

2.1 Preliminaries

To make the presentation simple, we assume two routing layers, one for horizontal wires and the other for vertical wires. The routing layers are divided by pre-routed 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 1.

According to [7], two signal nets N_i and N_j are *logically sensitive* (or in short, *sensitive*) to each other if, through logic synthesis or timing analysis [12], a switching event on N_j causes N_i to malfunction (due to extraordinary crosstalk or delay variation). In this case we call N_j an aggressor for N_i and N_i a victim of N_j. 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 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.

2.2 SINO Problem

Given a set of parallel interconnects with a uniform wire length, the SINO problem [7] finds a minimum area shield insertion and net ordering (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. It has been shown that the SINO problem is NP-hard, therefore a simulated annealing algorithm is developed to obtain a high-quality solution. The reader is suggested to learn details of the SINO formulation and algorithm in [7] for better understanding of the full-chip routing optimization to be presented.

Since 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 prior*. 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; (2) How to partition RLC crosstalk bounds specified at sinks into bounds for net segments so that

[1] This assumption is compatible with the Keff model to be presented in Section 2.3

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 region R_t
O_t	total number of tracks occupied by obstacles in region R_t
G_t	set of net segments in region R_t
$\|G_t\|$	total number of net segments in region R_t
S_t	set of shield segments in region 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 region 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} of net N_i
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

Table 1: Notations that are frequently used in this paper. They will be explained in detail when they are first used.

the area overhead due to SINO can be minimized.

We propose an efficient RLC crosstalk model to address the first problem in Section 2.3, 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 shielding estimation in Section 2.4 and budgeting formulation in Section 3.

2.3 LSK Model for RLC Crosstalk

It is proposed in [13] 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

$$K_{it,jt} = \frac{M_{it,jt}}{\sqrt{L_{it} \cdot L_{jt}}} \quad (1)$$

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 $K_{it,jt}$ [13]. When N_{it} and N_{jt} are in different "blocks" separated by shield segments, $K_{it,jt}$ = 0 or a small constant. When the two net segments are in the same block, we first consider the following two special cases: as shown in Figure 1 where 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}=n_{jt}, the mutual inductance is reduced to self inductance and $K_{it,it} = 1$ by definition; 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 a same current loop and should be 0 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

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

where $f(i,t) = \frac{(n_{it} - s_{lt})}{(n_{jt} - s_{lt})}$ and $g(j,t) = \frac{(s_{rt} - n_{jt})}{(s_{rt} - n_{it})}$ are both linear interpolation of 0 and 1.

The K model is reasonably accurate – within +20% to -10% error range compared to the three-dimensional field solver FastHenry [14] – and tends to be conservative. Further, 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

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

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. I.e, a signal net in a SINO solution with a higher K value given by

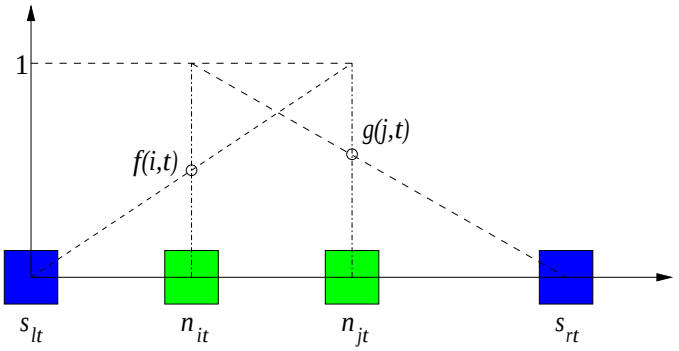

Figure 1: Illustration of $K_{it,jt}$ computation in a given block. n_{it} and n_{jt} are track ordering numbers for net segment N_{it} and N_{jt}, and s_{lt} and s_{rt} are track ordering numbers for the edge shield segments of the block. $f(i,t)$ and $g(j,t)$ are two linear interpolation functions 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.

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 in all routing regions, we propose a length-scaled K_{eff} (LSK) model, where the LSK value for a net N_i at its j^{th} 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 and K_{it} is the total coupling for N_{it}. This model can be justified by the following experiments. We randomly choose 4 SINO solutions, assign different wire lengths to all four SINO solutions, and generate the distributed RLC circuit models. SPICE simulation is then carried out to find the worst case noise according to the algorithm proposed in [15] for all SINO solutions with different wire lengths. From Figure 2, 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 [16, 11] to model the capacitive crosstalk by the product of the coupling length and the coupling capacitance. However, no SPICE simulation results were shown in [16, 11] to verify the linear relationship between the length and the RC crosstalk coupling.

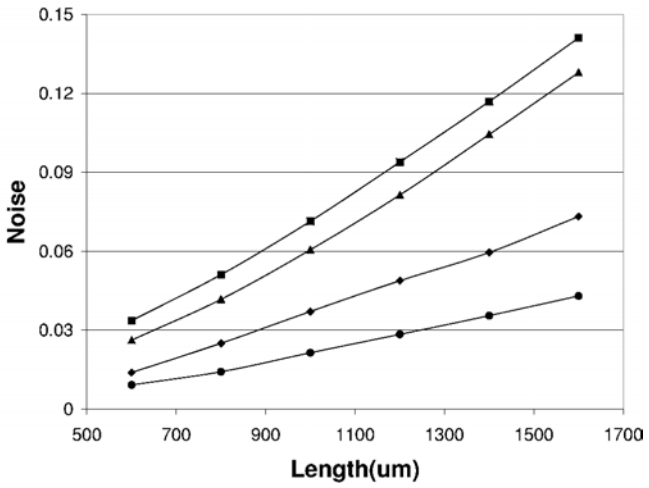

Figure 2: Linear scalability of SPICE computed worst case noise versus the length for four min-area SINO solutions.

2.4 Shield Estimation

In this paper, we consider the SINO formulation under K_{eff} model, i.e., the inductive noise bounds are given as $\overline{K_{it}}$. The following formula has been developed in [17] to estimate the number of shields in a min-area SINO solution within region R_t. The formula is given as:

$$\begin{aligned} |S_t| &= \frac{c_1}{|G_t|} \cdot \sum_{N_{it} \in G_t} \overline{K_{it}} \cdot \sum_{N_{it} \in G_t} r_{it}^2 + c_2 \cdot \sum_{N_{it} \in G_t} r_{it}^2 + \frac{c_3}{|G_t|^2} \cdot \sum_{N_{it} \in G_t} \overline{K_{it}} \cdot \sum_{N_{it} \in G_t} r_{it}^2 + \frac{c_4}{|G_t|} \cdot \sum_{N_{it} \in G_t} r_{it}^2 \\ &+ \frac{c_5}{|G_t|} \cdot \sum_{N_{it} \in G_t} \overline{K_{it}} \cdot \sum_{N_{it} \in G_t} r_{it} + c_6 \cdot \sum_{N_{it} \in G_t} r_{it} + \frac{c_7}{|G_t|^2} \cdot \sum_{N_{it} \in G_t} \overline{K_{it}} \cdot \sum_{N_{it} \in G_t} r_{it} + \frac{c_8}{|G_t|} \cdot \sum_{N_{it} \in G_t} r_{it} \\ &+ c_9 \cdot \sum_{N_{it} \in G_t} \overline{K_{it}} + c_{10} \cdot |G_t| + \frac{c_{11}}{|G_t|} \cdot \sum_{N_{it} \in G_t} \overline{K_{it}} + c_{12} \end{aligned} \quad (5)$$

where notations from Table 1 are used, and c_1 to c_{12} are constant coefficients.

We propose to use an estimation formula different from (5). The formula newly developed in this paper has the following form:

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

where a_1 and a_2 are constant coefficients. The reasons for adopting this new shield estimation formula will become clear after we show how to compute the constant coefficients in (5) and (6).

In order to obtain the coefficients in (5) and (6) for a given routing region, we generate $10,000$ routing solutions with different combinations of the number of net segments, sensitivity rate r_{it} (ranging from 20%

to 80%), and $\overline{K_{it}}$ values. After running SINO algorithm for all cases, the number of shields in each SINO solution is collected. The data are then evenly divided into five groups. A multi-variable curve-fitting process that minimizes the least square error [18, 19, 20] is employed to obtain the coefficients in (5) and (6) under different groups. As shown in Table 2, the values of the coefficient of determination (R^2) under

	I	II	III	IV	V	δ
c_1	-0.26806	-0.40766	-0.03988	-0.85798	-0.70735	0.72337
c_2	1.01418	1.59382	0.67660	2.74040	2.76210	0.54896
c_3	0.68241	-1.25205	0.78929	1.35848	0.85998	2.06415
c_4	-2.21778	2.57886	-1.59408	-3.09895	-2.92099	1.60659
c_5	0.20536	0.485102	-0.00088	1.05309	0.88727	0.84514
c_6	-0.60561	-1.63289	-0.49800	-3.22059	-3.49498	0.74809
c_7	-1.10653	-0.48731	-0.84777	-2.37823	-1.98626	0.58277
c_8	3.43140	2.53806	2.91279	6.80146	7.27214	0.49249
c_9	-0.061256	-0.16475	-0.03141	-0.36545	-0.31620	0.79443
c_{10}	0.21836	0.57123	0.29208	1.17813	1.35860	0.71619
c_{11}	0.11256	0.33975	0.06822	0.76353	0.65881	0.80852
c_{12}	-0.42311	-1.27649	-0.67789	-2.48691	-2.83667	0.0489
R^2	0.8048	0.8015	0.8382	0.8712	0.8959	-
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	-

Table 2: Coefficients for shield estimation equation (5) and (6). The last column δ are the absolute values of the standard deviation over 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.

all groups show a very nice goodness-of-fit for both (5) and (6) in terms of estimation accuracy [2]. However, the coefficients obtained from different groups are only consistent for a_1 and a_2 in (6), but varies drastically for c_1 to c_{12} in (5), as indicated by the smaller δ for a_1 and a_2, but larger δ for c_1 to c_{12}. The better consistency of a_1 and a_2 implies that (6) is more robust than (5). In addition, (6) is much simpler than (5). Another observation from Table 2 is that the coefficient of a_1 are negative across all groups. This property will ensure that all coefficients of $\overline{K_{it}}$ in (6) are negative as well. The physical meaning of this property is that increasing crosstalk bound will reduce the number of shields needed for a min-area SINO solution. Therefore, ensuring the negative sign of $\overline{K_{it}}$'s coefficient is very important to maintain shield estimation

[2]The higher the coefficient of determination (R^2), the better the goodness-of-fit.

(6) physically correct. In contrast, it is very difficult to maintain such physical meaning for (5) due to the divergence of the coefficients. Because of these nice properties, (6) instead of (5) is employed in this paper. The coefficients finally used in this paper are obtained by merging all data from the above five groups together and re-doing the curve-fitting process, leading to $a_1 = -0.10491$, $a_2 = 0.49392$ with $R^2 = 0.87$.

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 coupling bounds:

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

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.

3 Problem Formulations and Algorithms

3.1 Full-chip Routing Optimization Problem

Formulation 1 (Full-chip routing optimization) *Given a global routing solution and the RLC crosstalk bound for each sink, 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.*

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.

Figure 3: Overview of the three-phase algorithm for full-chip routing optimization.

The full-chip routing optimization problem has a high complexity, as its sub-problem to find a SINO solution within a region is already NP-hard [7]. Therefore, we develop the following three-phase heuristic algorithm (see Figure 3). In Phase I we find the crosstalk bound for each net segment in all regions, which we call 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 of the coupling bound for each net segment (in K value)

is the input to the Phase II algorithm. In Phase II we perform SINO in each region by using the algorithm developed in [7]. In Phase III, we carry out a local refinement (LR) procedure to completely eliminate the remaining (but very limited) RLC crosstalk violations and further reduce the number of shields. Below we discuss Phase I and Phase III algorithms in detail.

3.2 RLC Crosstalk Budgeting

3.2.1 Common Constraints

In this work, we present an *optimal RLC crosstalk budgeting* scheme based on *linear programming* (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. Figure 4 illustrates this scenario where a victim (V) has three aggressors (A) in a region of 7 tracks. The victim N_{it}'s K_{it} value obtained in this case is $\overline{K_{it}^{max}}$.

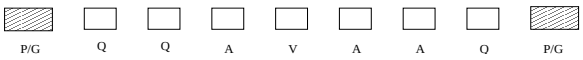

Figure 4: Illustration on how to obtain the maximum $\overline{K_{it}^{max}}$ for a victim net segment N_{it} (V) with three aggressors (A) and three quite signals (Q) or empty tracks for a given routing region R_t. The leftmost and the rightmost are power/ground networks.

The three constraints can be expressed formally as follows:

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

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

$$\overline{K_{it}} \leq \overline{K_{it}^{max}} \qquad \forall N_{it} \in R_t \; and \; \forall R_t \in \mathcal{R} \tag{10}$$

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

3.2.2 One-Dimension Optimal Budgeting

Without loss of generality, we call the global routing within a row of regions that allow only horizontal wires as one-dimension routing. We also call the number of tracks occupied by net segments, shields and/or

obstacles in region R_t as its height h_t, and the maximum height h_{max} among all regions as the height of the routing solution. Further, let *critical regions* be routing regions that define h_{max}. We then formulate the following 1D problem:

Formulation 2 (One-dimension crosstalk budgeting (1D) problem) *For a given one-dimension routing solution, the 1D problem partitions crosstalk bounds among regions such that the maximum height h_{max} is minimized.*

The 1D problem can be mathematically stated as:

minimize h_{max}

s.t.

$$\sum_{N_{it} \in G_t} \alpha_t \cdot \overline{K_{it}} + \beta_t + |G_t| + O_t \leq h_{max} \quad \forall R_t \in \mathcal{R} \tag{11}$$

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

3.2.3 Pseudo Two-Dimension Optimal Budgeting

For a two-dimensional 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 CLM's (ROW's). Then, 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. Same as 1D problem, *critical regions* are the routing regions that define h_{max} or w_{max}. The pseudo two-dimension optimal budgeting (2D-p) problem is defined as follows:

Formulation 3 (Pseudo two-dimension crosstalk budgeting (2D-p) problem) *For a given global routing solution, the 2D-p 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 2D-p problem can be mathematically stated as:

minimize $\gamma \cdot h_{max} + \theta \cdot w_{max}$

s.t.

$$\sum_{R_t \in CLM} (\sum_{N_{it} \in G_t} \alpha_t \cdot \overline{K_{it}} + \beta_t + |G_t| + O_t) \leq h_{max} \quad \forall CLM \in \mathcal{CLM} \tag{12}$$

$$\sum_{R_t \in ROW} (\sum_{N_{it} \in G_t} \alpha_t \cdot \overline{K_{it}} + \beta_t + |G_t| + O_t) \leq w_{max} \quad \forall ROW \in \mathcal{ROW} \tag{13}$$

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 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[3].

3.2.4 Main Theorem

According to the RLC crosstalk budgeting formulations, we have the following theorem.

Theorem 1 *Both 1D and 2D-p problems are linear programming (LP) problems.*

Sketch of proof: It is easy to verify that all constraints (8) – (13) are linear, and the objective functions of 1D and 2D-p are linear too, hence both 1D and 2D-p are linear programming problems. □

There are many very mature and robust linear programming solvers available from both the shelf and the academia. In order to solve our crosstalk budgeting problem, we can utilize any of these available solvers. We use LP to represent either 1D or 2D-p whenever there is no ambiguity for the rest of the paper.

3.3 Local Refinement

As shown in Figure 5, Phase III contains two passes of local refinement to eliminate crosstalk violations (denoted as LR-I) and reduce the number of shields (LR-II). 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. 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

[3]Minimizing the product of h_{max} and w_{max} is a quadric programming problem, but minimizing the sum is a linear programming problem. Furthermore, minimizing h_{max} and w_{max} is closely related to minimizing routing congestion in *critical regions*.

violation at each sink is indicated by a negative 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 allowed during the simulated annealing process. Inserting a shield in R_t may reduce K_{it} as well as K_{jt} for any other segment N_{jt} in R_t, hence we need to update the LSK slacks for all nets passing R_t. The iteration is stopped when there is no crosstalk violation for every net.

Pass two of the local refinement algorithm (LR-II) reduces the total shield number by exploiting the remaining LSK slacks to remove unneeded 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 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. Since removing a shield may increase some net segments' K values, we must check if these increments can be compensated by their LSK slacks respectively. If the answer is yes, then no crosstalk violation occurs, 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 is stopped when removing shields is no longer possible in any region.

4 Experiment Results and Algorithm Tuning

We have implemented our algorithm in C/C++ on UNIX/Linux platforms. A simplex based LP engine, *lp-solver* ([21]) is used to solve the LP-based budgeting in Phase I. We present experiments using two-pin bus structures and MCNC benchmark circuits. The MCNC benchmark circuits are placed by DRAGON [22], and routed by our own router implementing the iterative deletion algorithm [23]. In all experiments, we assume that buffers are inserted so that no wires are longer than 1000 μm, and consider average logic sensitivity rates of 50% and 70% over the chip. We also assume that all sinks have the same bound $\overline{LSK} = 1000$, but our algorithm and implementation can handle non-uniform bounds. Further, we randomly generate obstacles in each region. We summarize the test circuit characteristics in Table 3. In the following, we present initial experiment results in Section 4.1, and discuss the tuning of budgeting formulation and improved experiment results in Section 4.2.

LR-I: Eliminate crosstalk violations

While(there has crosstalk violation)

{

Find net N_i whose j^{th} sink p_{ij} has 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 a shield 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;

}

Figure 5: Phase III local refinement algorithm.

4.1 Initial Experiment Results and Discussions

4.1.1 Initial Comparison between 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 uniform budgeting (UD) scheme, which initially 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

Test circuit	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 circuit 1	607	1835	8×16	460
MCNC circuit 2	677	2155	16×16	643
MCNC circuit 3	814	2713	128×16	5758

Table 3: Test circuit characteristics.

budget:

$$\overline{K_{it}} = \frac{\overline{LSK_{ij}}}{len_{ij}} \tag{14}$$

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 must point out that the uniform distribution is for an individual net only; different nets will have different values according to their own crosstalk budgets. UD is not able to consider the non-uniform routing congestion among different regions.

For complete and fair comparison of our full-chip routing optimization, Phase 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; 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.

According to column 9 in Tables 4 and 5, the maximum/average 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. Further, when the sensitivity rate and obstacles are increased, the routing area and number of shields are increased for both algorithms. Same as the objective in the 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 4 and 5, 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 same as expected, and indicate the correctness of our problem formulation and program implementation.

Further, we compare Phase II and III results for UD+LR and LP+LR. From Table 4, 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 necessary better than UD+LR after Phases II and III. For MCNC circuits, Table 5 shows that LP+LR performs worse than UD+LR after Phase II and Phase III. We discuss below why LP+LR may be worse and

present the improved LP formulation in order for LP+LR to perform better than UD+LR.

1	2	3	4	5	6	7	8	9
Sensitive Rate	Algorithm	Phase I		Phase II		Phase III		$\overline{LSK}$
		Shield	h_{max}	Shield	h_{max}	Shield	h_{max}	1000
bus.1								
50%	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
70%	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								
50%	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
70%	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 4: 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 64-bit bus structures. The values in parenthesis are LP+LR's routing area reduction over UD+LR's in percentage.

4.1.2 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[4]. Second, the formula (7) only reflects the total effects of all net segments in a given region, therefore it 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 optimal budgeting solution. Because of this discrepancy between what our estimation formula can provide and what our LP+LR formulation requires, LP 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 in a given routing region, all net segments have the same sensitivity rate and the

[4]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 to our LP problem formulation.

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 Circuit 1								
50%	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
70%	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 Circuit 2								
50%	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
70%	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 Circuit 3								
50%	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
70%	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

Table 5: 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 MCNC benchmark circuits. The values in parenthesis are LP+LR's routing area reduction over UD+LR's in percentage.

sum of $\overline{K_{it}}$ over all net segments is fixed as $\overline{K_t^{sum}}$. In this case, evenly distributing $\overline{K_t^{sum}}$ among all net segments or giving $\overline{K_t^{max}}$ to only one net segment does not make difference in terms of our LP solution, but it may make difference in reality. For example, if a net segment has a high coupling bound and the rest segments 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 a more balanced coupling bounds for all net segments may have fewer shields.

4.2 Improved Budgeting Formulation and Experiment Results

4.2.1 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 with the precise relationship between each individual crosstalk bound and each individual net segment, 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 an LP problem that is more tractable.

There exist many possible heuristic constraints capable of improving the LP-based budgeting. The following two constraints are used in this paper due to their simplicity and linearity. The first one is the universal upper bound given by

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

where a_1 and a_2 are constant coefficients in formula (6). The universal upper bound prevents a budgeting scheme from favoring one net segment too much, which could result in a very unbalanced budgeting solution. An extreme case is already illustrated by the example in Section 4.1.2. Actually, (15) can 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}, since N_{it} is likely to suffer higher RLC crosstalk. I.e.,

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

Theoretically, the above constraint is valid only if we ignore the congestion differences between different routing regions. However, it leads to nice results in our experiments with the presence of nonuniform congestion distribution. Since (15) and (16) are linear, our 1D and 2D-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).

4.2.2 Comparison between UD and LP based algorithms

We compare the UD and LP based algorithms in Table 6 and Table 7. 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 almost after 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 when compared to UD+LR. However, 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 local refinement [5]. Ignoring the limited crosstalk violations that may exist after Phase II,

[5] UD+LR starts with a uniform budgeting, but adjusts the budgeting by Pass II in local refinement. Therefore, the final solution from UD+LR does not use uniform budgeting.

results shown in **bold** in columns 5 and 6 of Table 6 and 7 represent lower bounds of shields and area for a uniform budgeting that has no crosstalk violation. Compared to 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 and up to 8.09% for MCNC benchmarks.

An interesting observation from Tables 6 and 7 is that even though LP-based full-chip routing optimization algorithms consume far more shields than UD+LR, the routing areas for them are not necessary larger. Indeed, the routing areas are even smaller for LP(1)+LR and LP(2)+LR. These extra shields must belong to the non-critical 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 non-critical regions without increasing routing area defined in Section 3.2.3.

1	2	3	4	5	6	7	8	9
Sensitive Rate	Algorithm	Phase I		Phase II		Phase III		$\overline{LSK}$
		Shield	h_{max}	Shield	h_{max}	Shield	h_{max}	1000
bus.1								
50%	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
70%	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								
50%	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
70%	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 6: 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 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 erefinement.

Moreover, local refinement is effective to reduce the total number of shields. As illustrated by *MCNC*

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 Circuit 1								
50%	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
70%	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 Circuit 2								
50%	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
70%	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 Circuit 3								
50%	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
70%	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

Table 7: 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 MCNC benchmark circuits. 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 the uniform budgeting without local refinement.

circuits 3 under an average logic sensitivity 50%, local refinment 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, local refinement 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 column 6 and column 8).

We report the running time for different RLC crosstalk budgeting schemes as well as the total running time including Phase II and Phase III in Table 8. Only the running time for LP(2)+LR is reported as it is slowest among all LP-based algorithms. The running time is based on the three MCNC benchmark circuits. As shown in Table 8, 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. Since the runtime for budgeting is just a small fraction of the total runtime, we conclude that the future work on runtime reduction should focus on SINO and local refinement, instead of budgeting.

Sensitivity Rate	Test Circuit	UD+LR		LP(2)+LR	
		Budget	Total	Budget	Total
50%	MCNC Circuit 1	0.23	3002.26	10.54	2378.53
	MCNC Circuit 2	0.40	2709.08	5.75	2638.90
	MCNC Circuit 3	3.07	10878.41	140.06	11255.80
70%	MCNC Circuit 1	0.22	3010.56	31.36	2700.38
	MCNC Circuit 2	0.40	2935.31	12.04	2703.23
	MCNC Circuit 3	3.10	11648.60	113.56	13106.94

Table 8: 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.

5 Conclusions and Future Work

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 multi-phase algorithm. In phase I, we solve an optimal crosstalk budgeting based on linear programming (LP) 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 further reduce the total number of shields. Compared to uniform budgeting without local refinement, 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 local refinement and results in the best alternative (UD+LR) in this paper. Compared to 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.

Shields are naturally a part of the power/ground (P/G) network. We plan to study co-design of P/G nets (including shielding) and signal nets, with consideration of both power and signal integrity and routing area/congestion reduction.

References

[1] K. L. Shepard and V. Narayanan, "Noise in deep submicron digital design," in *ICCAD*, 1996.

[2] L. He, *Interconnect Modeling and Design with Consideration of On-Chip Inductance, Chapter 5 in Layout Optimization in VLSI Designs, edited by D.Z. Du and S. Sapatnekar*. Kluwer Academic Publishers, 2001.

[3] T. Gao and C. L. Liu, "Minimum crosstalk channel routing," in *ICCAD*, pp. 692–696, Nov. 1993.

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

[5] K. Chaudhary, A. Onozawa, and E. S. Kuh, "A spacing algorithm for performance enhancement and cross-talk reduction," in *ICCAD*, 1993.

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

[7] L. He and K. M. Lepak, "Simultaneous shielding insertion and net ordering for capacitive and inductive coupling minimization," in *ISPD*, 2000.

[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. on Computer Aided Design*, 2000.

[9] G. Zhong, H. Wang, C.-K. Koh, and K. Roy, "A twisted bundle layout structure for minimizing inductive coupling noise," in *ICCAD*, 2000.

[10] Y. Massoud, J. Kawa, D. MacMillen, and J. White, "Modeling and analysis of diferential signaling for minimizing inductive cross-talk," in *DAC*, 2001.

[11] T. Xue, E. S. Kuh, and D. Wang, "Post global routing crosstalk risk estimation and reduction," in *ICCAD*, pp. 302–309, Nov. 1996.

[12] P. Chen, Y. Kukimoto, C.-C. Teng, and K. Keutzer, "On convergence of switching windows computation in presence of crosstalk noise," in *ISPD*, 2002.

[13] K. M. Lepak, J. D. Ma, M. Xu, and L. He, "Simultaneous shield insertion and net ordering for capacitive and inductive coupling minimization," in *Univerisity of Wisconsin, Technical Report, ECE-00-002*, 2000.

[14] M. Kamon, M. Tsuk, and J. White, "Fasthenry: a multipole-accelerated 3d inductance extraction program," *IEEE Trans. on MTT*, 1994.

[15] J. Chen and L. He, "Determination of worst-case crosstalk noise for non-switching victims in GHz+ interconnects," in *TAU*, 2002.

[16] H. Zhou and D. F. Wong, "Global routing with crosstalk constraints," *TCAD*, November 1999.

[17] J. D. Ma, A. Parihar, and L. He, "Pre-routing estimation of shielding for RLC signal integrity," in *ICCD*, 2001.

[18] G. E. P. Box, W. G. Hunter., and J. S. Hunter, *Statistics for experimenters: an introduction to design, analysis, and model building*. John Wiley & Sons, New York, 1978.

[19] J. E. Dennis, D. M. Gay, and R. E. Welsch, "An adaptive non-linear least-squares algorithm," *ACM Trans. on Mathematical Software*, September 1981.

[20] P. H. Sherrod, "NLREG 4.0: a non-linear regression algorithm shareware," 1998.

[21] M. Berkelaar, “lp-solver 3.2: a public domain (MI)LP solver,” in *ftp://ftp.ics.ele.tue.nl/pub/lp_solve/*.

[22] M. Wang, X. Yang, and M. Sarrafzadeh, “DRAGON2000: Standard-cell placement tool for large industry circuits,” in *ICCAD*, 2000.

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