Noise-Constrained Performance Optimization by Simultaneous Gate and Wire Sizing Based on Lagrangian Relaxation *†

Hui-Ru Jiang[1], Jing-Yang Jou[1], and Yao-Wen Chang[2]

[1]Department of Electronics Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan

[2]Department of Computer and Information Science, National Chiao Tung University, Hsinchu 30010, Taiwan

Abstract

Noise, as well as area, delay, and power, is one of the most important concerns in the design of deep sub-micron ICs. Currently existing algorithms can not handle simultaneous switching conditions of signals for noise minimization. In this paper, we model not only physical coupling capacitance, but also simultaneous switching behavior for noise optimization. Based on Lagrangian relaxation, we present an algorithm that can optimally solve the simultaneous noise, area, delay, and power optimization problem by sizing circuit components. Our algorithm, with linear memory requirement overall *and linear runtime* per iteration, *is very effective and efficient. For example, for a circuit of 6144 wires and 3512 gates, our algorithm solves the simultaneous optimization problem using only 2.1 MB memory and 47 minute runtime to achieve the precision of within 1% error on a SUN UltraSPARC-I workstation.*

1 Introduction

With decreasing feature sizes, higher clock rates, and increasing interconnect densities, noise is getting a greater concern of comparable importance to power, area, and timing in integrated circuits [13]. While power, area, and timing have been extensively discussed in the recent literature, e.g., [2, 3, 4, 11], relatively less work has been done on noise.

Noise profoundly affects the performance of a circuit, especially in the deep sub-micron regime. Noise is an unwanted variation making the behavior of a manufactured circuit deviate from the expected response [12]. The deleterious influences of noise include malfunctioning and timing change, caused by switching behavior. Crosstalk is a type of noises introduced by an unwanted coupling between a node and its neighboring wire or between two neighboring wires. For example, two adjacent wires form a coupling capacitor and a mutual inductor. The inductive effects [10] must be considered as circuit frequencies increase above 500 MHz. The effects are beyond the scope of this paper.

In this paper, we focus on the capacitive effects of crosstalk. We refer to the capacitance created by the physical geometry as the *physical coupling capacitance*. The physical coupling capacitance is directly proportional to the overlap length of adjacent wires and is inversely proportional to the distance between them. There exist other models to view the physical coupling capacitance from different perspectives, e.g., [6, 15]. Coupling capacitance is dominated not only by physical geometry, but also by switching conditions [9]. However, currently existing literature handles only physical coupling capacitance. The influence of switching conditions can be explained by the Miller and the anti-Miller effects [1]. Assume that the physical coupling capacitance between two neighboring wires is C_c. The Miller effect occurs when the adjacent wires switch in opposite directions; the equivalent coupling is $2C_c$. On the contrary, the anti-Miller effect happens when the adjacent wires switching in the same direction; the equivalent coupling is 0. In the appearance of the anti-Miller effect, the transition of wires can be shortened so that the logic values become stable earlier. If two wires have very large physical coupling capacitance but possess the same switching behavior, the inter-wire crosstalk can be very small. Hence, it is often too pessimistic if we consider only the Miller effect. However, the anti-Miller effect is hard to be considered because of its uncertainty. Though some previous work has mentioned this problem, yet there is no literature solving this problem so far.

In this paper, we model not only physical coupling capacitance but also simultaneous switching behavior for crosstalk optimization. We first consider a more accurate model, compared with most of the literature, of crosstalk between wire i and wire j:

$$\begin{aligned} crosstalk(i,j) &= switching\_similarity(i,j) \\ &\times \; coupling\_capacitance(i,j). \end{aligned} \tag{1}$$

For this model, we propose a two-stage strategy to minimize the crosstalk in a circuit. In the first stage, using geometry wire ordering, we place the wires with similar switching behavior in closer proximity; this *switching similarity* problem is an NP-hard problem [15]. Therefore, we resort to heuristics to deal with it. In the second stage, we minimize the inter-wire physical coupling capacitance by sizing wires. We formulate the constraints for physical coupling capacitance in a posynomial (positive polynomial) form, which can be optimally solved by Lagrangian relaxation.

The second stage not only deals with the crosstalk problem, but also optimizes area, power and delay by sizing gates and wires. Gate and wire sizing has been extensively studied in the literature for optimizing area, power, and/or delay (e.g., [2, 3, 4], etc). In the previous work, Lagrangian relaxation has been proven to be an effective approach for simultaneous performance optimization [2, 3]; this fact encourages us to adopt the Lagrangian relaxation method for our problem. In this paper, based on Lagrangian relaxation, we present an algorithm that can optimally solve the simultaneous crosstalk, area, power, and delay optimization problem by sizing circuit components. Our algorithm, with linear memory requirement *overall* and linear runtime *per iteration*, is very effective and efficient. For example, for a circuit of 6144 wires and 3512 gates, our algorithm solves the simultaneous optimization problem using only 2.1 MB memory and 47 minute runtime to achieve the precision of within 1% error on a SUN UltraSPARC-I workstation.

2 Problem Description

In this section, we introduce the representation of a circuit and some notation used throughout the paper, present circuit and delay models, and formulate a performance optimization problem.

*The work of Hui-Ru Jiang and Jing-Yang Jou was partially supported by the National Science Council of Taiwan ROC under Grant No. NSC88-2215-E-009-070. Email: huiru@cis.nctu.edu.tw, jyjou@bestmap.ee.nctu.edu.tw

†The work of Yao-Wen Chang was partially supported by National Science Council of Taiwan ROC under Grant No's NSC88-2622-E-009-004 and NSC88-2218-E-009-056. Email: ywchang@cis.nctu.edu.tw

DAC 99, New Orleans, Louisiana
(c) 1999 ACM 1-58113-109-7/99/06..$5.00

2.1 Circuit Representation and Modeling

For a digital circuit, we can partition it into two groups—combinational and sequential parts. We can improve the performance by optimizing the combinational part. Hence, we focus on the combinational circuits. The way we interpret a circuit is similar to that used in [3].

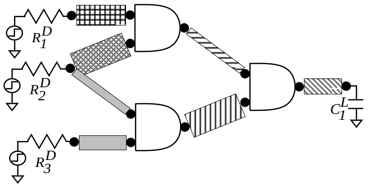

Figure 1: A circuit with three input drivers, seven wires, three gates, and one output load, in which the gate and wire sizes can be varied.

Given a combinational circuit with s primary inputs, t primary outputs and n gates or wires. The sizes of gates and wires can be changed according to our objectives. For the i^{th} primary input, $1 \le i \le s$, we have one corresponding input resistor, R_i^D, as its input driver. Similarly, for the j^{th} primary output, $1 \le j \le t$, we have one corresponding output capacitor, C_j^L, as its output load. Figure 1 depicts an example.

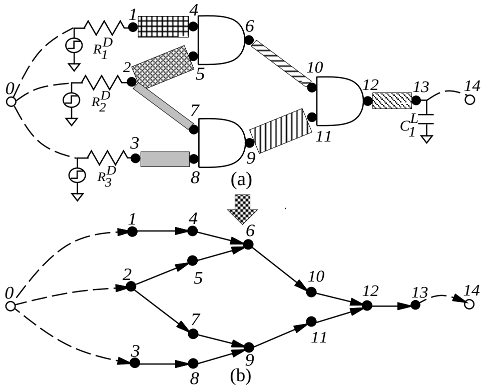

Figure 2: (a) Two artificial nodes, 0 and 14, are added into the circuit depicted in Figure 1. (b) The corresponding circuit graph.

A *component* is a circuit element: a gate, a wire, or an input driver. A node is located at the output of a component, either connecting two components or linking one primary output to one output load. Thus, a circuit has $n + s$ nodes. Figure 2 illustrates the circuit graph of the circuit given in Figure 1. A circuit graph $H = (V, E)$ is a directed acyclic graph with $n + s + 2$ nodes. The set V of nodes consists of two additional artificial nodes as well as $n + s$ nodes corresponding to the $n + s$ components. One added node is viewed as the *source*, $\tilde{s}$, connecting to every input driver, the other as the *sink*, $\tilde{t}$, consisting of all output loads. Let $S = \{\tilde{s}\}$ and $T = \{\tilde{t}\}$. Therefore, the node set $V = G \cup W \cup R \cup S \cup T$ contains the set G of gates, the set W of wires, the set R of input drivers, the set S of source, and the set T of sink. The index of a node is labeled such that if node i is the input of node j, then $i < j$. This indexing can be labeled by topological sorting. Hence, the index of $\tilde{s}$ is 0, and that of $\tilde{t}$ is $n + s + 1$. For $1 \le i \le n + s$, index i refers to a component. On the other hand, the set E of edges represents the connections between nodes. An edge (i, j), an ordered pair, connects node i to node j, $1 \le i < j \le n + s$, if data flow from node i to node j. Additional edges are added to connect $\tilde{s}$ to s input drivers and connect t primary outputs to $\tilde{t}$. The connectivity relationship between parents and children are defined by $input()$ and $output()$, where $input(i) = \{j | (j, i) \in E\}$, and $output(i) = \{j | (i, j) \in E\}$.

Figure 3 illustrates the gate and wire models used in this paper. We choose the π model [12] to approximate wire behavior. For a gate i with size x_i, the resistance r_i is $\hat{r}_i / x_i$, and the capacitance c_i is $\hat{c}_i x_i$, where $\hat{r}_i$ and $\hat{c}_i$ are the resistance and capacitance of gate i with unit size, respectively. For a wire j with size x_j, the resistance r_j is $\hat{r}_j / x_j$, and the capacitance c_j is $\hat{c}_j x_j + f_j$, where $\hat{r}_j$ and $\hat{c}_j$ are the respective resistance and capacitance of wire j with unit size, and f_j is the fringing capacitance of wire j. In addition, for an input driver $i, 1 \le i \le s$, r_i is equal to the input resistor R_i^D.

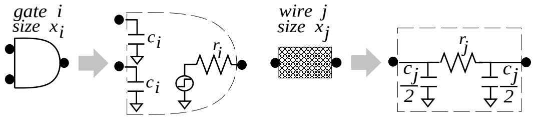

Figure 3: A gate or a wire is modeled as a combination of RC elements. A gate is the loading of its upstream, but is the driver of its downstream. A wire is represented by the π model.

With the circuit model, a combinational circuit can be transformed to a network with resistors and capacitors. Figure 4 illustrates the resulting circuit modeling for the circuit shown in Figure 1. In the transformed circuit, for $1 \le i \le n + s$, $upstream(i)$ is the set with all the nodes except i on the paths from node i to all reachable drivers; similarly, $downstream(i)$ is the set with all the nodes on the paths from node i to all reachable loads. For instance, in Figure 4, $upstream(10) = \{6\}$ and $downstream(2) = \{2, 5, 7\}$. We adopt the Elmore delay model [7] to compute the delays of gates and wires. The delay D_i of node i is $r_i C_i$, where C_i is the downstream capacitance of i including self-loading. In Section 4, C_i also contains the physical coupling capacitance, considering the impact of crosstalk on delay. For the time being, R_i is referred to the upstream resistance of node i, whereas $\mathcal{R}_i$ means the weighted upstream resistance of node i in Section 4.

In the circuit graph H of a circuit, each node i is tagged with some attributes, including size x_i, node type G, W, S or T, unit-width resistance $\hat{r}_i$, unit-width capacitance $\hat{c}_i$ and fringing capacitance f_i ($f_i = 0$ if $i \in G$). Thus, we shall optimize a circuit through manipulating the corresponding circuit graph but ignoring the transformed RC network.

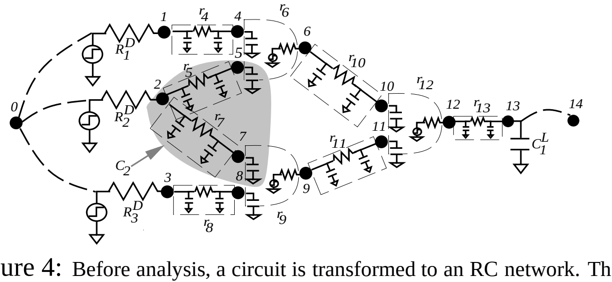

Figure 4: Before analysis, a circuit is transformed to an RC network. The delay D_i lumped in r_i is computed by $r_i C_i$, e.g., $D_2 = R_2^D C_2$, where C_2 represents the capacitance for all the capacitors in the shaded area.

2.2 Problem Description

For practical requirement, area is the greatest concern in circuit design. This paper targets to minimize area subject to noise, timing, and power constraints. Let A, X, D and P denote the total area, the total crosstalk, the delay on the critical path, and the total power of the circuit, respectively, and X^B, D^B and P^B denote the upper bounds of the total crosstalk, the delay on the critical path, and the total power of the circuit, respectively. A generic formulation of this problem is given as follows.

$$\begin{array}{lll} \mathcal{M}: & Minimize & A \\ & Subject\ to & \Pi \leq \Pi^B, \forall\ \Pi \in \{X, D, P\}. \end{array}$$

In Section 4, we will give more detailed problem definitions and present our algorithms for the problem.

3 Crosstalk Modeling

In this section, we will focus on the crosstalk problem. We will deal in turn with the two crucial factors which affect the crosstalk—physical coupling capacitance and switching behavior.

3.1 Physical Coupling Capacitance

We compute the crosstalk between two wires i and j using the model mentioned in Equation (1). Figure 5 depicts a simple case where two parallel wires i and j, belonging to different routing trees, have coupling capacitance.

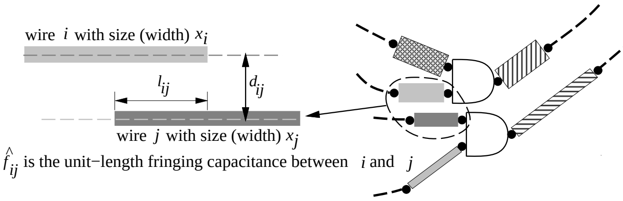
wire i with size (width) x_i
l_{ij}
d_{ij}
wire j with size (width) x_j
$\hat{f}_{ij}$ is the unit−length fringing capacitance between i and j

Figure 5: The physical coupling capacitance between two wires.

According to Figure 5, the physical coupling capacitance c_{ij} between two neighboring wires i and j can be calculated as follows:

$$c_{ij} = \frac{\hat{f}_{ij} l_{ij}}{d_{ij} - \frac{x_i + x_j}{2}} = \left(\frac{\hat{f}_{ij} l_{ij}}{d_{ij}}\right)\left(\frac{1}{1 - \frac{x_i + x_j}{2d_{ij}}}\right), \qquad (2)$$

where x_i and x_j are the sizes of wires i and j ($x_i, x_j > 0$), $\hat{f}_{ij}$ is the unit-length fringing capacitance between wires i and j, l_{ij} is the overlap length of wires i and j, and d_{ij} is the middle-to-middle distance between wires i and j. Equation (2) reflects the impact of wire sizing on crosstalk. If x_i increases, c_{ij} consequently increases. This change would also cause variation on delay. In Equation (2), the first term, $\hat{f}_{ij} l_{ij} / d_{ij}$, is a constant computed by technology files, and the second term, $(1 - (x_i + x_j)/2d_{ij})^{-1}$, is what we are concerned. Let $x = (x_i + x_j)/2d_{ij}$, the second term becomes $(1 - x)^{-1}$, $0 < x < 1$. For the term $(1 - x)^{-1}$, we have the following properties.

Theorem 1 *Let* $f(x) = \frac{1}{1-x}, |x| < 1$.

(1) $f(x) = \sum_{n=0}^{\infty} x^n$*;*

(2) If $\hat{f}(x) = \sum_{n=0}^{k-1} x^n$*, then* $error\ ratio\ \epsilon = \frac{f(x) - \hat{f}(x)}{f(x)} = x^k$.

Theorem 1 says that $(1-x)^{-1}$ can be approximated by $\sum_{n=0}^{k-1} x^n$, the first k terms in the summation. The error ratio is small; for example, for the case $x = 0.25$, the error ratio is less than 6.3%, 1.6%, 0.4%, and 0.1% when k is 2, 3, 4, and 5 respectively. For the purpose of easier presentation, we choose $k = 2$, and thus $f(x) \approx \sum_{n=0}^{1} x^n = 1 + x$. Extensions to a larger k are simple. Therefore, Equation (2) can be approximated as follows:

$$c_{ij} \approx \frac{\hat{f}_{ij} l_{ij}}{d_{ij}}(1 + \frac{x_i + x_j}{2d_{ij}}) = \tilde{c}_{ij}(1 + \frac{x_i + x_j}{2d_{ij}}), \qquad (3)$$

where $\tilde{c}_{ij} = \hat{f}_{ij} l_{ij} / d_{ij}$ is a constant. Note that Equation (3) is in a posynomial form [8], an important property to guarantee the optimality of our algorithm presented in Section 4.

3.2 Switching Behavior

For two adjacent wires with coupling C_c, one is interfered when the other switches. In the worst case, the two wires simultaneously switch in different directions. As a result, the transitions on these wires are longer than expected. This phenomenon, called the Miller effect [1], is like the effect caused by large loading. On the contrary, the anti-Miller effect benefits the transitions. While two neighboring wires toggle in the same direction, they can help each other. Consequently, the transition time is reduced. This phenomenon is like the effect caused by small loading.

Taking advantage of the switching conditions for crosstalk minimization, we shall analyze the switching behavior of signals. The test patterns are available from the logic simulation stage. When analyzing the switching behavior, we first assume each gate or wire is of the minimum size or of other sizes extracted from profiles. Therefore, the similarity of switching behavior between two wires i and j can be defined as follows:

$$similarity(i, j) = \frac{\int_0^{T_D} f(i, t) f(j, t) dt}{T_D},$$

where T_D is the simulation duration, $f(i, t)$ is the normalized waveform of wire i at time t. $f(i, t) = 1$ if node i is high; otherwise, $f(i, t) = -1$ if node i is low. For any two wires i and j, $-1 \leq similarity(i, j) \leq 1$. The closer to -1 for $similarity$, the less similar their behavior; the closer to 1 for $similarity$, the more similar their behavior.

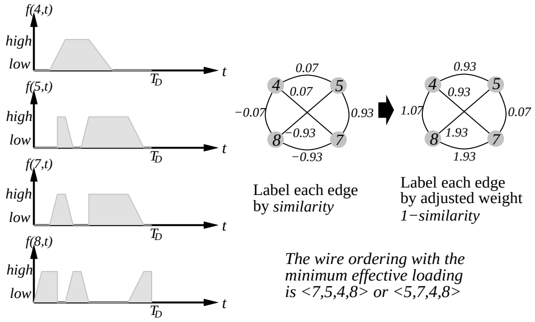
f(4,t)
high
low
t
T_D
f(5,t)
high
low
t
T_D
f(7,t)
high
low
t
T_D
f(8,t)
high
low
t
T_D
0.07
4
5
0.07
−0.07
0.93
−0.93
8
7
−0.93
0.93
4
5
0.93
1.07
0.07
1.93
8
7
1.93
Label each edge by *similarity*
Label each edge by adjusted weight *1−similarity*
The wire ordering with the minimum effective loading is <7,5,4,8> or <5,7,4,8>

Figure 6: The waveforms of wires and the similarity between wires.

Two wires with most similar switching behavior are assigned to closer tracks to minimize the effective loading. The problem for minimizing the effective loading is equivalent to a graph-theoretic one. We build a complete graph K_n for n wires. In K_n, each node i corresponds to a wire i, and every edge (i, j) is associated with a $weight(i, j)$ equal to $1 - similarity(i, j)$. An ordering is a sequence composed of all nodes, $< w_1, w_2, ..., w_n >$. Accordingly, the total effective loading between neighboring wires is $\sum_{i=1}^{n-1} weight(w_i, w_{i+1})$. Hence, the *Switching Similarity* problem $\mathcal{SS}$ is defined in the following.

$$\begin{array}{lrl} \mathcal{SS}: & Given & n\ wires\ and\ their\ switching\ behavior. \\ & Find & an\ ordering\ for\ the\ wires, \\ & such\ that & the\ total\ effective\ loading\ between \\ & & neighboring\ wires\ is\ minimized. \end{array}$$

The MCWO problem [15], which is NP-complete, can be reduced to the $\mathcal{SS}$ problem. Therefore, the $\mathcal{SS}$ problem is NP-hard, and we resort to heuristics. Specifically, we need an approximation algorithm with a performance guarantee. However, we have the negative result shown below.

Theorem 2 *If* $P \neq NP$ *and* $\rho \geq 1$*, there is no polynomial-time approximation algorithm with ratio bound* ρ *for the* $\mathcal{SS}$ *problem.*

```
**Algorithm:** **WOSS** (Wire Ordering for the $\mathcal{SS}$ Problem)
**Input:** the complete graph $K_n$ for $n$ wires
**Output:** A wire ordering $O$
**A1.** $O \leftarrow < w_1, w_2 >$, where $(w_1, w_2) =$ minimum-weighted edge.
**A2. for** $k = 3$ **to** $n$ **do**
        Choose a minimum-weighted edge $(w_{k-1}, j)$,
              $j \notin \{w_1, w_2, ..., w_{k-1}\}$;
        $O \leftarrow < w_1, w_2, ..., w_{k-1}, w_k >$, where $w_k = j$.
```

Figure 7: Wire ordering for the $\mathcal{SS}$ Problem.

We propose an efficient heuristic named **WOSS** for the $\mathcal{SS}$ problem as shown in Figure 7. Basically, the **WOSS** algorithm does the *Depth First Search* for the complete graph K_n in $O(n^2)$ time.

Solving the Switching Similarity problem, we can obtain a geometry ordering for all wires with the minimum effective loading. Therefore, we can know the adjacency relationship between wires. The *neighborhood* $N(i)$ of wire i is defined as the set of adjacent wires; the *dominating index* of $N(i)$, denoted by $I(i)$, of wire i is defined as the set of adjacent wires with the indexes greater than i. For instance, in Figure 6, if we choose $< 5, 7, 4, 8 >$ as the resulting track assignment, $N(5) = \{7\}$, $N(7) = \{5, 4\}$, $N(4) = \{7, 8\}$ and $N(8) = \{4\}$; $I(5) = \{7\}$, $I(7) = \emptyset$, $I(4) = 8$ and $I(8) = \emptyset$.

4 Optimal Area Minimization Under Crosstalk, Delay, and Power Constraints

In this section, we give the problem formulation and an algorithm for simultaneous area, crosstalk, delay, and power optimization. Since area is typically the most important concern in VLSI design, we choose area as the objective function of the optimization problem.

4.1 Problem Formulation

For each component i, $s+1 \leq i \leq n+s$, the corresponding area is proportional to its size x_i. Given the unit-sized area α_i, the area of component i is $\alpha_i x_i$; the total area of a circuit is thus $\sum_{i=s+1}^{n+s} \alpha_i x_i$. The areas of input drivers are ignored, i.e., $x_i = 0$, $1 \leq i \leq s$. This is also true for output loads. If the crosstalk, power, and delay bounds of a circuit are X_B, P_B and A_B, respectively, we have

$$\begin{gathered}\sum_{i \in W} \sum_{j \in I(i)} c_{ij} \leq X_B, \\ V^2 f \sum_{i=s+1}^{n+s} c_i \leq P_B, \\ \sum_{i \in \delta} D_i \leq A_B, \ \forall\ \delta \in \mathbf{\Delta},\end{gathered}$$

where δ is one path of the path set $\mathbf{\Delta}$. Note that, though not presented here, the above crosstalk constraint can easily be extended to the case with a distributed crosstalk bound on each net. The optimization problem we want to solve can be formulated as follows.

$$\begin{array}{lll}\mathcal{P}: & Minimize & \sum_{i=s+1}^{n+s} \alpha_i x_i \\ & Subject\ to & \sum_{i \in \delta} D_i \leq A_B, \forall\ \delta \in \mathbf{\Delta}, \\ & & \sum_{i \in W} \sum_{j \in I(i)} c_{ij} \leq X_B, \\ & & V^2 f \sum_{i=s+1}^{n+s} c_i \leq P_B, \\ & & L_i \leq x_i \leq U_i, \forall\ s+1 \leq i \leq n+s.\end{array}$$

From Section 3.1, the crosstalk between two adjacent wires i and j is their inter-wire physical coupling capacitance, $\tilde{c}_{ij}(1 + (x_i + x_j)/2d_{ij})$. Hence, the crosstalk constraint can be simplified by subtracting both sides by $\sum_{i \in W} \sum_{j \in I(i)} \tilde{c}_{ij}$; the constraint becomes $\sum_{i \in W} \sum_{j \in I(i)} \tilde{c}_{ij}(x_i + x_j)/2d_{ij} \leq X_B - \sum_{i \in W} \sum_{j \in I(i)} \tilde{c}_{ij}$. If we define X_0 as $X_B - \sum_{i \in W} \sum_{j \in I(i)} \tilde{c}_{ij}$ and $\hat{c}_{ij}$ as $\tilde{c}_{ij}/2d_{ij}$, the modified crosstalk constraint is $\sum_{i \in W} \sum_{j \in I(i)} \hat{c}_{ij}(x_i + x_j) \leq X_0$.

Assume the supply voltage V and frequency f are fixed. The power constraint can be simplified by dividing both sides by $V^2 f$. Let P_0 be $P_B/(V^2 f)$. The power constraint becomes $\sum_{i=s+1}^{n+s} c_i \leq P_0$. Because, in deep sub-micron technology, the interconnect densities of a circuit can be very high, the circuit graph could be very dense. Hence, the path set $\mathbf{\Delta}$ can be far greater than, or even grows exponentially with, the circuit size. It is prohibitively expensive to traverse all paths to check the constraints. To conquer this problem, we associate a_i to each node i, which represents the arrival time of that node [3]. Let $m = n + s + 1$ and $A_0 = A_B$ in the following discussion. We have

$$\begin{array}{ll} a_j \leq A_0 & j \in input(m)\ /*\ primary\ outputs\ */ \\ a_j + D_i \leq a_i & i = s+1, ..., n+s\ and\ \forall j \in input(i) \\ D_i \leq a_i & i = 1, ..., s\ /*\ primary\ inputs\ */ \end{array}$$

Consequently, the problem P can be modified as follows.

$$\begin{array}{lll}\mathcal{PP}: & Minimize & \sum_{i=s+1}^{n+s} \alpha_i x_i \\ & Subject\ to & a_j \leq A_0\ \ j \in input(m), \\ & & a_j + D_i \leq a_i\ \ i = s+1, ..., n+s \\ & & \qquad and\ \forall j \in input(i), \\ & & D_i \leq a_i\ \ i = 1, ..., s, \\ & & \sum_{i=s+1}^{n+s} c_i \leq P_0, \\ & & \sum_{i \in W} \sum_{j \in I(i)} \hat{c}_{ij}(x_i + x_j) \leq X_0, \\ & & L_i \leq x_i \leq U_i,\ \ \forall s+1 \leq i \leq n+s.\end{array}$$

The objective function and constraints of the problem $\mathcal{PP}$ are all in the posynomial form. Through variable transformation, a convex programming problem is obtained. Hence, problem $\mathcal{PP}$ has a unique global optimum.

4.2 Lagrangian Relaxation

To solve the problem $\mathcal{PP}$, we apply Lagrangian relaxation by introducing one Lagrange multiplier to each constraint: β to the power constraint, γ to the crosstalk constraint, λ_{ji} to each delay constraint. λ_{ji} can be viewed as a timing weight on edge (j, i). Let $\mathbf{x}$ = $(x_{s+1}, ..., x_{n+s})$ and $\mathbf{a}$ = $(a_1, ..., a_{n+s})$. The Lagrangian function, therefore, is

$$\begin{aligned} L_{\lambda,\beta,\gamma}(\mathbf{x}, \mathbf{a}) &= \sum_{s+1}^{n+s} \alpha_i x_i + \sum_{j \in input(m)} \lambda_{jm}(a_j - A_0) \\ &+ \sum_{i=s+1}^{n+s} \sum_{j \in input(i)} \lambda_{ji}(a_j + D_i - a_i) \\ &+ \sum_{i=1}^{s} \lambda_{0i}(D_i - a_i) + \beta \left(\sum_{i=s+1}^{n+s} c_i - P_0 \right) \\ &+ \gamma \left(\sum_{i \in W} \sum_{j \in I(i)} \hat{c}_{ij}(x_i + x_j) - X_0 \right). \end{aligned}$$

The corresponding Lagrangian relaxation subproblem is

$$\begin{array}{lll}\mathcal{LRS}\mathit{1}: & Minimize & L_{\lambda,\beta,\gamma}(\mathbf{x}, \mathbf{a}) \\ & Subject\ to & L_i \leq x_i \leq U_i, \quad \forall s+1 \leq i \leq n+s.\end{array}$$

To solve the Lagrangian relaxation subproblem, we derive the optimality conditions by the Kuhn-Tucker conditions [16].

Theorem 3 *The optimality conditions on Lagrange multipliers are given by*

$$\sum_{k \in output(i)} \lambda_{ik} \quad = \quad \sum_{j \in input(i)} \lambda_{ji},\ for\ 1 \leq i \leq n+s. \quad (4)$$

Theorem 3 says that the sum of in-degree multipliers equals that of out-degree multipliers for every node except the source. This theorem is analogous to the *Kirchhoff's Current Law* [5]: The algebraic sum of the currents flowing into a node equals that of the currents leaving from the node for all times.

Theorem 4 *For any $\boldsymbol{\lambda}$ satisfying Equation (4) in Theorem 3, solving $\mathcal{LRS}1$ is equivalent to solving*

$$\begin{array}{lll} \mathcal{LRS}2: & Minimize & L_{\mu,\beta,\gamma}(\mathbf{x}) \\ & Subject\ to & L_i \le x_i \le U_i, \quad \forall s+1 \le i \le n+s, \end{array}$$

where $\boldsymbol{\mu} = (\mu_1, ...\mu_m)$, $\mu_i = \sum_{j \in input(i)} \lambda_{ji}$ for $1 \le i \le m$, and

$$\begin{aligned} L_{\mu,\beta,\gamma}(\mathbf{x}) &= \sum_{i=s+1}^{n+s} \alpha_i x_i + \beta \left(\sum_{i=s+1}^{n+s} c_i - P_0 \right) \\ &+ \gamma \left(\sum_{i \in W} \sum_{j \in I(i)} \hat{c}_{ij}(x_i + x_j) - X_0 \right) + \sum_{i=1}^{n+s} \mu_i D_i. \end{aligned}$$

We derive the optimal sizing solution and present a greedy, optimal algorithm to solve the Lagrangian relaxation subproblem $\mathcal{LRS}2$.

Theorem 5 *Let $\tilde{\mathbf{x}} = (\tilde{x}_{s+1}, ..., \tilde{x}_{n+s})$ be a solution, then the optimal resizing of component i is given by $x_i^* = \min(U_i, \max(L_i, opt_i))$, where*

$$opt_i = \sqrt{\frac{\mu_i \hat{r}_i \left(C_i' + \sum_{j \in N(i)} \hat{c}_{ij} x_j \right)}{\alpha_i + (\beta + R_i)\hat{c}_i + \gamma \sum_{j \in N(i)} \hat{c}_{ij}}}.$$

In summary, we have the following theorem.

Theorem 6 $(\mathbf{x}^*, \mathbf{a}^*)$ *is an optimal sizing solution if and only if there exists a vector $\boldsymbol{\lambda}^* = (\lambda_{01}^*, ..., \lambda_{m-1\ m}^*)$, β^*, and γ^* such that*

(1) $\sum_{k \in output(i)} \lambda_{ik}^* = \sum_{j \in input(i)} \lambda_{ji}^*, \ \forall 1 \le i \le n+s;$

(2) $\lambda_{jm}^*(a_j - A_0) = 0, \ \forall j \in input(m);$
$\lambda_{ji}^*(a_j + D_i - a_i) = 0, \ \forall s+1 \le i \le n+s, j \in input(i);$
$\lambda_{0i}^*(D_i - a_j) = 0, \ \forall 1 \le i \le s;$
$\beta^*(\sum_{s+1}^{n+s} c_i - P_0) = 0;$
$\gamma^*(\sum_{i \in W} \sum_{j \in I(i)} \hat{c}_{ij}(x_i^* + x_j^*) - X_0) = 0;$

(3) $a_j^* \le A_0, \ \forall j \in input(m);$
$a_j^* + D_i \le a_i^*, \ \forall s+1 \le i \le n+s;$
$D_i \le a_j^*, \ \forall 1 \le i \le s;$
$\sum_{s+1}^{n+s} c_i \le P_0;$
$\sum_{i \in W} \sum_{j \in I(i)} \le X_0;$

(4) $\lambda_{ji}^* \forall \ge 0, \ 1 \le i \le m, j \in input(i); \beta^* \ge 0; \gamma^* \ge 0;$

(5) $x_i^* = \min(U_i, \max(L_i, opt_i)), s+1 \le i \le n+s, \ where$

$$opt_i = \sqrt{\frac{\mu_i \hat{r}_i \left(C_i' + \sum_{j \in N(i)} \hat{c}_{ij} x_j \right)}{\alpha_i + (\beta + R_i)\hat{c}_i + \gamma \sum_{j \in N(i)} \hat{c}_{ij}}}.$$

In the above theorem, (1) is the optimality condition, (2) reflects the complementary slackness conditions, (3) represents constraints, (4) restricts non-negative multipliers, and (5) is the optimal sizing.

We propose a greedy algorithm **LRS** in Figure 8 to optimally solve the Lagrangian relaxation subproblem $\mathcal{LRS}2$ (and equivalently to solve $\mathcal{LRS}1$). As mentioned earlier, the Lagrangian relaxation problem has a unique global optimum. This property guarantees that a greedy algorithm can find the optimal solution.

The following gives the Lagrangian dual problem.

$$\begin{array}{lll} \mathcal{LDP}: & Maximize & D(\boldsymbol{\lambda}, \beta, \gamma) \\ & Subject\ to & \boldsymbol{\lambda}\ in\ the\ optimal\ condition, where \\ & & D(\boldsymbol{\lambda}, \beta, \gamma) = \min L_{\lambda,\beta,\gamma}(\mathbf{x}, \mathbf{a}). \end{array}$$

If $\boldsymbol{\lambda}$ is the optimal solution of the $\mathcal{LDP}$ problem, then $\boldsymbol{\lambda}$ also optimizes the $\mathcal{PP}$ problem. We present Algorithm **OGWS** listed in Figure 9 to solve $\mathcal{LDP}$.

Theorem 7 *Algorithm* **OGWS** *converges to the global optimal.*

```
Subroutine: LRS (Lagrangian Relaxation Subroutine)
Input: the circuit graph $H$ and Lagrange multipliers $\boldsymbol{\mu}, \beta, \gamma$
Output: x=$(x_{s+1}, ..., x_{n+s})$ which minimizes $L_{\mu,\beta,\gamma}(\mathbf{x})$
S1. $x_i \leftarrow L_i, \forall\ s+1 \le i \le n+s.$
S2. Compute $C_i', \forall\ s+1 \le i \le n+s$
    by traversing $H$ in the reverse topological order.
S3. Compute $R_i, \forall\ s+1 \le i \le n+s$
    by traversing $H$ in the topological order.
S4. for $i = s+1$ to $n+s$ do
        $x_i \leftarrow \min(U_i, \max(L_i, opt_i))$, where
            $opt_i = \sqrt{\frac{\mu_i \hat{r}_i \left( C_i' + \sum_{j \in N(i)} \hat{c}_{ij} x_j \right)}{\alpha_i + (\beta + R_i)\hat{c}_i + \gamma \sum_{j \in N(i)} \hat{c}_{ij}}}.$
S5. Repeat S2-S4 until no improvement.
```

Figure 8: Lagrangian Relaxation Subroutine.

5 Experimental Results

We implemented our algorithm in the C language on a SUN UltraSPARC-I workstation and tested on the ISCAS85 benchmark circuits. The circuit sizes ranged from 640 to 9656. The supply voltage was set to 3.3 V, and the working frequency was set to 200 MHz. The unit-sized resistance and capacitance of a gate were $10\ \Omega \cdot \mu m$ and $0.16\ fF/\mu m$, and those of a wire were $0.07\ \Omega \cdot \mu m$ and $0.024\ fF/\mu m$, respectively. The respective lower and upper bounds for a gate or wire size are $0.1\ \mu m$ and $10\ \mu m$. Table 1 shows the experimental results, where #G denotes the number of gates, #W denotes the number of wires, tot denotes the total number of gates and wires, Init denotes the initial values before sizing, Fin denotes the final values after sizing, ite denotes the number of iterations, time denotes the runtime, mem denotes the memory requirement, and Impr(%) denotes the average improvement in %. The improvement for each term is calculated by $\frac{Init - Fin}{Init} \times 100\%$.

The results show that our algorithm, on the average, improved the respective area, noise, power, and delay by 87.90%, 89.67%, 86.82% and 5.3% after wire and gate sizing. Further, our algorithm is effective and efficient. For example, for the largest circuit, c7552, with 3512 gates and 6144 wires, our algorithm needed only 47 min and 2.1 MB storage to achieve the precision of within 1% error.

Note that the results show that sizing benefits delay not much. When a component is enlarged, it will increase ont only the loading of the components on the upstream path of the sized component and the driving capability for the components on the downstream path but the physical coupling capacitance also. Consequently, up-sizing causes that the delay for the upstream part increases, while the delay for the downstream part decreases. Similarly, down-sizing reduces the delay for the upstream part and harms that for the downstream part. As a result, the delay over the whole circuit would not be significantly improved.

In Figures 10(a) and (b), the storage requirement and runtime per iteration (denoted by the vertical axis) are plotted as functions of the total number of gates and wires in a circuit (represented by the horizontal axis), respectively. Figures 10(a) and (b) show that the runtime per iteration and the storage requirements of our algorithm approach linear in the total number of gates and wires. As revealed by Figure 10, some points deviate from the linear line; a probable reason is that these circuits are not regular and their structures are different from each other.

Ckt Name	Ckt Size			Noise (pF)		Delay (ps)		Power (mW)		Area (um^2)		ite	time (sec)	mem (KB)
	#G	#W	tot	Init	Fin	Init	Fin	Init	Fin	Init	Fin			
c1355	546	1064	1610	20.53	2.14	1005.57	1098.90	228.34	28.45	48299	5203	9	56	1096
c1908	880	1498	2378	24.55	2.45	1444.57	1338.62	357.09	41.45	71338	7369	13	155	1184
c2670	1193	2076	3269	33.46	3.35	1480.65	1499.87	486.38	58.45	98067	10319	7	444	1320
c3540	1669	2939	4608	50.24	5.03	1713.47	1685.51	682.19	79.53	138242	14292	8	553	1472
c432	214	426	640	7.89	.95	1442.28	958.20	89.95	18.35	19200	2984	7	21	976
c499	514	928	1442	16.37	1.72	875.81	799.31	211.25	27.88	43259	4834	10	97	1072
c5315	2307	4386	6693	82.06	8.23	1649.38	1548.37	959.28	113.92	200803	20768	7	1321	1752
c6288	2416	4800	7216	95.36	9.53	4888.33	4494.26	1015.03	129.94	216495	23341	14	2705	1808
c7552	3512	6144	9656	103.30	10.33	1615.32	1619.37	1433.49	168.91	289707	30120	7	2823	2120
c880	383	729	1112	13.12	1.35	931.49	794.43	159.30	22.14	33359	3827	12	94	1032
Impr(%)	-			89.67%		5.3%		86.82%		87.90%		-		

Table 1: Experimental results in noise, delay, power and area.

```
Algorithm: OGWS (Optimal Gate and Wire Sizing)
Input: the circuit graph $H$
Output: $\boldsymbol{\lambda}$, $\beta$, $\gamma$ which maximize $\min L_{\lambda,\beta,\gamma}(\mathbf{x})$
A1. $k \leftarrow 1$;
    $\boldsymbol{\lambda} \leftarrow$ arbitrary vector in the optimality condition;
    $\beta \leftarrow$ an arbitrary positive number;
    $\gamma \leftarrow$ an arbitrary positive number.
A2. $\boldsymbol{\mu} = (\mu_1, ..., \mu_{n+s+1})$, where $\mu_i = \sum_{j \in input(i)} \lambda_{ji}$.
A3. Call LRS and compute $a_1, ...a_{n+s}$.
A4. Adjust multipliers $\lambda_{ji}$'s, $\beta$, $\gamma$:
    for $i = 1$ to $n + s + 1$ do
        forall $j \in input(i)$ do
            $\lambda_{ji} \leftarrow \begin{cases} \lambda_{ji} + \theta_k(a_j - A_0) & \text{if } i \in T \\ \lambda_{ji} + \theta_k(a_j + D_i - a_i) & \text{if } i \in G \cup W \\ \lambda_{ji} + \theta_k(D_i - a_i) & \text{if } i \in R \end{cases}$
            $\beta \leftarrow \beta + \theta_k(\sum_{i=s+1}^{n+s} c_i - P_0)$
            $\gamma \leftarrow \gamma + \theta_k(\sum_{i \in W} \sum_{j \in I(i)} \hat{c}_{ij}(x_i + x_j) - X_0)$
            where the step size $\theta_k$ satisfies $\lim_{k \to \infty} \theta_k = 0$
                and $\sum_{j=1}^{k} \theta_j \to \infty$.
A5. Project $\boldsymbol{\lambda}$ onto the nearest point in the optimality condition.
A6. $k \leftarrow k + 1$.
A7. If $(\sum_{i=s+1}^{n+s} \alpha_i x_i - L_{\lambda,\beta,\gamma}(\mathbf{x})) \geq$ error bound, goto A2.
```

Figure 9: Optimal Gate and Wire Sizing Algorithm.

6 Concluding Remarks

We have modeled the crosstalk optimization problem by considering both of the switching conditions and the physical coupling capacitance. We have proposed a two-stage method for crosstalk minimization: the first stage handles geometry wire ordering by exploiting the switching conditions to reduce the effective loading; the second stage, further, simultaneously optimizes physical coupling capacitance, power, and delay. Based on the Lagrangian relaxation method, our **OWGS** algorithm can economically optimize all the above objectives. The experimental results show that our algorithm is very effective for performance optimization, especially for noise, power, and area minimization.

7 Acknowledgments

The authors would like to thank Dr. Chung-Ping Chen of Intel Corp. for his precious suggestions and kind help.

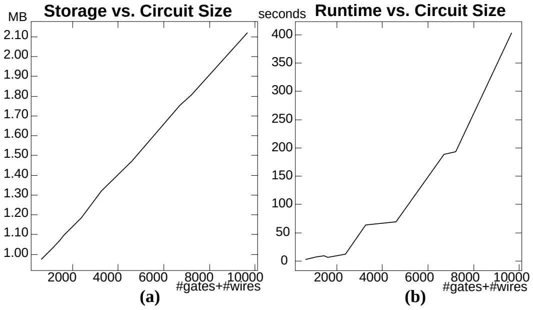

Figure 10: The storage and runtime requirement of our algorithm vs. circuit size.

References

[1] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI,* Addison-Wesley Pub. Company Inc., 1990.

[2] C.-P. Chen, Y.-W. Chang and D. F. Wong, "Fast Performance-Driven Optimization for Buffered Clock Trees Based on Lagrangian Relaxation," *Proc. DAC,* pp. 405–408, June 1996.

[3] C.-P. Chen, C. C. N. Chu and D. F. Wong, "Fast and Exact Simultaneous Gate and Wire Sizing by Lagrangian Relaxation," *Proc. ICCAD,* pp. 617–624, Nov. 1998.

[4] D.-S. Chen and M. Sarrafzadeh, "An Exact Algorithm for Low Power Library-Specific Gate Re-Sizing," *Proc. DAC,* June 1996.

[5] L. O. Chua, C. A. Desoer and E. S. Kuh, *Linear and Nonlinear Circuits,* McGraw-Hill Book Company, 1987.

[6] A. Devgan, "Efficient Coupled Noise Estimation for On-Chip Interconnects," *Proc. ICCAD,* pp. 147–151, Nov. 1997.

[7] W. C. Elmore, "The Transient Response of Damped Linear Networks with Particular Regard to Wide Band Amplifiers," *J. Applied Physics,* 19(1), 1948.

[8] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research,* 5th ed., McGraw-Hill Publishing, 1990.

[9] M. Marek-Sadowska, "Impact of Deep Sub-micron Technologies on Physical Design," Lecture notes and Private Communication, Aug. 1998.

[10] Y. Massoud, S. Majors, T. Bustami and J. White, "Layout Techniques for Minimizing On-Chip Interconnect Self Inductance," *Proc. DAC,* pp. 566–571, June 1998.

[11] M. Nemani and F. N. Najm, "High-Level Area and Power Estimation for VLSI Circuits," *Proc. ICCAD,* pp. 114–119, Nov. 1997.

[12] J. Rabaey, *Digital Integrated Circuits: A Design Perspective,* Prentice-Hall, Inc., 1996.

[13] K. L. Shepard, "Design Methodologies for Noise in Digital Integrated Circuits," *Proc. DAC,* pp. 94–99, June 1998.

[14] H.-P. Tseng, L. Scheffer, and C. Sechen, "Timing and Crosstalk Driven Area Routing," *Proc. DAC,* pp. 378–381, June 1998.

[15] A. Vittal and M. Merek-Sadowska, "Crosstalk Reduction for VLSI," *IEEE Trans. CAD,* pp. 290–298, Vol. 16, No. 3, Mar. 1997.

[16] W. L. Winston, *Operations Research: Applications and Algorithms,* 3rd ed., Int Thomson Publishing, 1994.

[17] T. Xue, E. S. Kuh and D. Wang, "Post Global Routing Crosstalk Risk Estimation and Reduction," *Proc. ICCAD,* pp. 302–309, Nov. 1996.