Simultaneous Shield Insertion and Net Ordering for Capacitive and Inductive Coupling Minimization

Kevin M. Lepak, James D. Z. Ma, Min Xu, and Lei He
Electrical and Computer Engineering Department,
University of Wisconsin,
Madison, WI 53706

In this paper, we first show that existing net ordering formulations to minimize noise are no longer sufficient with the presence of inductive noise, and shield insertion is needed to minimize inductive noise. Using a K_{eff} model as the figure of merit for inductive coupling, we then formulate two simultaneous shield insertion and net ordering (SINO) problems: the optimal SINO/NF problem to find a minimal area SINO solution that is free of capacitive and inductive noise, and the optimal SINO/NB problem to find a minimal area SINO solution that is free of capacitive noise and is under the given inductive noise bound. We reveal that both optimal SINO problems are NP-hard, and propose effective approximate algorithms for the two problems. Experiments show that our SINO/NB algorithm uses from 51% to 82% fewer shields compared to uniform shield insertion and net ordering (US+NO), and uses from 4% to 47% fewer shields compared to separated net ordering and shield insertion (NO+SI). Furthermore, the SINO/NB solutions under practical noise bounds use from 38% to 61% fewer shields compared to SINO/NF solutions, and use up to 36% fewer shields compared to the theoretical lower bound for optimal SINO/NF solutions. Moreover, we show that the K_{eff} model has a high fidelity versus the noise voltage computed using accurate RLC circuit models and SPICE simulations. To the best of our knowledge, it is the first work that presents an in-depth study on the automatic layout optimization of multiple nets to minimize both capacitive and inductive noise.

Key Words: VLSI physical design automation, signal integrity, noise minimization, shielding, net ordering, and on-chip inductance

1. INTRODUCTION

In deep sub-micron (DSM) designs, the wire thickness is often larger than the wire width, and the spacing between adjacent wires is often smaller than the distance between adjacent metal layers. This makes the coupling capacitance (herein referred to as Cx) between adjacent wires on the same layer larger than the ground capacitance (the sum of area and fringe capacitance), and in turn makes the coupling noise between adjacent wires a concern for DSM designs. Because the coupling capacitance between non-adjacent wires is negligible, we may permute the net ordering (or track assignment) so that sensitive nets[1] are not adjacent in order to

[1]See section 2.1.2 for the definition of net sensitivity.

This research was partially supported by NSF CAREER Award CCR-0093273 and a grant from Intel Design Science and Technology Committee. We used computers donated by SUN Microsystems and HP. Lei He is now also Assistant Professor of Electrical Engineering Department at UCLA. Address comments to lhe@ee.ucla.edu.

reduce the impact of coupling noise. The net ordering (track assignment) problem has been studied in [Gao and Liu 1993], [Gao and Liu 1994], [Xue and Kuh 1997], [Yim and Kyung 1999], [Kay and Rutenbar 2000], and [Chang and Cong 2000], under the assumption that coupling is determined only by directly adjacent nets. However, this assumption is no longer true if we consider coupling inductance. This has been illustrated by the experiment in [He et al. 1999], where the coupling of a 18-bit bus is computed by SPICE simulations using the interconnect RLC model. It was assumed that all signal wires in the bus are switching simultaneously, except that the two central wires are quiet victims. As shown in Table 1, when there are no shielding wires (in short, shields), the noise in the quiet victims is 0.71V. When two or five shields are inserted uniformly, the noise is drastically reduced to 0.38V and 0.17V, respectively. In either case, the shields are not adjacent to the victims and therefore do not change the Cx coupling for the victims. Hence, the noise reduction is due to reducing the inductive coupling (i.e., Lx coupling) that depends on both adjacent and non-adjacent nets by shield insertion. It is assumed in [He et al. 1999] that all nets are sensitive to one another, and the uniform shield insertion scheme is used without net ordering. As shown later in this paper, uniform shield insertion may lead to larger routing area compared to simultaneous shield insertion and net ordering (SINO) in the general case where not all nets are sensitive to one another.

# of Shields	Noise (% of Vdd)
0	0.71V (55%)
2	0.38V (29%)
5	0.17V (13%)

Table 1. SPICE-computed noise for victims with uniform shield insertion in [He et al. 1999] .

This paper studies the SINO problems for capacitive and inductive coupling minimization. The rest of this paper is organized as follows: Section 2 introduces the formula-based K model, the K_{eff} model based on it, and formulates the SINO/NF and SINO/NB problems. Section 3 presents properties and algorithms for the two SINO problems. Section 4 details the experimental settings and compares results obtained by different problem formulations and algorithms. Section 5 studies fidelity of the K_{eff} model to the SPICE-computed noise voltage for SINO solutions. Section 6 concludes the paper.

2. PROBLEM FORMULATIONS

2.1 Preliminaries

2.1.1 *Coplanar Interconnect Structures.* Throughout this work, we consider only parallel coplanar interconnect structures with all wires having the same length. These are characterized as a number of signal wires and power/ground wires which run parallel in the same layer. We give an example of this structure in Figure 1. In the figure, P and G represent the power and ground grids (P/G grids), s represents signal wires (denoted as *s-wire*), and g is a *shield* that often has similar width as an s-wire and is connected to P/G grids. Both P/G grids and shields provide dedicated current return paths for signals, and are denoted as *g-wire*'s in this paper. We use the terms "wire" and "net" interchangeably.

Fig. 1. A cross-sectional view of a coplanar interconnect structure with a shield inserted.

An interconnect structure can be represented by a string, where each symbol stands for an s-wire or a g-wire. For example, the interconnect structure in Figure 1 can be represented by $gssgssg$ if we do not distinguish these s-wires (or alternatively, g2sg2sg). If we label the s-wires from left to right as s_1, s_2, s_3 and s_4, then the string $gs_1s_2gs_3s_4g$ is a unique representation of a net ordering and shield insertion solution (referred to as a SINO solution or a SINO string). In this paper, a SINO string implicitly includes shields (the power and ground grids) as its first and last elements. These P/G grids are shield resources, but are not considered explicitly in solution size computations as they are present generally, with or without noise considerations. The "size" of a SINO solution can be determined directly from the length of the SINO string. As an example, consider the following: $< g > s_1s_3gs_2s_4s_5gs_7s_6s_0 < g >$. This string represents an eight (signal) bit interconnect structure with two g-wires (plus two implicit g-wires for the P/G grids denoted as $< g >$). Note that we can apply our formulations and algorithms to be presented to any group of wires which may contain pre-routed g-wires more than just a pair of P/G grids. We call the group of wires sandwiched between adjacent g-wires a *block*, and the number of s-wires in a block as the block size. A block can be represented as a substring of a SINO string (i.e. the first block of the above string would be written as s_1s_3). As in the original SINO string, the g-wires on each end are implicit with the substring.

2.1.2 *Net Sensitivity.* We define two nets s_1 and s_2 to be *sensitive* to each other if a switching signal on s_1 will cause s_2 to malfunction (due to extraordinary crosstalk or delay variation) or vice-versa. Further, we assume that two non-sensitive nets do *not* switch simultaneously. Net sensitivity is depicted graphically in Figure 2.

Fig. 2. Illustration of net sensitivity. The y-axis indicates signal (voltage) level and the x-axis indicates time. The switching event on the aggressor induces a noise voltage in the two victims as shown. For $victim_1$, the noise pulse occurs within its sampling window–hence the aggressor and $victim_1$ are sensitive. For $victim_2$, the noise pulse does not occur during its sampling window–hence the aggressor and $victim_2$ are not sensitive.

The sensitivity for all s-wires in a given problem can be represented compactly with a sensitivity matrix S of size $n \times n$, where n is the number of s-wires and an entry S_{ij} of 1 or 0 in location (i, j) indicates that s_i and s_j are sensitive or not sensitive, respectively, to one another. By definition, the matrix must be symmetric (i.e. $S_{ij} = S_{ji}$). For all formulations, we assume that an appropriate sensitivity matrix indicating design parameters and net relationship semantics is given a priori.

2.1.3 *Characteristics of Inductive Coupling.* For the formulation of the SINO problems, we assume that current will return from shields and therefore we assume a loop inductance model.[2] Under such a model, we illustrate the characteristics of inductive coupling in Figure 3, where we show the mutual inductance from the leftmost s-wire to all other s-wires. Cases (a) and (b) in the figure show two interconnect structures, $g18sg$ and $g6sg6sg6sg$, respectively. As shown by (a) where there is no shield between s-wires, the mutual inductance decreases slowly from left to right. This explains why net ordering is not effective to reduce inductive coupling. In comparing (a) with (b) in the figure, the mutual inductance between wires separated by shields becomes much smaller compared with coupling to wires within the same block. Therefore, shields are effective to reduce inductive coupling.

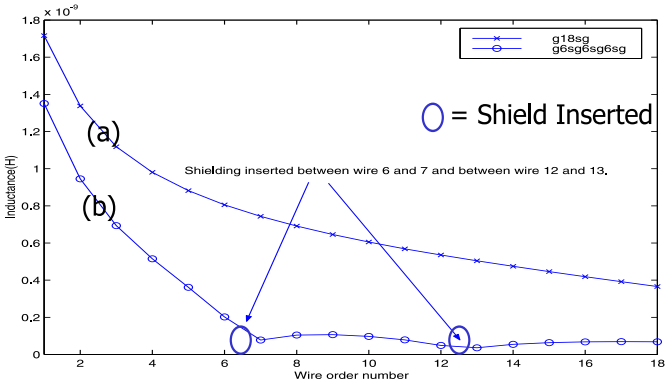

Fig. 3. Illustration of mutual inductive coupling from a signal wire (s_1) to other signal traces for two coplanar structures. Structures (a) and (b) show coupling from s_1 to all other s-wires as a function of wire-order number for the g18sg and g6sg6sg6sg structures, respectively.

In this paper we use the coupling coefficient between two s-wires to characterize the inductive coupling effect between them. The coefficient is defined as

$$K_{ij} = \frac{m_{ij}}{\sqrt{l_i \cdot l_j}} \tag{1}$$

where m_{ij} is the mutual inductance between s_i and s_j, and l_i and l_j is self inductance for s_i and s_j, respectively.

[2]However, we use the Partial Equivalence Circuit (PEEC) model [Ruehli 1974; He et al. 1999] and SPICE simulations to verify our modeling and problem formulation in section 5. The PEEC model does not assume any current return path, and is widely regarded as an accurate circuit model.

Computing the coupling coefficient (K_{ij}) efficiently is of paramount importance in our coupling minimization techniques. Therefore, we need to understand the variance of the coupling coefficient under differing technology parameters to efficiently approximate it. To explore this variance, we present K_{ij}'s of the same $g(6sg)^2 6sg$ (i.e., $g6sg6sg6sg$) structure under varying technology parameters of width (w), thickness (t), length (l), spacing of g-wires and s-wires (d), as well as frequency (f). The resulting values for K_{ij}'s are computed by the three-dimensional field solver FastHenry [Kamon et al. 1994] and are presented in Table 2.

We use the following parameters for high performance global interconnects as the baseline case: $w = 0.8\mu m$, $d = 0.8\mu m$, $t = 2.0\mu m$, l=2000μm, and f = 30GHz[3] (see row 1 in the table). When we change the wire length from 2000μm to 1000μm (comparing rows 1 and 2 in the table), the maximum difference for K_{ij} is only 3.0%. When we change the wire spacing from 0.8μm to 1.6μm (comparing rows 1 and 3 in the table), the maximum difference for K_{ij} is only 3.0%. When we change the wire width from 0.8μm to 1.6μm (comparing rows 1 and 4 in the table), the maximum difference for K_{ij} is only 2.6%. When we change the wire thickness from 2.0μm to 1.0μm (comparing rows 1 and 5 in the table), the maximum difference for K_{ij} is only 3.0%. When we change the frequency from 30GHz to 50GHz (comparing rows 1 and 6 in the table), the maximum difference for K_{ij} is only 4.9%. This data shows that the coupling coefficient is relatively independent of technology parameters, making it a desirable metric for measuring inductive coupling. In Section 5 we will further explore the correlation of the weighted sum of K_{ij} (defined as K_{eff} model in section 2.1.5) to actual observed noise voltage.

w, d, t, l, f	K_{12}	K_{13}	K_{14}	K_{15}	K_{16}
0.8, 0.8, 2.0, 2000, 30	0.71(0%)	0.53(0%)	0.41(0%)	0.31(0%)	0.21(0%)
0.8, 0.8, 2.0, 1000, 30	0.72(1.4%)	0.54(1.8%)	0.42(2.3%)	0.32(2.7%)	0.22(3.0%)
0.8, 1.6, 2.0, 2000, 30	0.71(0.0%)	0.52(1.8%)	0.41(0.0%)	0.30(2.7%)	0.20(3.0%)
1.6, 0.8, 2.0, 2000, 30	0.71(0.4%)	0.54(1.9%)	0.42(2.3%)	0.32(2.3%)	0.21(2.6%)
0.8, 0.8, 1.0, 2000, 30	0.73(2.8%)	0.54(1.8%)	0.42(2.3%)	0.32(2.7%)	0.22(3.0%)
0.8, 0.8, 2.0, 2000, 50	0.74(4.2%)	0.55(3.8%)	0.43(4.9%)	0.32(2.3%)	0.22(3.0%)

Table 2. Coupling coefficients between one s-wire and other s-wires inside a block for the $g(6sg)^2 6sg$ structure.

2.1.4 *Modeling of Inductive Coupling Coefficient.* We use formula-based K model to compute the inductive coupling coefficient between two s-wires. When s-wires i and j are in different blocks, the coupling coefficient $K_{ij} = 0$ or a small constant. When the two s-wires are in the same block, as shown in Figure 4 where N_i and N_j are track ordering numbers for the two s-wires, and g_l and g_r are track ordering numbers for the two edge g-wires, we consider the following cases for the coupling coefficient computation: when $i = j$, the mutual inductance is reduced to self inductance and $K_{ii} = 1$ by definition; when N_i (or N_j) becomes g_l (or g_r), because we assume the loop inductance model with current returning from shields, inductive coupling K_{ij} is between the two segments of the same current loop and is 0 under

[3]30GHz here is the operating frequency used by FastHenry to calculate inductance. It is defined as $0.34/t_r$ with t_r being the signal transition time [Lillis et al. 1999].

the loop inductance model; for other general cases, K_{ij}=K_{ji} should be between 0 and 1, and is approximated by

$$K_{ij} = \frac{(f(i) + g(j))}{2} \tag{2}$$

which is the mean of $f(i)$ and $g(j)$, where $f(i) = \frac{(N_i - g_l)}{(N_j - g_l)}$ and $g(j) = \frac{(g_r - N_j)}{(g_r - N_i)}$ are two linear interpolation functions.

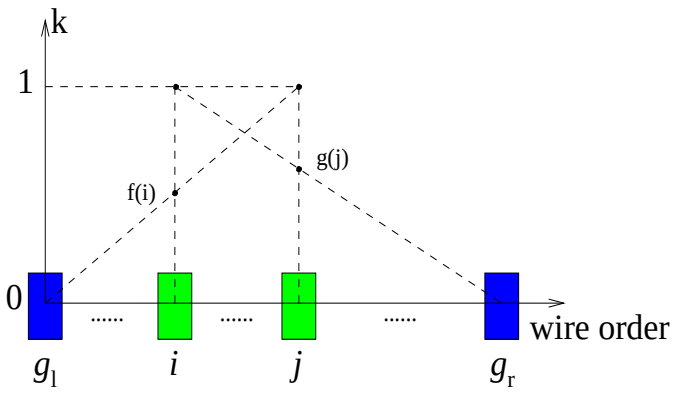

Fig. 4. Illustration of K_{ij} computation. N_i and N_j are two signal wires in the same block sandwiched by ground wires g_l and g_r. $f(i)$ and $g(j)$ are two linear interpolation functions as shown by the sloping dotted line. The mutual inductive coupling is given by the mean of $f(i)$ and $g(j)$.

To verify the accuracy of the formula-based K model, in Figure 5 we generate more than 1,000 data points to compare the coupling coefficients given by the formula-based K model and FastHenry under different wire widths, thicknesses, lengths, and spacings, as well as different frequencies. These parameters are summarized in Table 3. We use coplanar interconnect structures $g(6sg)^2 6sg$ and $g(3sg)^5 3sg$. In the figure, the x-axis is the coupling coefficient given by FastHenry and the y-axis is the coupling coefficient given by the formula-based K model. The three dotted lines indicate when the formula-based K model overestimates by 20%, is exactly equal, or under-estimates by 10%, the true value of K_{ij} computed with FastHenry. We observe that all data points essentially fall within the +20% to -10% range. Therefore, the formula-based K model is reasonably accurate and tends to be conservative.[4] The computational simplicity of the formula-based K model, along with its technology independence (shown previously), makes it highly desirable, and therefore we use it extensively in the problem formulations and algorithms to be presented.

[4]In practice, a model for K_{ij} is accurate enough whenever it leads to a high fidelity for the K_{eff} model to be defined in section 2.1.5. Therefore, the high fidelity shown in section 5 further validates the effectiveness of our formula-based K model.

wire width (μm)	0.8, 1.0, 1.2, 1.4, 1.6
wire thickness (μm)	0.8, 1.0, 1.2
wire length (μm)	1000, 1200, 1500, 1800, 2000
wire spacing (μm)	0.8, 1.0, 1.2, 1.4, 1.6
frequency (GHz)	30, 40, 50

Table 3. The parameters used in FastHenry to compute K_{ij} for comparison against the formula-based K model.

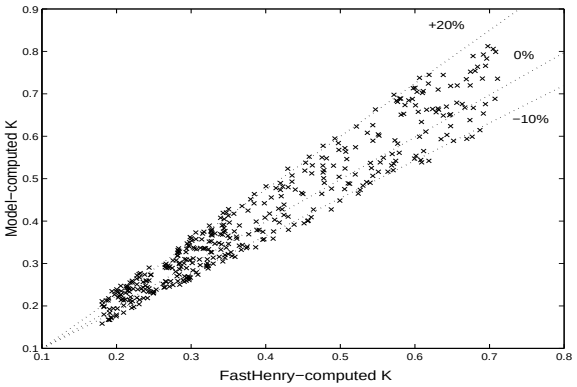

Fig. 5. Comparison of K_{ij} between the formula-based K model and FastHenry. The dotted lines indicate errors in the formula-based K model of +20%, 0%, and -10%.

2.1.5 *K_{eff} Model as a Figure of Merit for Inductive Coupling Considering Net Sensitivity.* We define the effective inductive coupling, or the effective K model (in short, K_{eff} model) as

$$K_i = \sum_{j \neq i} S_{ij} \cdot K_{ij} \tag{3}$$

where s_j is another s-wire. We will use the K_{eff} model as the figure of merit for the inductive noise that is induced on s_i. In section 5 we will show that the K_{eff} model has a high fidelity over the SPICE-computed noise voltage for SINO solutions under distributed RLC circuit models.

2.2 Optimal SINO Problems

We say that an s-wire s_i is capacitive noise free if it is not directly adjacent to any other s-wire s_j that is sensitive to it. Similarly, we say that s_i is inductive noise free if it does not share a block with any other sensitive wire s_j. We say a placement P (or equivalently, a SINO solution, or a SINO string) is noise free if, and only if, all nets s_i within P are free of both capacitive noise and inductive noise. With respect to these concepts, we define the following SINO problem to eliminate Cx and Lx noise and call it the noise-free SINO problem (SINO/NF).

FORMULATION 1. *(Optimal SINO/NF problem) For a given placement P, find a new placement P' by simultaneous shield insertion and net re-ordering such that P' is noise free and the total area of P' is minimal.*

In general, the SINO/NF problem is over-constrained and may lead to over-designed solutions as shown in section 4. To address more realistic design constraints, we define the following SINO problem to meet a given noise bound and call it the noise-bounded SINO problem (SINO/NB).

FORMULATION 2. *(Optimal SINO/NB problem) For a given placement P, find a placement P' with the minimum area by simultaneous shield insertion and net re-ordering such that any s_i in P' is free of capacitive noise and its inductive coupling to all sensitive wires s_j is less than a given noise bound.*

Therefore, we solve the following optimal SINO/NB problem under the K_{eff} model (defined as SINO/NB-K_{eff} problem): For a given placement P, find a new placement P' by simultaneous shield insertion and net re-ordering such that P' is free of capacitive coupling and the inductive coupling K_i satisfies $K_i \leq \overline{K_i}$ for any s-wire s_i where $\overline{K_i}$ is given a priori and is a measure of inductive noise that can be tolerated in s_i to maintain correct operation. Throughout the rest of this paper, we use a uniform value of $\overline{K_i}$ denoted as K_{th} (a noise threshold for K_i). However, our formulation, algorithms, and implementation are all able to handle non-uniform K_{th}'s. We only consider insertion of minimum width shields because it has already been shown that, in general, using additional shields is more effective than increasing the shield wire width [He et al. 1999].

We attempt to solve both the SINO/NF problem and SINO/NB-K_{eff} problem in section 3. For simplicity of presentation, we use SINO/NB as shorthand for SINO/NB-K_{eff} throughout the remainder of the paper. Note that our formulation and algorithms to be presented are applicable to inductive noise models more accurate than the K_{eff} model.

3. PROPERTIES AND ALGORITHMS

3.1 Properties of SINO/NF and SINO/NB Problems

THEOREM 1. *The Optimal SINO/NF problem is NP-hard.*

Proof: The optimal graph coloring problem determines a minimum number of colors and a color assignment for nodes in a graph such that no two adjacent nodes have the same color [Garey and Johnson 1979]. Given a graph coloring problem, we may always formulate a SINO/NF problem to solve it.[5] In the SINO/NF problem, an s-wire is a node in the graph, and two s-wires are sensitive to one another if there is an edge between two corresponding nodes in the graph. By definition, a solution to the SINO/NF problem separates all sensitive s-wires into different blocks. If we assign a unique color to each and every block, then the optimal SINO/NF solution with the minimal number of blocks solves the optimal graph coloring problem. Because the optimal graph coloring problem is NP-complete [Garey and Johnson 1979], the SINO/NF problem is NP-hard.

Because we can view the SINO/NF problem as a special case of the SINO/NB problem with the minimum noise bound (for SINO/NF, the noise bound is zero), we have the following theorem concerning the complexity for the SINO/NB problem:

[5]We are free to consider only inductive coupling in this case (and capacitive coupling is automatically included) because any capacitively coupled nets will always be inductively coupled (for SINO/NF).

THEOREM 2. *The optimal SINO/NB problem is NP-hard.*

Given that the SINO/NF and SINO/NB problems are NP-hard, we will focus on developing heuristic/approximate algorithms to solve the problems with satisfactory results. Before we present these algorithms, we first introduce the concept of a sensitivity graph. We can build a sensitivity graph such that a node corresponds to an s-wire and an edge exists between two nodes if and only if the correspondent s-wires are sensitive to one another. We then propose the following lower bound for the optimal SINO/NF solution:

THEOREM 3. *The maximum clique size in the sensitivity graph is a lower bound of the number of blocks required in all optimal SINO/NF solutions.*

The theorem follows directly from the formulation of the sensitivity graph and the definition of the maximum clique in a graph [Garey and Johnson 1979]. To illustrate this conclusion and further show that the maximum clique size is not an upper bound for the total number of blocks in optimal SINO/NF solutions, we present two examples in Figure 6. For the sensitivity graph in Figure 6(a), it is easy to verify that the graph has a maximum clique size of two. An optimal SINO/NF solution is $ABgCDE$, and there are two blocks in the solution. The sensitivity graph in Figure 6(b) is nearly the same as in Figure 6(a), except that an edge is added between C and D, i.e. nets C and D are sensitive to each other. The reader may easily verify that indeed this graph still only has a maximum clique size of two, but it is not possible to find a SINO/NF solution for this graph with two blocks. One optimal solution, consisting of three blocks, is $AEgCDgB$. Therefore, the maximum clique size is not a strictly lower bound of the number of blocks required in an optimal SINO/NF solution. Equivalently, it is not a strictly upper bound on the number of shields. Comparisons between the lower bound and the number of shield wires used will be presented in section 4 to illustrate the quality of our approximate algorithms.

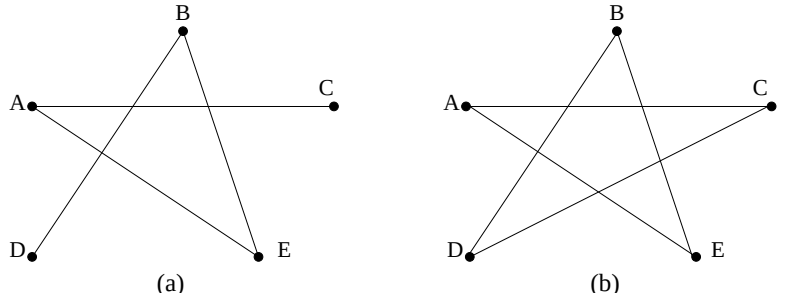

Fig. 6. Sensitivity graphs: (a) Both the maximum clique size and the number of blocks are two; (b) The maximum clique size is two, but the minimum number of required blocks is three.

3.2 Algorithms for SINO Problems

We develop graph coloring based algorithm (SINO/GC algorithm) to solve the SINO/NF problem, and greedy shield insertion (SI) algorithm, net ordering for

minimizing Cx noise followed by SI algorithm (NO+SI algorithm), and simulated annealing based algorithm (SINO/SA algorithm) for the SINO/NB problem. We will compare different problem formulations and algorithms in section 4. We consider Lx and Cx noise explicitly in all algorithms. In order to more succinctly describe the algorithms and not clutter them with trivial details, we also define the following operations and quantities:

—$Insert\_Shield(a)$: Given a placement P of size m, move all wires (s-wires and g-wires) at locations $a, a+1, ..., m-1$ to locations $a+1, a+2, ..., m$ in the placement, creating a new placement P'. Then, insert a g-wire at location a in P'. Finally set $P = P'$.

—$Compute\_Coupling(s_i)$: Given a placement P of size m, for each $s_k \neq s_i$ in the current block in P, if s_i is sensitive to s_k compute the K_{eff} between s_i and s_k. Sum the K_{eff} over all s_k's.

—$Compute\_Block\_Coupling(s_i)$: For each s_i in the current block, find the maximal K_i computed by $Compute\_Coupling(s_i)$ and return it (the maximal K_i for any s-wire within the block).

—$Max\_Clique(S)$: Compute the maximum clique in sensitivity graph S.

—$Compute\_Placement\_Cost()$: Compute the cost for a placement. Details of this are given in section 3.2.3.1.

With these definitions in place, we can succinctly describe our SINO/NF and SINO/NB algorithms. The SI and SINO/GC algorithms can be described easily and intuitively. SINO/SA is slightly more complicated, hence we assume that the reader is familiar with SA (for a discussion of SA in other VLSI design contexts, see [Sechen 1997], [Sherwani 1999]).

3.2.1 *Graph Coloring Based SINO/NF Algorithm.* In Figure 7, we present the graph coloring based SINO (SINO/GC) algorithm. This algorithm attempts to approximate a solution to the weighted graph coloring problem by using the maximum clique as a starting point. It works in the following way: First, determine the maximum clique in the sensitivity graph S. We saw in section 2 that this is a lower bound for the number of shields required in the SINO/NF problem. Let m be the maximum clique size. We first create a placement with m blocks. We then take each net s_i and try to place it into a block with no other sensitive wires. If we cannot do this, we simply create a new block and place s_i into it.

Note that the SINO/NF problem can be mapped into the graph coloring problem as outlined in the proof of Theorem 3, and then be solved using the graph coloring algorithms given in [Coudert 1997], [Kirovski and Potkonjak 1998].

3.2.2 *Greedy Shield Insertion SINO/NB Algorithm.* In Figure 8, we present the greedy shield insertion (SI) algorithm. The essence of the algorithm is the following: Run through the given placement P. If we encounter two adjacent sensitive nets, insert a shield between them. Also, at each location in the placement, calculate the maximum value of K_i that would exist in the current block if we allowed net s_i to become a member. If K_i is greater than K_{th}, then create a new block.

Because one shield is needed for every pair of adjacent sensitive wires, the solution given by the SI algorithm depends on the initial placement. Obviously, the number

```
Graph Coloring (GC) Algorithm:
Given an initial sensitivity graph S of signal nets:
MC = Max_Clique(S)
P = new placement with MC blocks
for each s-wire s_i
        for each block b in P (sorted from largest to smallest)
                if Compute_Block_Coupling(s_i) == 0
                        Insert s_i into b
                        next s_i
        endfor
        Insert_Shield(end_of_placement)
        Insert s_i into the new block
endfor
```

Fig. 7. Graph coloring based SINO/NF algorithm (SINO/GC)

```
SI Algorithm: Given a placement P
for each s-wire s_i in P at location a:
        if s_j at location P(a − 1) is sensitive to s_i
                Insert_Shield(a)
        BC = Compute_Block_Coupling(s_i)
        if (BC > K_th)
                Insert_Shield(a)
                P(a + 1) = s_i
endfor
```

Fig. 8. Greedy shield insertion SINO/NB algorithm (SI)

of shields can be reduced by first running existing net ordering algorithms to reorder nets so that no sensitive nets are adjacent to each other, then invoking the SI algorithm. This leads to the NO+SI algorithm.

```
Simulated Annealing Algorithm: Given a placement P:
Temp = Initial_Temperature;
Repeat
        Repeat
                Random_Move(P, P');
                Candidate_Cost = Compute_Cost(P');
                ds = Candidate_Cost − Compute_Cost(P);
                if (ds < 0)
                        P = P';
                else
                        r = RANDOM(0, 1);
                        if (r < exp(−ds/Temp))
                                P = P';
        Until equilibrium at Temp is reached;
        Temp = Temp * Temperature_Adjustment;
        /*(0 < Temperature_Adjustment < 1)*/
Until Temp == Freezing_Point;
```

Fig. 9. Simulated annealing based SINO/NB algorithm (SINO/SA)

3.2.3 *Simulated Annealing Based SINO/NB Algorithm.* In Figure 9, we present the simulated annealing based SINO (SINO/SA) algorithm. We give the details of our SINO/SA algorithm in the following subsections:

3.2.3.1 *Cost Function.* $Compute\_Cost(P)$ computes the cost for a placement P. The cost is the weighted sum of the following components: (i)*Cap_violation*: total number of nets that are adjacent to their sensitive nets in P; (ii)*Area*: total number of g-wires present in P; (iii)*Ind_Noise*: total number of $K_i > K_{th}$ violations in P; and (iv)*Inductance Violation Figure*. It is computed for a placement as shown in Figure 10. The purpose of the Inductance Violation Figure is to penalize a placement for the magnitude of K_{th} violations. Its usage (as opposed to simply forbidding placements P' that have K_{th} violations) allows the algorithm to potentially trade inductive noise violations for smaller overall placement size depending on the result desired, and can be useful in different SINO formulations. The weighting factor for each cost component can be tuned for different design objectives. In this paper our stated goal is to minimize placement size without violating K_{th} noise constraints, hence weighting factors were chosen to help us achieve these goals with maximal efficiency.

```
Total_Violation_Figure = 0;
for each s_i in placement P
        if Compute_Coupling(s_i) > K_th
                Total_Violation_Figure+ = (1 + K_eff − K_th)^3 − 1
end
```

Fig. 10. Computation of the Inductance_Violation_Figure

3.2.3.2 *Random Moves.* $Random\_Move(P, P')$ performs one of the following changes to placement P to generate a new placement P': (i) Combine two random blocks in P, (ii) Swap two random s-wires in P, (iii) Move a single random s-wire to a new and random location in P, (iv) Insert a g-wire at a random location in P. It is worthwhile to note that combining two random blocks in a placement P is also equivalent to removing a g-wire if the two blocks are adjacent. Moves which create two adjacent g-wires in a placement are categorically rejected and a new move is tried.

3.2.3.3 *Temperature Adjustment and Stopping Criterion.* The method of temperature adjustment is shown in Figure 9. We use a simple multiplicative constant of the current temperature. At each temperature step, the variance of the current placement cost from its previous value is taken and averaged over several random moves to determine the stability of the system at each temperature. When the variance is less than a set threshold, we move to the next temperature step. The starting temperature, freezing point, temperature adjustment, and variance threshold factors were all determined experimentally.

4. EXPERIMENTAL RESULTS

We have implemented all algorithms in the C programming language, and have tested our implementations using a large number of examples. In this section, we

Maximum and average values of K_i						
		noise-free	noise-bounded			
	K_{th}	SINO/GC	SI	NO+SI	SINO/SA	US+NO
Number of nets: 32	**Net sensitivity rate: 40%**					
	0.5	0/0	0.39/0.21	0.48/0.40	0.43/0.32	0.32/0.18
	1.0		0.64/0.55	0.84/0.66	0.82/0.73	0.61/0.52
	1.5		1.27/1.18	1.36/1.29	1.44/1.38	1.28/1.16
	2.0		1.53/1.34	1.73/1.49	1.86/1.62	1.66/1.43
	Net sensitivity rate: 50%					
	0.5	0/0	0.50/0.38	0.44/0.31	0.46/0.37	0.45/0.33
	1.0		0.75/0.62	0.86/0.64	0.93/0.75	0.73/0.60
	1.5		1.31/1.19	1.37/1.21	1.45/1.32	1.38/1.17
	2.0		1.68/1.56	1.70/1.56	1.87/1.73	1.58/1.52
	Net sensitivity rate: 60%					
	0.5	0/0	0.47/0.44	0.45/0.41	0.46/0.43	0.44/0.35
	1.0		0.83/0.69	0.88/0.74	0.92/0.84	0.80/0.65
	1.5		1.39/1.22	1.47/1.42	1.43/1.31	1.27/1.13
	2.0		1.74/1.59	1.88/1.76	1.95/1.82	1.75/1.62
Number of nets: 64	**Net sensitivity rate: 40%**					
	0.5	0/0	0.42/0.29	0.46/0.43	0.44/0.39	0.35/0.26
	1.0		0.70/0.61	0.92/0.74	0.85/0.77	0.69/0.58
	1.5		1.35/1.23	1.42/1.34	1.41/1.35	1.30/1.22
	2.0		1.62/1.47	1.77/1.53	1.95/1.78	1.74/1.60
	Net sensitivity rate: 50%					
	0.5	0/0	0.46/0.39	0.44/0.28	0.43/0.40	0.47/0.39
	1.0		0.74/0.65	0.91/0.76	0.96/0.84	0.80/0.67
	1.5		1.35/1.18	1.46/1.30	1.45/1.28	1.41/1.24
	2.0		1.73/1.59	1.79/1.62	1.90/1.77	1.64/1.60
	Net sensitivity rate: 60%					
	0.5	0/0	0.42/0.38	0.42/0.36	0.47/0.42	0.45/0.36
	1.0		0.80/0.69	0.91/0.78	0.94/0.85	0.87/0.70
	1.5		1.34/1.15	1.44/1.36	1.43/1.35	1.29/1.23
	2.0		1.78/1.67	1.86/1.73	1.94/1.85	1.82/1.66

Table 4. Experiment Summary I: In each cell of columns 3-7, the first value is the maximum coupling and the second value is the average coupling observed for different SINO algorithms.

first compare results obtained by different approximate algorithms to the SINO/NB formulation, and then compare results given by the two formulations (noise-free versus noise-bounded). We also report the running times for the SINO/SA algorithm.

We use coplanar interconnect structures containing 32 and 64 s-wires as examples to determine the performance of the algorithms for different combinations of K_{th} and sensitivity rate. We consider the following values for K_{th}: 0.5, 1.0, 1.5, and 2.0. When $K_{th} = 0.5$, the total coupling value for any net is less than 0.5 in the target SINO solution. We iterate the following sensitivity rates: 40%, 50%, and 60%. When the sensitivity rate is 40%, each net is sensitive to 40% of all nets, and these sensitive nets are picked randomly for the given s-wire.

For each combination of K_{th} and sensitivity rate, we report the resulting numbers of shields for different formulations and algorithms. To make the comparisons "fair" among the different algorithms, we run each algorithm on the same set of 20 different random initial placements and sensitivity matrices. The average values of these 20 runs are shown in Tables 4 and 5. Note that there is no entry in Table 4 for Cx

Average number of shields						
		noise-free	noise-bounded			
	K_{th}	SINO/GC	SI	NO+SI	SINO/SA	US+NO
Number of nets: 32	**Net sensitivity rate: 40%**					
	0.5	8.5(6.0)	16.7	6.3	5.3	17
	1.0		15.9	5.3	4.4	9
	1.5		15.4	4.4	3.6	9
	2.0		13.8	4.0	3.2	9
	Net sensitivity rate: 50%					
	0.5	11.0(8.0)	18.9	8.4	5.7	17
	1.0		18.4	5.7	5.4	17
	1.5		18.0	4.8	4.2	9
	2.0		17.5	4.4	3.8	9
	Net sensitivity rate: 60%					
	0.5	12.0(9.0)	22.8	6.6	6.3	17
	1.0		22.1	6.0	5.8	17
	1.5		22.0	5.2	5.0	17
	2.0		21.5	4.5	4.1	17
Number of nets: 64	**Net sensitivity rate: 40%**					
	0.5	15.6(13.0)	30.1	11.3	9.5	33
	1.0		28.5	9.2	7.5	17
	1.5		26.2	7.5	6.1	17
	2.0		23.5	6.8	5.4	17
	Net sensitivity rate: 50%					
	0.5	19.3(16.0)	32.1	14.3	10.2	33
	1.0		31.3	9.8	9.1	33
	1.5		30.6	8.6	7.6	17
	2.0		31.5	7.9	6.9	17
	Net sensitivity rate: 60%					
	0.5	21.6(17.0)	39.7	12.8	12.0	65
	1.0		39.2	11.4	10.7	33
	1.5		38.0	9.4	9.0	33
	2.0		36.8	8.1	7.4	33

Table 5. Experiment Summary II: Number of shields inserted by SINO different algorithms. The numbers in parentheses in each cell of column 3 are lower bounds on the number of shields required in SINO/NF solutions.

noise because there was not any in all cases. Finally, it is worthwhile to point out that we did not tune our SINO/SA algorithm for different examples.

4.1 Comparison Between Different Noise-Bounded Algorithms

We first compare approximate SINO/NB solutions given by the following algorithms: greedy shield insertion (SI), net ordering followed by SI (NO+SI), and simulated annealing based SINO (SINO/SA). All solutions satisfy the required noise bound as shown in Table 4. One may also easily see from Table 5: SI is always significantly worse than all of the other solutions in terms of the number of shields inserted. In the worst case, SI yields a result about 497% worse than SA (see 64 nets, 60% sensitivity rate, and 2.0 K_{th} in the table). Furthermore, performing net ordering before SI (i.e., NO+SI) significantly outperforms SI only because we need not insert shields for Cx violations which may be present without performing net ordering.

As we expect, SINO/SA always performs the best for any given settings. In terms

of the number of shields, compared to NO+SI, it is minimally 4% better at 32 nets, 60% sensitivity rate, and 1.0 K_{th}, and is maximally 47% better at 32 nets, 50% sensitivity rate, and 0.5 K_{th}. Therefore, it is important to consider simultaneous net ordering and shield insertion, rather than separated net ordering and shield insertion.

Note that all the above noise-bounded SINO algorithms use random shields in the sense that all shields can be placed at arbitrary tracks. This may be undesirable when a uniform P/G structure is preferred due to other design constraints. Also in Table 5, we compare these solutions with the minimal area uniform shield insertion and net ordering (denoted as US+NO) problem formulation, which attempts to determine a minimum placement size subject to a noise bound (under the K_{eff} model) for each s-wire with the additional constraint of a uniform shielding structure. The US+NO algorithm and additional description of the problem is presented in [Ma and He 01 1]. It is obvious from the table that the min-area US+NO solutions are only slightly better than the SI solutions, but much worse than the NO+SI and SINO/SA solutions, indicating that allowing random shields is critical for reducing overall shield resource usage.

4.2 Comparison Between Noise-Free and Noise-Bounded Formulations

We base our comparison on the results produced by the best SINO/NB algorithm (SINO/SA) and the GC algorithm for the SINO/NF formulation. For the smallest K_{th} value of 0.5 (in order to make SINO/NB most closely approximate SINO/NF), compared to the SINO/NF formulation, the SINO/NB formulation uses about 38% to 61% fewer shields for different numbers of nets and sensitivity rates. Because our GC algorithm provides an approximate to the SINO/NF formulation, we compute the average lower bound of the number of shields via average maximum clique size (based on Theorem 3), and present these lower bounds between parentheses in the column for GC algorithm, SINO/NF formulation in Table 5. Compared to these lower bounds, the SINO/SA solutions still use up to 36% fewer shield wires for K_{th}=0.5.

Number of nets	**Net sensitivity rate: 40%**				**Net sensitivity rate: 50%**				**Net sensitivity rate: 60%**			
	K_{th}				K_{th}				K_{th}			
	0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0
16	1.5	1.2	1.0	0.8	1.8	1.5	1.2	1.0	2.3	1.9	1.6	1.4
32	2.9	2.4	2.1	1.5	3.4	3.0	2.2	1.8	4.5	3.6	3.1	2.6
48	4.3	3.5	3.0	2.3	5.5	4.4	3.6	2.9	7.0	5.7	5.0	4.5
64	5.5	4.8	3.9	2.7	6.7	5.7	4.5	3.7	9.2	7.1	6.4	5.3

Table 6. Approximate running times (in seconds) for SINO/SA algorithm.

Finally, we report the running times of SINO/SA algorithm in Table 6 for different combinations of number of nets, sensitivity rate, and K_{th}, where the times are for a single run for a single interconnect structure. The machine used to collect the running times has a 450MHz Intel Pentium II processor. The running time is roughly linear with respect to the number of nets. Also the running time increases

with respect to increased sensitivity rate and decreased K_{th}. For layout optimization at the full chip level, we expect to first divide the full chip into a number of routing regions and then carry out SINO within each region. Therefore, the chip level application should also have a roughly linear complexity in the number of regions (i.e., chip area). This approach has been applied in [Ma and He 2002] for an extended global routing problem with acceptable running times. Nevertheless, a SINO algorithm that is more efficient than the simulated annealing based approach is planned as part of our future work.

5. FIDELITY OF K_{EFF} MODEL

The K_{eff} model is computationally simple and convenient to use at a high design level or in an early design stage. The computational simplicity of the model also leads to extremely fast running times for high quality solutions as illustrated in Table 6. However, it is not clear how well the K_{eff} model works as a figure of merit for the actual noise voltage that will be seen in circuits. Specifically, one limitation of the K_{eff} model is that it assumes the current returns from the nearest shield, which may not be true in general. The current often returns from quiet wires within the current block if there are plenty of quiet wires in the block. On the other hand, the current often returns from shields or quiet wires outside the current block when multiple wires in the current block switch simultaneously.

We study the fidelity of the K_{eff} model for SINO solutions in this section. The SINO solutions are all generated by the SINO/SA algorithm with respect to the SINO/NB problem formulation. We use SPICE to simulate the SINO solutions under two technologies. We use the PEEC inductance model that is able to consider the skin effect [Ruehli 1974; He et al. 1999]. Different from the loop inductance model used in our problem formulation, the PEEC model does not assume a current return path during inductance extraction. The current return path is calculated via circuit simulations under the distributed RLC circuit model. We generate an RLC segment for each $100\mu m$ wire segment with a coupling capacitance to its adjacent segment that does not belong to the same wire. There is a coupling inductance between any pair of segments even if they belong to the same wire. We use SPICE to simulate such a RLC circuit model, and use parameters based on ITRS [Semiconductor Industry Association 2000] predicted $0.10\mu m$ and $0.07\mu m$ technologies (see Table 7)."

In Table 8 we first present the SPICE-computed noise for SINO solutions to 32 nets given in Table 4 under $0.10\mu m$ technology. As the Table indicates, despite the fact that the K_{eff} model does not allow specification of an explicit noise voltage, it has a high fidelity in the following sense: a SINO solution with a higher coupling value (K_i) under the K_{eff} model also has a higher SPICE-computed average noise voltage under distributed RLC circuit models.

To further verify the fidelity for the K_{eff} model as a figure of merit for inductive coupling, and therefore also explicit noise voltage, we rank each net in every SINO solution–first according to its coupling value K_i under the K_{eff} model and then according to its SPICE-computed noise voltage under accurate RLC circuit models. If the ranking difference is small enough, the K_{eff} model has a high fidelity. Similar ranking techniques have been used to study the fidelity of the Elmore delay model in [Boese et al. 1994; Cong and He 1996].

technology	$0.10\mu m$	$0.07\mu m$
Vdd	1.05V	0.75V
clock frequency	3GHz	5GHz
input rising time	$33ps$	$20ps$
driver resistance	150Ω	100Ω
load capacitance	60fF	40fF
wire width	$1.0\mu m$	$1.0\mu m$
wire thickness	$1.1\mu m$	$1.1\mu m$
wire spacing	$0.8\mu m$	$0.8\mu m$
wire length	$2000\mu m$	$1400\mu m$

Table 7. Experiment settings for the fidelity study. We derive Vdd, clock frequency and rising time based on ITRS, and assume geometries and driver/receiver sizes for high-performance global interconnects.

K_{th}	$K_{i,max}$	$K_{i,avg}$	$Noise_{max}$	$Noise_{avg}$
Net sensitivity rate: 40%				
0.5	0.43	0.32	0.1237	0.1032
1.0	0.82	0.73	0.1258	0.1097
1.5	1.44	1.38	0.1301	0.1164
2.0	1.86	1.62	0.1366	0.1207
Net sensitivity rate: 50%				
0.5	0.46	0.37	0.1263	0.1054
1.0	0.93	0.75	0.1288	0.1102
1.5	1.45	1.32	0.1339	0.1195
2.0	1.87	1.73	0.1397	0.1261
Net sensitivity rate: 60%				
0.5	0.46	0.43	0.1303	0.1076
1.0	0.92	0.84	0.1364	0.1128
1.5	1.43	1.31	0.1401	0.1215
2.0	1.95	1.82	0.1445	0.1294

Table 8. Fidelity of the K_{eff} model to SPICE-computed noise voltage using detailed RLC circuit models under $0.10\mu m$ technology. Columns 2 and 3 indicate the maximum and average values of K_i and columns 4-7 indicate actual noise voltage (in V), respectively.

Moreover, we calculate the SPICE-computed average noise difference with respect to the average ranking difference in the following way: let the average ranking difference be d. For a SINO/NB solution whose SPICE-computed average noise ranking is i, we compute the relative difference between the $(i+d)$-th and i-th SPICE-computed average noise, as well as that between the $(i-d)$-th and i-th SPICE-computed average noise. Between the two values, the one with the larger absolute value divided by the i-th SPICE-computed average noise value is defined as the *noise difference* for the solution. In other words, it shows how much the noise difference is introduced if we use K_{eff} model to approximate the SPICE-computed average noise voltage under accurate RLC circuit models.

We report in Figure 11 the distribution of ranking differences for SINO solutions under $0.10\mu m$ and $0.07\mu m$ technologies. Over 3,840 SINO solutions, the average ranking difference is only 124.5 for $0.10\mu m$ technology and 153.8 for $0.07\mu m$ technology. The average noise difference with respect to such a ranking difference is about 6.20% for $0.10\mu m$ technology and 7.94% for $0.07\mu m$ technology. In addition,

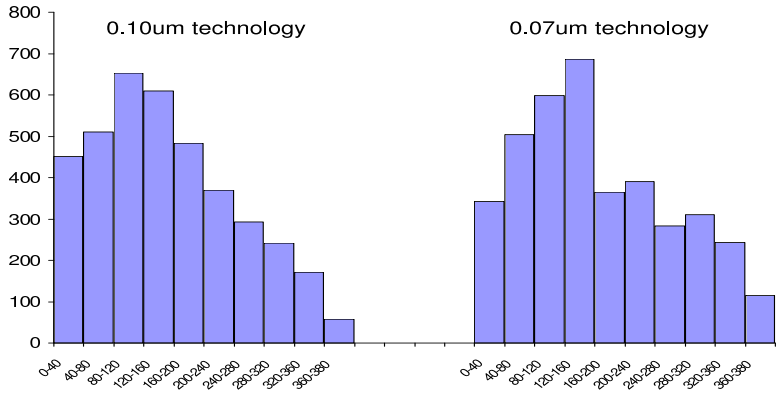

Fig. 11. The distribution of ranking differences for 3,840 SINO solutions under $0.10\mu m$ and $0.07\mu m$ technologies, respectively.

in Table 9 we present the average noise difference for SINO solutions with different coupling values. One can easily see that the smaller the coupling, the smaller the average noise difference. This observation holds for both technologies. We assume in our experiment that the largest coupling value is 2.0 under the K_{eff} model. The maximum $K_{th} = 2.0$ is a "good" bound in the sense that the resulting SINO solutions have SPICE-computed maximum noise close to, but still less than, 15% of the supply voltage Vdd (see Table 8).

	$0 < K_i < 0.5$	$0.5 < K_i < 1.0$	$1.0 < K_i < 1.5$	$1.5 < K_i < 2.0$
$0.10\mu m$ technology	5.119%	5.987%	6.552%	7.149%
$0.07\mu m$ technology	6.925%	7.348%	8.061%	9.259%

Table 9. The average noise difference for different coupling values

Based on the above empirical evidence, we conclude that the K_{eff} model has a high fidelity as compared to SPICE-computed noise using accurate RLC circuit models. This fidelity has been illustrated for min-area SINO solutions under a practical noise bound. The SINO algorithms are able to provide sufficient shields and quiet wires for a switching net, and allow the current of the switching net to return mainly within the block defined by the nearest shields. Satisfaction of this fundamental assumption leads to high fidelity of the K_{eff} model. A similar fidelity may *not* hold for other layout design schemes such as the twisted-bundle layout scheme where the current is assumed to return from the nearest shield [Zhong et al. 2000].

6. CONCLUSIONS AND DISCUSSIONS

We have shown that existing net ordering formulations to minimize noise are no longer sufficient when considering inductive noise, and shield insertion is needed to minimize inductive noise. We have formulated two simultaneous shield insertion and net ordering (SINO) problems: the optimal SINO/NF problem to find a

min-area SINO solution that is free of capacitive and inductive noise, and the optimal SINO/NB problem to find a min-area SINO solution that is free of capacitive noise and is under the given inductive noise bound. We have revealed that both optimal SINO problems are NP-hard, and have proposed effective approximate algorithms for the two problems. Experiments have shown that our best SINO/NB algorithm uses from 4% to 47% fewer shields when compared to separated net ordering and shield insertion (NO+SI). Furthermore, under practical noise bounds, the SINO/NB solutions use up to 61% fewer shields compared to SINO/NF solutions and use up to 36% fewer shields compared to the theoretical lower bound for optimal SINO/NF solutions. Additionally, the best simulated annealing based SINO/NB algorithm is very efficient and finish all examples in a few seconds. In this work, the K_{eff} model is used as a figure of merit for inductive coupling. We have also applied the partial inductance model to generate accurate RLC circuit models and used SPICE simulations (without pre-setting any current return path) to verify our problem formulation and SINO solutions. Extensive experiments have shown that the K_{eff} model has a high fidelity versus the accurate noise voltage computed using accurate RLC circuit models and SPICE simulations for the coplanar interconnect structure we studied here, i.e., the higher the coupling given by the K_{eff} model, the larger the accurate SPICE-computed noise voltage. To the best of our knowledge, this is the first work that presents an in-depth study on the automatic layout optimization of multiple nets to minimize both capacitive and inductive noise.

In [Ma and He 2001], a set of simple yet accurate formulae has been developed to estimate the numbers of shields needed by the minimum area SINO solutions under the K_{eff} model. Moreover, such pre-routing estimation has been used to solve a new minimum area SINO problem. The new SINO problem defines an optimal P/G structure such that the minimal number of shields are needed by the min-area SINO solution with respect to the above P/G structure which is also considered a shield resource. The min-area SPR solutions maintain the regularity of the P/G structure, but reduce the total shielding area by up to 44% compared to the min-area US+NO solutions.

An explicit noise computation model has been developed in [Lepak et al. 2001]. It computes the worst-case noise voltage considering both capacitive coupling and long-range inductive coupling based on the partial inductance model and therefore allows for specification of an explicit noise voltage (N_{th}) as the noise bound as compared to K_{th} used in this work. The simulated annealing based SINO algorithm has been extended to consider this new noise model. Experimental results show that the new SINO solutions can use up to 20% fewer shields compared to US+NO and up to 30% fewer shields compared to NO+SI for explicit noise voltage bounds.

A parallel bus interconnect structure within a routing region is assumed in this work. We are extending our SINO formulations to consider additional design freedom of buffer insertion to further minimize coupling noise, and incorporating the resulting algorithm into a chip level global router with RLC crosstalk constraints [Ma and He 2002].

REFERENCES

Boese, K. D., Kahng, A. B., McCoy, B. A., and Robins, G. 1994. Rectilinear Steiner

trees with minimum Elmore delay. In *Proc. Design Automation Conf* (1994), pp. 381–386.

Chang, C.-C. and Cong, J. 2000. Pseudo pin assignment with crosstalk noise control. In *Proc. Int. Symp. on Physical Design* (April 2000).

Cong, J. and He, L. 1996. Optimal wiresizing for interconnects with multiple sources. *ACM Trans. on Design Automation of Electronics Systems*.

Coudert, O. 1997. Exact coloring of real-life graphs is easy. In *Proc. Design Automation Conf* (1997).

Gao, T. and Liu, C. L. 1993. Minimum crosstalk channel routing. In *Proc. Int. Conf. on Computer Aided Design* (Nov. 1993), pp. 692–696.

Gao, T. and Liu, C. L. 1994. Minimum crosstalk switchbox routing. In *Proc. Int. Conf. on Computer Aided Design* (Nov. 1994), pp. 610–615.

Garey, M. R. and Johnson, D. S. 1979. *Computers and Intractability: A guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco.

He, L., Chang, N., Lin, S., and Nakagawa, O. S. 1999. An efficient inductance modeling for on-chip interconnects. In *Proc. IEEE Custom Integrated Circuits Conference* (May 1999), pp. 457–460.

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

Kay, R. and Rutenbar, R. A. 2000. Wire packing: A strong formulation of crosstalk-aware chip-level track/layer assignment with an efficient integer programming solution. In *Proc. Int. Symp. on Physical Design* (April 2000).

Kirovski, D. and Potkonjak, M. 1998. Efficient coloring of a large spectrum of graphs. In *Proc. Design Automation Conf* (1998).

Lepak, K. M., Luwandi, I., and He, L. 2001. Simultaneous shield insertion and net ordering for coupled RLC nets under explicit noise constraint. *Proc. Design Automation Conf*.

Lillis, J., Cheng, C., Lin, S., and Chang, N. to appear in 1999. *High-performance interconnect analysis and synthesis*. John Wiley.

Ma, J. D. and He, L. 2001. Pre-routing estimation of shielding for RLC signal integrity. In *Proc. IEEE Int. Conf. on Computer Design* (2001).

Ma, J. D. Z. and He, L. 2002. Towards global routing with RLC crosstalk constraints. *accepted by Design Automation Conference*.

Ma, J. D. Z. and He, L. ECE-01-1. Simultaneous signal and power routing based on interconnect estimation. In *U. of Wisconsin at Madison, Technical Report* (ECE-01-1). (Accessible from http://eda.ece.wisc.edu/publications.html#TechReport).

Ruehli, A. 1974. Equivalent circuit models for three-dimensional multiconductor systems. *IEEE Trans. on MIT*.

Sechen, C. 1997. An improved simulated annealing algorithm for row-based placement. In *Proc. Int. Conf. on Computer Aided Design* (1997), pp. 478–481.

Semiconductor Industry Association. 2000. *International Technology Roadmap for Semiconductors*. Semiconductor Industry Association.

Sherwani, N. 1999. *Algorithms for VLSI Physical Design Automation*. Kluwer Academic Publishers.

Xue, T. and Kuh, E. S. 1997. Post global routing crosstalk synthesis. *IEEE Trans. CAD-16*, 1418–1430.

Yim, J. S. and Kyung, C. M. 1999. Reducing cross-coupling among interconnect wires in deep-submicron datapath design. In *Proc. Design Automation Conf* (June 1999).

Zhong, G., Wang, H., Koh, C.-K., and Roy, K. 2000. A twisted bundle layout structure for minimizing inductive coupling noise. In *Proc. Int. Conf. on Computer Aided Design* (2000).

Summary of Revision

Major Changes

We have revised the manuscript carefully according to reviewers' comments and suggestions. The following is a summary of the major changes:

1. Further explanation on derivation of Keff model

In this revision, we use formula-based K model to refer to the computation of K_{ij} via formula (called "K_{eff} model" in the first submission), but use K_{eff} model to refer to applying the weighted sum of K_{ij} as the figure of merit for inductive noise. We feel that introducing two separated terms helps to better explain the above two orthogonal concepts. Therefore, the derivation of K_{eff} model in the original version is the derivation of formula-based K model explained below.

In section 2.1.3 of this revision, we discuss the characteristics of inductive coupling, define the inductive coupling coefficient as Formula (1) in page 4, and use FastHenry to show that the coefficient is not sensitive to wire width, thickness, length, and spacing, as well as frequency (see Table 2 in page 5).

This also stimulates us to propose the formula-based K model (see Formula (2) in page 6) in section 2.1.4, with further explanations of the intuition to derive the model as follows (see page 6):
"When the two s-wires are in the same block, as shown in Figure 4 where N_i and N_j are track ordering numbers for the two s-wires, and g_l and g_r are track ordering numbers for the two edge g-wires, we consider the following cases for the coupling coefficient computation: when $i = j$, the mutual inductance is reduced to self inductance and $K_{ii} = 1$ by definition; when N_i (or N_j) becomes g_l (or g_r), because we assume the loop inductance model with current returning from shields, inductive coupling K_{ij} is between the two segments of the same current loop and is 0 under the loop inductance model; for other general cases, K_{ij}=K_{ji} should be between 0 and 1, and is approximated by Formula (2), which is the mean of $f(i)$ and $g(j)$, where $f(i) = \frac{(N_i - g_l)}{(N_j - g_l)}$ and $g(j) = \frac{(g_r - N_j)}{(g_r - N_i)}$ are two linear interpolation functions."

In section 2.1.5, we define the effective inductive coupling, or the K_{eff} model as Formula (3) in page 7 (i.e., a weighted sum of K_{ij} values). We use this model as a figure of merit for the inductive noise that is induced on the net under study.

2. Further justification/validation of Keff model (on multiple technologies?)

In this revision, we verified the accuracy of the formula-based K model and the fidelity of the K_{eff} model separately.

In Figure 5, page 7 of this revision we verify the accuracy of the formula-based K model by using much more data than the earlier submission to compare the coupling coefficients given by the model and FastHenry under different wire widths, thicknesses, lengths, and spacings, as well as different frequencies. We also point out in footnote 4, page 6:
"In practice, a model for K_{ij} is accurate enough whenever it leads to a high fidelity for the K_{eff} model to be defined in section 2.1.5. Therefore, the high fidelity shown in section 5 further validates the effectiveness of our formula-based K model."

Moreover, in section 5, pages $16 - 18$ we have carried out SPICE simulation under

two different technologies and have shown that the K_{eff} model has good fidelity in both technologies (see Table 9 and Figure 11).

3. Report your experimental results on larger test cases

For experiments in sections 4.1 and 4.2, we use both original 32-bit buses and new 64-bit buses. We report the running times in Table 6, page 15 for different numbers of nets, sensitivity rates and K_{th}'s. We point out the following in pages 15 and 16:

"The running time is roughly linear with respect to the number of nets. Also the running time increases with respect to increased sensitivity rate and decreased K_{th}. For layout optimization at the full chip level, we expect to first divide the full chip into a number of routing regions and then carry out SINO within each region. Therefore, the chip level application should also have a roughly linear complexity in the number of regions (i.e., chip area). This approach has been applied in [Ma and He 2002] for an extended global routing problem with acceptable running times. Nevertheless, a SINO algorithm that is more efficient than simulated annealing based approach is planned as part of our future work."

Specific Changes

In response to specific comments of the reviewers, we have included the following discussions:

Reviewer #1.

1. *Is the information in the paper sound, factual, and accurate? YES NO If no please explain why.*

Seems OK – but the concern I have is that the Keff model is "validated" with only one semi-physical simulation, in only one geometric configuration, in only one set of technology rules. It works OK for the one case presented (which is also, it seems, in a not-very-deep-into-submicron technology) but one would like to see a few different scenarios simulated, across a few different technologies (if possble, which I understand is not straightforward). The problem with E&M is that as the problems get more aggressively coupled, the behavior gets more difficult, eg, skin effects, difficulty in knowing where the current return, etc. The fact that it works in one geometric/electrical case is not really sufficient to ensure that it works in general.

Thanks for the suggestion. As indicated by the summary of changes, we have considered more geometries and technologies in Table 2 in page 5 and Figure 5 in page 7 for the calculation of K_{ij} values. Further, we have used the ITRS [Semiconductor Industry Association 2000] predicted $0.10\mu m$ and $0.07\mu m$ technologies as shown in Table 7, page 17 for the fidelity study of the K_{eff} model. Specifically, we have pointed out the following in page 16:

"We use the PEEC inductance model that is able to consider the skin effect [Ruehli 1974; He et al. 1999]. Different from the loop inductance model used in our problem formulation, the PEEC model does not assume a current return path during inductance extraction. The current return path is calculated via circuit simulations under the distributed RLC circuit model. We generate an RLC segment for each $100\mu m$ wire segment with a coupling capacitance to its adjacent segment that does

not belong to the same wire. There is a coupling inductance between any pair of segments even if they belong to the same wire. We use SPICE to simulate such a RLC circuit model, and use parameters based on ITRS predicted $100nm$ and $70nm$ technologies (see Table 7)."

2. *Should anything be deleted from or condensed in the paper? If yes, please explain.*

Given the simplicity of the optimization algorithms, perhaps some detail here could be omitted.

Thanks for the suggestion. We omit the figure to illustrate the data structure for net sensitivity. However, we choose not to further omit algorithms, theorems, and proofs considering the self-containing requirement of a journal paper.

3. *Is the treatment of the subject complete (no important ideas, analyses, or information omitted)? If no, please explain.*

Would be very useful to see more validation on the Keff model.

Thanks. Please see the summary of changes.

4. *Are these algorithms fast enough to really do large amounts of interconnect? 0.1s for the NO+SI for 32 wires. If we assume 1cm x 1cm, in a 0.12um process, we could pack about 20,000 tracks/cm across one edge of the chip. If we have 1M wire segments, do we need 1M *0.1s = 100,000s = 28hours to do this optimization? Since placers and routers at this size take days to run anyway, this may be ok.*

Some speculation on scaling (directly, or via some partitioning) to problems of practical scale, would be useful.

Thanks. Please see the summary of changes.

Reviewer #2.

1. *Is the information in the paper sound, factual, and accurate? YES NO If no please explain why.*

Most of the parts in the paper are well written and sound. However, the formulation of the K_{eff} needs more explanations. The intuition of forming such approximate formula seems clear, however, why choosing such formula needs more justifications. Although, the simulations show the errors are within 15% windows for two configurations, it is not clear if this error bound can be held for other configurations.

Thanks. Please see the summary of changes.

2. *Should anything be deleted from or condensed in the paper? YES NO If yes, please explain.*

YES. The algorithms and NP-completeness proofs are simple. The authors may want to condense them.

See the answer to review #1, comment 2.

3. *Is the treatment of the subject complete (no important ideas, analyses, or information omitted)? YES NO If no, please explain.*

NO. The experiments contain only a 32-bit bus. It would be better if other test cases can be use. Because the K_{eff} model is obtained empirically, providing experiments on different test cases will be more convincing.

Thanks. Please see the summary of changes.

Reviewer #3.

1. *I read the paper carefully. The paper is well written and interesting. However, a minor comment is that the Keff theory should be elaborated. The reviewer's guess is that the linear interpolation functions g(i) and f(i) in figure 3 and the definition of keff are based on the curves shown in figure two. The authors just define their model and jump to experimental results without any justification or theoretical background for the validity of this model.*

Thanks. Please see the summary of changes.