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

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 do 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 which 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 1.8 MB memory and 47 minute runtime to achieve the precision of within 1% error on a SUN Sparc Ultra-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 [20, 22]. While power, area, and timing have been extensively discussed in the recent literature, e.g., [3, 4, 5, 6, 7, 10, 18, 21], 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 which makes the behavior of a manufactured circuit deviate from the expected response [19]. The deleterious influences of noise can be classified into two categories. One is malfunctioning, which makes the logic values of nodes differ from what we desire; the other is timing change, which is caused by switching behavior.

Generally, crosstalk is a type of noises which is 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. A voltage or a current change on one wire can thus interfere the signal on the other wire. The inductive effects [15, 17] must be considered as circuit frequencies increase above 500 MHz. So far, the typical strategies to minimize the on-chip inductance are shielding wires and/or shielding layers. The inductive 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. Currently existing literature handles only physical coupling capacitance. Miscellaneous heuristics and techniques have been proposed to minimize the overlap length or to maximize the distance between the wires; these methods include track permutation [12, 13] and wire spacing [23, 25], etc.

In fact, coupling capacitance is dominated not only by physical geometry, but also by switching conditions [16]. The influence of switching conditions can be explained by the Miller and the anti-Miller effects [2]. 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. In this case, the equivalent coupling is $2C_c$. On the contrary, the anti-Miller effect happens when the adjacent wires switching in the same direction. In this case, the equivalent coupling is $-2C_c$. In other words, the coupling effect is not always undesirable. In the appearance of the anti-Miller effect, the wires are charged or discharged by the currents from all drivers. Thus, 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 only consider 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 of crosstalk between wire i and wire j:

$$crosstalk(i,j) \quad = \quad switching\_similarity(i,j) \times coupling\_capacitance(i,j).$$

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 equivalent to the minimum-weighted Hamiltonian path problem in a complete graph, which is an NP-hard problem. Therefore, we resort to heuristics for dealing with the switching similarity problem. 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 solved by Lagrangian relaxation optimally.

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., [3, 4, 5, 6, 7], etc). In the previous work, Lagrangian relaxation has been proven to be an effective approach for simultaneous performance optimization [4, 5, 6]; this fact encourages us to adopt the Lagrangian relaxation method for our problem. In this paper, based on Lagrangian relaxation, we present an algorithm which 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 1.8 MB memory and 47 minute runtime to achieve the precision of within 1% error on a SUN Sparc Ultra-I workstation.

The remainder of this paper is organized as follows. Section 2 gives a circuit model and the problem description. The crosstalk modeling is detailed in Section 3, in which coupling capacitance and simultaneous switching are discussed. In Section 4, based on Lagrangian relaxation, we propose an algorithm to minimize the total area under noise, power, and delay constraints. Section 5 shows the experimental results. Concluding remarks are given in Section 6.

2 Circuit Modeling and 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.

2.1 Circuit Representation

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. For example, in order to speed up the working frequency, we have to minimize the clock period. We may achieve this goal by minimizing the delay of the critical path in any combinational subcircuit between two latch elements. Hence, we can focus on the combinational circuits. The way we interpret a circuit is similar to that used in [5].

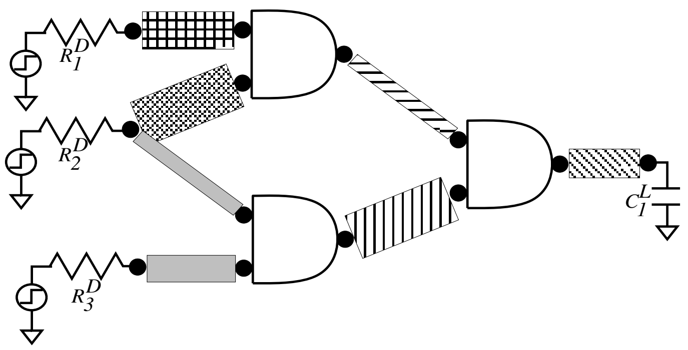

Figure 1: A combinational circuit with three input drivers, seven wires, three gates, and one output load, in which the gate and wire sizes can be varied for optimization.

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, $1 \leq i \leq s$, we have one corresponding input resistor, R_i^D, as its input driver. Similarly, for the j^{th} primary output, $1 \leq j \leq t$, we have one corresponding output capacitor, C_j^L, as its output load. Figure 1 depicts a combinational circuit with three input drivers and one output load.

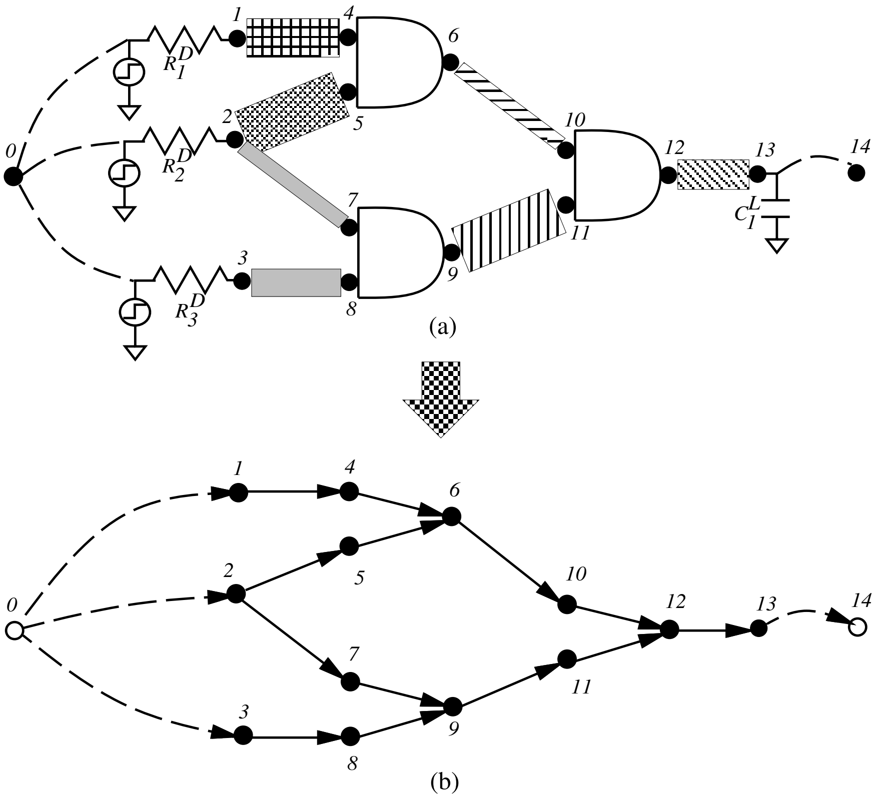

Figure 2: (a) Two artificial nodes, 0 and 14, are added into the circuit depicted in Figure 1. (b) This graph is the corresponding circuit graph.

A *component* is a circuit element which can be a gate, a wire, or an input driver. A node is located at the output of a component, which either connects two components or links one primary output to one output load. Because every node obviously connects to a distinct node, a circuit has $n + s$ nodes. In order to conveniently manipulate the circuit, a circuit graph is constructed. Figure 2 illustrates the circuit graph of the circuit given in Figure 1. A circuit graph $H = (V, E)$ is a directed acyclic graph which contains $n + s + 2$ 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*, S, connecting to every input driver; the other is viewed as the *sink*, T, consisting of all output loads. Therefore, the node set V, $V = G \cup W \cup R \cup \{S\} \cup \{T\}$, contains the set G of gates, the set W of wires, the set R of input drivers, the source S, and the sink T. The index of a node is labeled such that if node i is the input of node j, then $i < j$. For an acyclic directed graph, this indexing can be labeled by topological sorting [9] with runtime linear in the graph size. Hence, the index of the source is 0, and that of the sink is $n + s + 1$. For $1 \leq i \leq n + s$, index i is referred to a gate, a wire, or an input driver. On the other hand, the set E of edges expresses the connections between nodes. An edge (i, j), an ordered pair, connects node i to node j,

$1 \leq i < j \leq n+s$, if data flow from node i to node j. Additional edges are added to connect the source to s input drivers and connect t primary outputs to the sink. The connectivity relationship between parents and children are defined by $input()$ and $output()$, where $input(i) = \{j|(j,i) \in E, 0 \leq j < i \leq n+s+1\}$, and $output(i) = \{j|(i,j) \in E, 0 \leq j < i \leq n+s+1\}$. Furthermore, i belongs to $input(j)$ if and only if j belongs to $output(i)$. For example, in Figure 2, $input(6) = \{4,5\}$, $output(0) = \{1,2,3\}$.

2.2 Circuit and Delay Models

In order to explore a circuit, we shall model the circuit elements by analyzable electric components, like resistors and capacitors. Figure 3 illustrates the gate and wire models used in this paper. We choose the π model [19] 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 $i, 1 \leq i \leq s$, r_i is equal to the input resistor R_i^D.

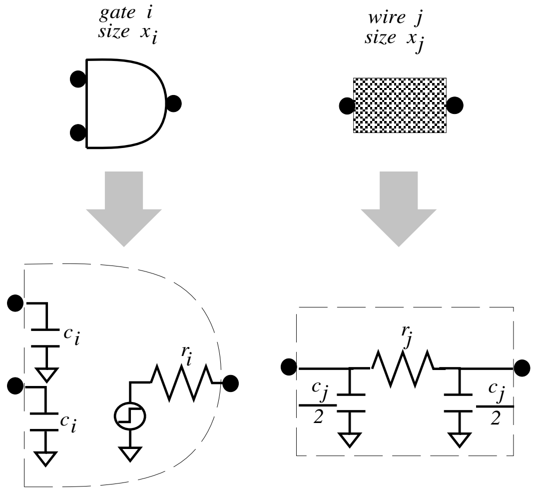

Figure 3: A gate or a wire is modeled as a combination of RC elements. A gate is the loading of its upstream, but is the driver of its downstream. A wire is represented by the π model.

With the gate and wire models, 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 $1 \leq i \leq n+s$, $upstream(i)$ means the proper set (all elements are distinct) containing all the nodes except i on the paths from node i to all reachable drivers; similarly, $downstream(i)$ means the proper set containing 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 [11] 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. For the time being, R_i is referred to the upstream resistance of node i, whereas 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.

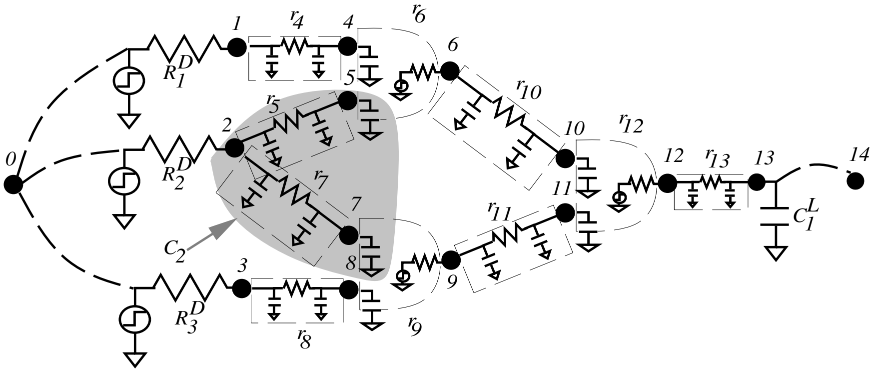

Figure 4: Before being analyzed, a circuit is transformed to an RC network. Hence, the delay D_i lumped in r_i can be computed by r_iC_i. For example, the delay of node 2 is $R_2^DC_2$, where C_2 represents the capacitance for all the capacitors in the shaded area.

2.3 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 bound 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.

$$\begin{aligned} \mathcal{M} : Minimize \quad & A \\ Subject\ to \quad & X \leq X^B, \\ & D \leq D^B, \\ & P \leq P^B. \end{aligned}$$

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

3 Crosstalk Modeling

In the preceding section, we have introduced preliminaries about representing and interpreting a circuit. In this section, we will focus on the crosstalk problem, which has been briefly described in section 1. 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 physical coupling capacitance between two wires i and j using the model mentioned in Section 1:

$$crosstalk(i, j) = switching\_similarity(i, j) \times coupling\_capacitance(i, j).$$

Figure 5 depicts a simple case where two parallel wires i and j, belonging to different routing trees, have coupling capacitance.

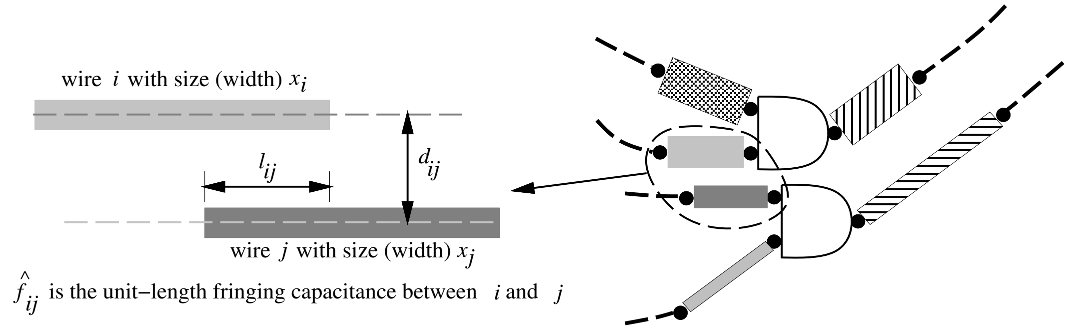

Figure 5: The physical coupling capacitance between two wires.

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

$$\begin{aligned} c_{ij} &= \frac{\hat{f}_{ij} \times l_{ij}}{d_{ij} - \frac{x_i + x_j}{2}} \\ &= \frac{\hat{f}_{ij} \times l_{ij}}{d_{ij}(1 - \frac{x_i + x_j}{2d_{ij}})} \\ &= \frac{\hat{f}_{ij} \times l_{ij}}{d_{ij}} \times \frac{1}{1 - \frac{x_i + x_j}{2d_{ij}}}, \end{aligned} \tag{1}$$

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 distance from the center line of wire i to that of wire j. In Equation (1), the first term, $\frac{\hat{f}_{ij} \times l_{ij}}{d_{ij}}$, is a constant which can be computed by technology files, and the second term, $\frac{1}{1 - \frac{x_i + x_j}{2d_{ij}}}$, is what we are concerned. Let $x = \frac{x_i + x_j}{2d_{ij}}$, the second term of Equation (1) becomes $\frac{1}{1-x}$. Moreover, $0 < x < 1$, since the two wires would *physically overlap*, i.e., short/contact, with each other when $x \geq 1$, and it is impossible that $x = 0$ because x_i and x_j are two positive quantities. For the term $\frac{1}{1-x}$, 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* $\epsilon = \frac{f(x) - \hat{f}(x)}{f(x)} = x^k$*.*

Proof:

(1) For

$$f(x) = \frac{1}{1-x},$$

the n^{th}-order derivative at $x = x_0$ of $f(x)$ is given by

$$f^{(n)}(x_0) = n!(1 - x_0)^{-(n+1)}.$$

Let $x_0 = 0$,

$$f^{(n)}(x_0) \quad = \quad n!.$$

The resulting Taylor series at $x = 0$ associated with the function $f(x)$ is

$$\begin{aligned} f(x) &= \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \\ &= \sum_{n=0}^{\infty} \frac{n!}{n!} x^n \\ &= \sum_{n=0}^{\infty} x^n. \end{aligned}$$

By the absolute-ratio test, we examine

$$\lim_{n \to \infty} \left| \frac{x^{n+1}}{x^n} \right| = |x|.$$

Since $|x| < 1$, the series converges absolutely, hence converges; the radius of convergence is 1.

(2) If $f(x)$ is approximated by the first k terms of the infinite series in *(1)*, we have

$$\hat{f}(x) \quad = \quad \sum_{n=0}^{k-1} x^n = \frac{1 - x^k}{1 - x}.$$

The error is thus

$$e \quad = \quad f(x) - \hat{f}(x) = \frac{x^k}{1 - x},$$

and the resulting error ratio ϵ

$$\epsilon \quad = \quad \frac{e}{f(x)} = x^k.$$

□

Theorem 1 says $\frac{1}{1-x}$ 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 $x = 0.25$,

$$\begin{aligned} k &= 2, \qquad \textit{error ratio} < 6.3\%; \\ k &= 3, \qquad \textit{error ratio} < 1.6\%; \\ k &= 4, \qquad \textit{error ratio} < 0.4\%; \\ k &= 5, \qquad \textit{error ratio} < 0.1\%. \end{aligned}$$

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 larger k are simple. Therefore, Equation (1) can be approximated as follows:

$$\begin{aligned} c_{ij} &\approx \frac{\hat{f}_{ij} \times l_{ij}}{d_{ij}}\left(1 + \frac{x_i + x_j}{2d_{ij}}\right) \\ &= \tilde{c}_{ij}\left(1 + \frac{x_i + x_j}{2d_{ij}}\right), \end{aligned} \qquad (2)$$

where $\tilde{c}_{ij} = \frac{\hat{f}_{ij} \times l_{ij}}{d_{ij}}$ is a constant. Note that Equation (2) is in a posynomial (positive polynomial) form [14]. It will be clear that this is an important property for guaranteeing the optimality of our algorithm to be presented in Section 4.

3.2 Switching Behavior

For two adjacent wires with coupling C_c, when one switches, the current may flow through C_c to the other wire, thus interfering the signal on the other wire. 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 [2], 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.

In order to take 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.

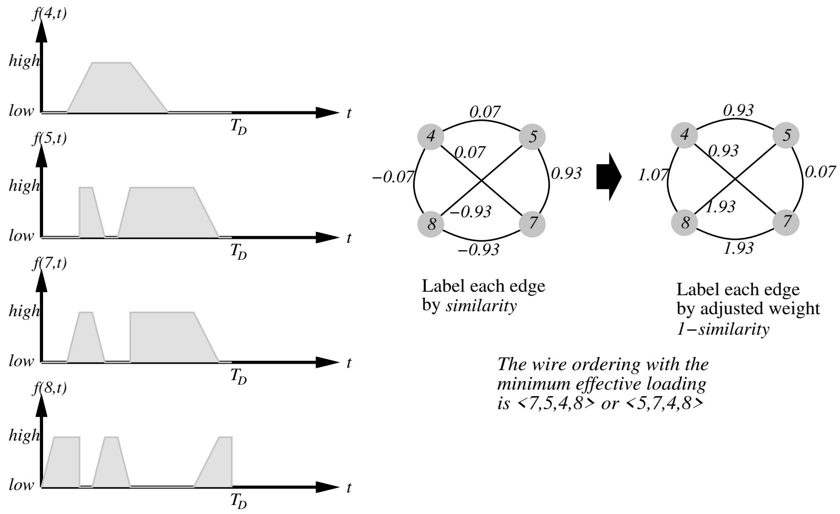

Figure 6: The waveforms of wires and the similarity between each pair of the wires.

Two wires with most similar switching behavior are assigned to closer tracks to minimize the effective loading. We can show that 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.

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

We have the following theorem for the complexity of the $\mathcal{SS}$ problem.

Theorem 2 *The Switching Similarity problem $\mathcal{SS}$ is NP-hard.*

The $\mathcal{SS}$ problem can be reduced from the Hamiltonian path problem, which is NP-hard. The reduction is similar to that from the Hamiltonian cycle problem to the traveling-salesman problem in [9]. We briefly describe the reduction in the following. Given a general graph $G = (V, E)$, the existence of a Hamiltonian path in G is hard. We construct a complete graph $G' = (V, E')$ by adding all non-edges of G, thus $E' = \{(i,j)|i,j \in V\}$. In addition, the weight of each edge (i,j) is assigned as follows.

$$weight(i,j) = \begin{cases} 1 & \text{if } (i,j) \in E \\ 2 & \text{if } (i,j) \notin E \end{cases}$$

It can be seen that G has a Hamiltonian path if and only if the minimum total effective loading of the ordering in G' is $n-1$. Therefore, the $\mathcal{SS}$ problem is NP-hard.

Since the $\mathcal{SS}$ problem is NP-hard, we resort to heuristics. Specifically, we need an approximation algorithm with a performance guarantee. However, we have a negative result described in the following theorem.

Theorem 3 *If $P \neq NP$ and $\rho \geq 1$, there is no polynomial-time approximation algorithm with ratio bound ρ for the $\mathcal{SS}$ problem.*

The above theorem can be proved by contradiction. The details of its proof is similar to that of no polynomial-time approximation algorithm with the traveling-salesman problem [9].

By the above two theorems, the $\mathcal{SS}$ problem is NP-hard and there exists no efficient approximation algorithm. We propose an efficient heuristic 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.

Algorithm: WOSS (Wire Ordering for the Switching Similarity 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) is the 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: The heuristic of Wire Ordering for the Switching Similarity Problem.

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 these four wires are routed in the same channel, the geometry ordering is equivalent to track assignment. 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 formulate the performance optimization problem as to minimize the total area of a circuit subject to crosstalk, delay and power constraints.

We summarize the Lagrangian relaxation method here [1]. Consider the following generic linear optimization method formulated in terms of a vector x of decision variables:

$$\begin{array}{ll} Minimize & cx \\ Subject\ to & Ax \leq b, \\ & x \in X. \end{array}$$

The decision variables x lie in a given constraint set X. The Lagrangian relaxation method relaxes the set of constraints $Ax \leq b$ to the objective function by introducing *Lagrange multipliers*, λ, resulting in the *Lagrangian subproblem*:

$$\begin{array}{ll} Minimize & cx + \lambda(Ax - b) \\ Subject\ to & x \in X. \end{array}$$

Since the constraints are relaxed, the Lagrangian subproblem is easier to be solved. By the *Lagrangian Bounding Principle* [1], the *Lagrangian function* $L(\lambda) = \min\{cx + \lambda(Ax - b) : x \in X\}$ is always a lower bound on the optimal objective function value of the original problem. Lagrangian relaxation method can solve a problem optimally when all of the constraints are in $Ax \leq b$ or in $Ax = b$ form, and the objective and constraints are in a *posynomial* [14] form.

Section 4.1 formulates the primal problem in the linear programming form. Section 4.2 relaxes the primal problem to a Lagrangian relaxation problem and simplifies the relaxed problem. We demonstrate how to solve the corresponding Lagrangian relaxation subproblem in Section 4.3. In Section 4.4, we present the Lagrangian dual problem and solve it by the sub-gradient optimization technique.

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 occupied by input drivers are ignored because their areas are fixed, 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

$$\sum_{i \in W} \sum_{j \in I(i)} c_{ij} \leq X_B,$$

$$V^2 f \sum_{i=s+1}^{n+s} c_i \le P_B,$$

$$\sum_{i\in\delta} D_i \le A_B,\ \forall\, \delta \in \mathbf{\Delta},$$

where δ is one path of the path set $\mathbf{\Delta}$. Therefore, the optimization problem we want to solve can be formulated as follows.

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

From Section 3.1, the crosstalk between two adjacent wires i and j is their inter-wire physical coupling capacitance, $\tilde{c}_{ij}(1+\frac{x_i+x_j}{2d_{ij}})$, where $\tilde{c}_{ij}$ is a constant. 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

$$\sum_{i\in W}\sum_{j\in I(i)} \tilde{c}_{ij}(\frac{x_i+x_j}{2d_{ij}}) \le X_B - \sum_{i\in W}\sum_{j\in I(i)} \tilde{c}_{ij}.$$

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 $\frac{\tilde{c}_{ij}}{2d_{ij}}$, the modified crosstalk constraint is

$$\sum_{i\in W}\sum_{j\in I(i)} \hat{c}_{ij}(x_i+x_j) \le X_0.$$

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 $\frac{P_B}{V^2 f}$. The power constraint becomes

$$\sum_{i=s+1}^{n+s} c_i \le P_0.$$

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. This technique was also used in [5]. Therefore, we distribute the delay constraint into each edge in the circuit graph H. Let $m = n+s+1$ and $A_0 = A_B$ in the following discussion. We have

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

Consequently, the problem P can be modified as follows.

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

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. It is known that a convex programming problem has a unique global optimum [14]. Hence, problem $\mathcal{PP}$ has a unique global optimum, and it is ensured that each local optimum is the 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

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

The corresponding Lagrangian relaxation subproblem is

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

To solve the Lagrangian relaxation subproblem, we derive the optimality conditions by Kuhn-Tucker conditions [24].

Theorem 4 *The optimality conditions on Lagrange multipliers are given by*

$$
\sum_{k \in output(i)} \lambda_{ik} \quad = \quad \sum_{j \in input(i)} \lambda_{ji}, \ \mathit{for}\ 1 \leq i \leq n+s. \tag{3}
$$

Proof: By Kuhn-Tucker conditions [24], if the optimal solution $(\mathbf{x}^*, \mathbf{a}^*)$ of the Lagrangian relaxation subproblem $\mathcal{LRS}\mathit{2}$ is the optimal solution of primal problem $\mathcal{PP}$, then $(\mathbf{x}^*, \mathbf{a}^*)$ must satisfy

$$
\frac{\partial L_{\lambda,\beta,\gamma}}{\partial a_i}(\mathbf{x}^*, \mathbf{a}^*) = \mathbf{0}.
$$

Inspired by the work [5] on the optimality conditions for Lagrange multipliers, we have the following by rearranging $L_{\lambda,\beta,\gamma}$:

$$
L_{\lambda,\beta,\gamma} \quad = \quad \sum_{i=1}^{n+s} \left(\sum_{k \in output(i)} \lambda_{ik} - \sum_{j \in input(i)} \lambda_{ji} \right) a_i
$$

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

By checking Kuhn-Tucker conditions, this theorem thus follows. □

Theorem 4 says the sum of in-degree multipliers equals to that of out-degree multipliers for every node except the source. This theorem is analogous to the *Kirchhoff's Current Law* [8]: The algebraic sum of the currents flowing into a node equals that of the currents leaving from the node for all times.

Theorem 5 *For any λ satisfying Equation (3) in Theorem 4, 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 $\mu = (\mu_{\mathbf{1}}, \ldots \mu_{\mathbf{m}})$, $\mu_i = \sum_{j \in input(i)} \lambda_{ji}$ for $1 \le i \le m$, and

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

Proof: Applying the optimality condition, we get

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

Since $\mu_i = \sum_{j \in input(i)} \lambda_{ji}$, $1 \le i \le m$, we have

$$
\begin{aligned}
L_{\lambda,\beta,\gamma} &= \sum_{i=s+1}^{n+s} \alpha_i x_i + \beta \left(\sum_{i=s+1}^{n+s} c_i - P_0 \right) + \gamma \left(\sum_{i \in W} \sum_{j \in I(i)} \hat{c}_{ij}(x_i + x_j) - X_0 \right) \\
&- \mu_0 A_0 + \sum_{i=1}^{n+s} \mu_i D_i.
\end{aligned}
$$

For a fixed multiplier vector, $\mu_m A_0$ is a constant. Further, minimizing $L_{\lambda,\beta,\gamma}$ equals minimizing

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

□

4.3 Lagrangian Relaxation Subproblem

In the preceding subsection, we have obtained the Lagrangian relaxation subproblem $\mathcal{LRS}2$. In this subsection, we will derive the optimal sizing solution and present a greedy, optimal algorithm to solve this subproblem.

Theorem 6 *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*

$$x_i^* = \min\left(U_i, \max\left(L_i, \sqrt{\frac{\mu_i \hat{r}_i C_i'}{\alpha_i + (\beta + R_i)\hat{c}_i + \gamma \sum_{j \in N(i)} \hat{c}_{ij}}}\right)\right).$$

Proof: Define C_i' as follows:

$$C_i' = \begin{cases} C_i - \frac{\hat{c}_i x_i}{2} & \text{if } i \in W; \\ C_i & \text{otherwise.} \end{cases}$$

We have

$$\begin{aligned}
L_{\mu,\beta,\gamma} &= \sum_{i=s+1}^{n+s} \alpha_i x_i + \beta\left(\sum_{i=s+1}^{n+s} c_i - P_0\right) + \gamma\left(\sum_{i \in W}\sum_{j \in I(i)} \hat{c}_{ij}(x_i + x_j) - X_0\right) + \sum_{i=1}^{n+s} \mu_i D_i \\
&= \sum_{i=s+1}^{n+s} \alpha_i x_i + \beta\left(\sum_{i=s+1}^{n+s} c_i - P_0\right) + \gamma\left(\sum_{i \in W}\sum_{j \in I(i)} \hat{c}_{ij}(x_i + x_j) - X_0\right) \\
&\quad + \sum_{i \in G} \mu_i r_i C_i' + \sum_{i \in W} \mu_i r_i \left(C_i' + \frac{\hat{c}_i x_i}{2}\right) + \sum_{i=1}^{s} \mu_i r_i C_i' \\
&= \sum_{i=s+1}^{n+s} \alpha_i x_i + \beta\left(\sum_{i=s+1}^{n+s} c_i - P_0\right) + \gamma\left(\sum_{i \in W}\sum_{j \in I(i)} \hat{c}_{ij}(x_i + x_j) - X_0\right) \\
&\quad + \sum_{i=1}^{n+s} \mu_i r_i C_i' + \sum_{i \in W} \frac{\mu_i \hat{r}_i \hat{c}_i}{2}. \qquad (4)
\end{aligned}$$

We extract the terms dependent on x_i as follows. Let R_i be a weighted upstream resistance:

$$R_i = \sum_{j \in upstream(i)} \mu_j r_j.$$

Rewriting Equation (4), we have

$$\begin{aligned}
L_{\mu,\beta,\gamma}(x_i) &= \sum_{i=s+1}^{n+s} \alpha_i x_i + \beta\left(\sum_{i=s+1}^{n+s} c_i - P_0\right) + \gamma\left(\sum_{i \in W}\sum_{j \in I(i)} \hat{c}_{ij}(x_i + x_j) - X_0\right) \\
&\quad + \sum_{i=1}^{n+s} \mu_i r_i C_i' + \sum_{i \in W} \frac{\mu_i \hat{r}_i \hat{c}_i}{2} \\
&= \alpha_i x_i + \beta \hat{c}_i x_i + \gamma\left(\sum_{j \in N(i)} \hat{c}_{ij}\right) x_i \\
&\quad + \sum_{j \in upstream(i)} \mu_j r_j (\hat{c}_i x_i) + \frac{\mu_i \hat{r}_i C_i'}{x_i} + \textit{terms independent of } x_i \\
&= \left(\alpha_i + \beta \hat{c}_i + \gamma\left(\sum_{j \in N(i)} \hat{c}_{ij}\right)\right) x_i + \frac{\mu_i \hat{r}_i C_i'}{x_i} + \textit{terms independent of } x_i.
\end{aligned}$$

The minimum occurs when

$$\frac{\partial L_{\mu,\beta,\gamma}}{\partial x_i} = 0.$$

Therefore,

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

Considering the upper bound and lower bound of x_i, we have the optimal re-sizing for component i:

$$x_i^* = min\left(U_i, max\left(L_i, \sqrt{\frac{\mu_i \hat{r}_i C'_i}{\alpha_i + (\beta + R_i)\hat{c}_i + \gamma \sum_{j \in N(i)} \hat{c}_{ij}}}\right)\right).$$

□

In summary, we have the following theorem.

Theorem 7 $(\mathbf{x}^*, \mathbf{a}^*)$ *is an optimal sizing solution if and only if there exists a vector* $\lambda^* = (\lambda^*_{01}, ..., \lambda^*_{m-1\ m})$, β^*, *and* γ^* *such that*

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

(2) $\lambda^*_{jm}(a_j - A_0) = 0,\ \forall j \in input(m);$

(3) $\lambda^*_{ji}(a_j + D_i - a_i) = 0,\ \forall s+1 \leq i \leq n+s;$

(4) $\lambda^*_{0i}(D_i - a_j) = 0,\ \forall 1 \leq i \leq s;$

(5) $\beta^*(\sum_{s+1}^{n+s} c_i - P_0) = 0;$

(6) $\gamma^*(\sum_{i \in W} \sum_{j \in I(i)} \hat{c}_{ij}(x_i^* + x_j^*) - X_0) = 0;$

(7) $a_j^* \leq A_0,\ \forall j \in input(m);$

(8) $a_j^* + D_i \leq a_i^*,\ \forall s+1 \leq i \leq n+s;$

(9) $D_i \leq a_j^*,\ \forall 1 \leq i \leq s;$

(10) $\sum_{s+1}^{n+s} c_i \leq P_0;$

(11) $\sum_{i \in W} \sum_{j \in I(i)} \leq X_0;$

(12) $\lambda^*_{ji} \geq 0,\ 1 \leq i \leq m;$

(13) $\beta^* \geq 0;$

(14) $\gamma^* \geq 0;$

(15) $x_i^* = \min\left(U_i, \max\left(L_i, \sqrt{\frac{\mu_i \hat{r}_i C'_i}{\alpha_i + (\beta + R_i)\hat{c}_i + \gamma \sum_{j \in N(i)} \hat{c}_{ij}}}\right)\right), s+1 \leq i \leq n+s.$

In the above theorem, (1) is the optimality condition; (2)–(6) are the complementary slackness conditions; (7)–(11) are constraints; (12)–(14) restrict non-negative multipliers; (15) 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. In other words, if we find a local optimum, this local optimum equals the global optimum. This property guarantees that a greedy algorithm can find the optimal solution.

Theorem 8 *Subroutine* **LRS** *runs in* $O(n)$ *time per iteration using* $O(n)$ *storage, where* n *is the number of gates and wires.*

Subroutine: LRS (Lagrangian Relaxation Subroutine)
Input: the circuit graph H and Lagrange multipliers μ, β, γ
Output: $\mathbf{x}$=($x_{s+1}, ..., x_{n+s}$) which minimizes $L_{\mu,\beta,\gamma}(\mathbf{x})$
S1. $x_i \leftarrow L_i$, $\forall\ s+1 \leq i \leq n+s$.
S2. Compute C_i', $\forall\ s+1 \leq i \leq n+s$
by traversing H in the reverse topological order.
S3. Compute R_i, $\forall\ s+1 \leq i \leq n+s$
by traversing H in the topological order.
S4. for $i = s+1$ **to** $n+s$ **do**

$$x_i \leftarrow \min\left(U_i, \max\left(L_i, \sqrt{\frac{\mu_i \hat{r}_i C_i'}{\alpha_i + (\beta + R_i)\hat{c}_i + \gamma \sum_{j \in N(i)} \hat{c}_{ij}}}\right)\right).$$

S5. Repeat **S2-S4** until no improvement.

Figure 8: Lagrangian Relaxation Subroutine.

4.4 Lagrangian Dual Problem

The following gives the Lagrangian dual problem.

$$\mathcal{LDP}: \quad \begin{array}{ll} \textit{Maximize} & D(\lambda, \beta, \gamma) \\ \textit{Subject to} & \lambda \textit{ in optimal condition}, \end{array}$$

where

$$D(\lambda, \beta, \gamma) = \min L_{\lambda,\beta,\gamma}(\mathbf{x}, \mathbf{a}).$$

We present Algorithm **OGWS** listed in Figure 9 to solve $\mathcal{LDP}$. Initially, an arbitrary multiplier vector in the optimality condition is chosen as the initial one and β, γ are assigned to positive numbers in **A1**. In **A2**, μ are calculated with respect to λ in **A1**. **A3** calls the **LRS** subroutine. In **A4**, the **OGWS** algorithm iteratively adjusts the multipliers by the sub-gradient optimization method. It is well-known that if the step-size sequence $< \theta_k >$ satisfies the condition $\lim_{k\to\infty} \theta_k = 0$ and $\sum_{k=1}^{\infty} \theta_k = \infty$ (e.g., $\theta_k = \frac{1}{k}$), the sub-gradient optimization method will always converge to the global optimal. In **A5**, the updated Lagrange multipliers are projected onto the nearest point in the optimality condition. **A6** updates the iteration counter, while **A7** checks if the stop criteria holds.

Theorem 9 *Algorithm* **OGWS** *converges to the global optimal.*

5 Experimental Results

We implemented our algorithm in the C language on a Sun SPARC Ultra-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. Listed in Table 1, 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 μm and 10 μm. Table 2 shows the experimental results, where #G denotes the number of gates, #W denotes the number of wires,

Algorithm: OGWS (Optimal Gate and Wire Sizing)
Input: the circuit graph H
Output: λ which maximize min.$L_{\lambda,\beta,\gamma}(\mathbf{x})$
A1. $k \leftarrow 1$;
$\lambda \leftarrow$ arbitrary vector in the optimality condition;
$\beta \leftarrow$ an arbitrary positive number;
$\gamma \leftarrow$ an arbitrary positive number.
A2. $\mu = (\mu_1, ..., \mu_{n+s+1})$. where $\mu_i = \sum_{j \in input(i)} \lambda_{ji}$.
A3. Call **LRS**;
Compute $a_1, ... a_{n+s}$.
A4. Adjust multipliers λ_{ji}'s, β, γ:
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 θ_k satisfies $\lim_{k\to\infty} \theta_k = 0$ and $\sum_{j=1}^{k} \theta_j \to \infty$.
A5. Project λ onto the nearest point in optimality condition.
A6. $k \leftarrow k+1$.
A7. Repeat **A2-A6** until $(\sum_{i=s+1}^{n+s} \alpha_i x_i - L_{\lambda,\beta,\gamma}(\mathbf{x})) \leq$ error bound.

Figure 9: Optimal Gate and Wire Sizing Algorithm.

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 1.8 MB storage to achieve the precision of within 1% error.

In Figure 10, the storage requirement (denoted by the vertical axis) is plotted as a function of the total number of gates and wires in a circuit (represented by the horizontal axis). Similarly, the relationship between the runtime per iteration and the circuit size is depicted in Figure 11. Figures 10 and 11 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 11, 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.

6 Concluding Remarks

Noise immunity is of significant importance for a deep sub-micron digital circuit; it, as well as area, delay, and power, has become an important design metric. Switching conditions and coupling capacitance are

	$\hat{c}_i(fF/\mu m)$	$\hat{r}_i(\Omega \cdot \mu m)$	$L_i(\mu m)$	$U_i(\mu m)$
gate	0.16	10	0.1	10
wire	0.024	0.07	0.1	10

Table 1: RC parameters and size bounds of a gate or a wire.

Ckt	Ckt Size			Noise(pF)		Delay(ps)		Power(mW)		Area(um^2)		ite	time	mem
Name	#G	#W	tot	Init	Fin	Init	Fin	Init	Fin	Init	Fin		(sec)	(KB)
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 2: Experimental results in noise, delay, power and area.

two dominating considerations for crosstalk optimization; nevertheless, the switching condition is often neglected in previous work. We have modeled the crosstalk optimization problem by considering both of switching conditions and 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, optimizes not only physical coupling capacitance but also area, power, and delay. Based on the Lagrangian relaxation method, our simultaneous gate and wire sizing 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.

References

[1] R. K. Ahuja, T. L. Magnati and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications,* Prentice Hall, 1993.

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

[3] O. Coudert, "Gate Sizing: a General Purpose Optimization Approach," *Proc. European Design and Test Conference,* Paris, March 1996.

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

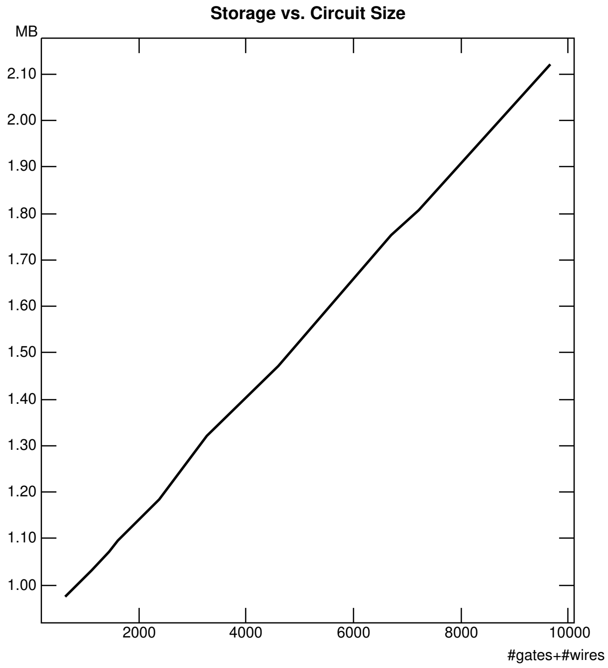

Figure 10: The storage requirement of our algorithm vs. circuit size.

[5] C.-P. Chen, C. C. N. Chu and D. F. Wong, "Fast and Exact Simultaneous Gate and Wire Sizing by Lagrangian Relaxation," *Proc. Int Conference on Computer-Aided Design,* Santa Clara, CA, November 1998.

[6] C.-P. Chen and D. F. Wong, "Optimal Wire-Sizing Function with Fringing Capacitance Consideration," *Proc. Design Automation Conference,* pp. 604–607, Anaheim, CA, June 1997.

[7] D.-S. Chen and M. Sarrafzadeh, "An Exact Algorithm for Low Power Library-Specific Gate Re-Sizing," *Proc. Design Automation Conference,* Las Vegas, NV, June 1996.

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

[9] T. H. Cormen, C. E. Leiserson and R. L. Rivest, *Introduction to Algorithms,* The MIT Press, 1990.

[10] H. Eisenmann and F. M. Johannes, "Generic Global Placement and Floorplanning," *Proc. Design Automation Conference,* pp. 269–274, San Francisco, CA, June 1998.

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

[12] T. Gao and C. L. Liu, "Minimum Crosstalk Channel Routing," *Proc. Int Conference on Computer-Aided Design,* pp. 692–696, Santa Clara, CA, November 1993.

[13] T. Gao and C. L. Liu, "Minimum Crosstalk Switchbox Routing," *Proc. Int Conference on Computer-Aided Design,* pp. 610–615, San Jose, CA, November 1994.

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

[15] M. Lee, A. Hill and M. H. Darley, "Interconnect Inductance Effects on Delay and Crosstalks for Long On-Chip Nets with Fast Input Slew Rates," *Proc. Int Symp. on Circuits and Systems,* Monterey, CA, May 1998.

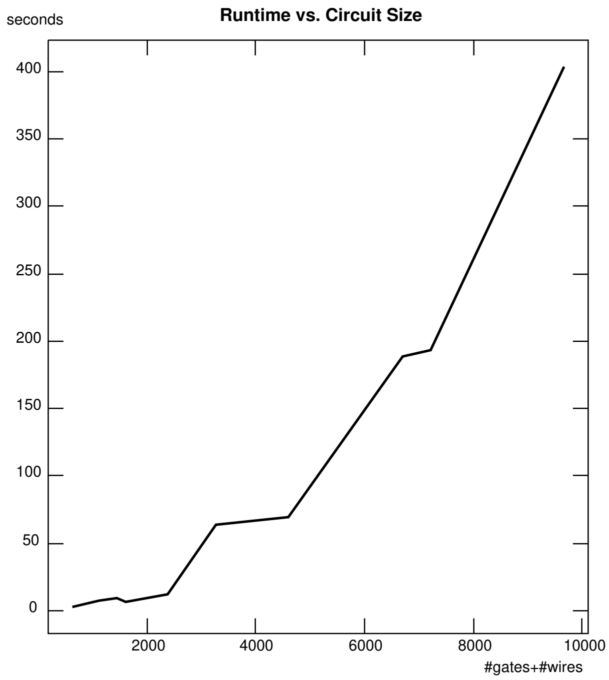

Figure 11: The runtime requirement of our algorithm vs. circuit size.

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

[17] Y. Massoud, S. Majors, T. Bustami and J. White, "Layout Techniques for Minimizing On-Chip Interconnect Self Inductance," *Proc. Design Automation Conference,* pp. 566–571, San Francisco, CA, June 1998.

[18] M. Nemani and F. N. Najm, "High-Level Area and Power Estimation for VLSI Circuits," *Proc. Int Conference on Computer-Aided Design,* pp. 114–119, San Jose, CA, November 1997.

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

[20] K. L. Shepard, "Design Methodologies for Noise in Digital Integrated Circuits," *Proc. Design Automation Conference,* pp. 94–99, San Francisco, CA, June 1998.

[21] M. Sarrafzadeh, D. Knol, and G. Tellez, "Unification of Budgeting and Placement," *Proc. Design Automation Conference,* pp. 758–761, Anaheim, CA, June 1997.

[22] K. L. Shepard and V. Narayanan, "Conquering Noise in Deep-Submicron Digital ICs," *IEEE Design & Test of Computers,* pp. 51–62, Jannuary-March, 1998.

[23] H.-P. Tseng, L. Scheffer, and C. Sechen, "Timing and Crosstalk Driven Area Routing," *Proc. Design Automation Conference,* pp. 378–381, San Francisco, CA, June 1998.

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

[25] T. Xue, E. S. Kuh and D. Wang, "Post Global Routing Crosstalk Risk Estimation and Reduction," *Proc. Int Conference on Computer-Aided Design,* pp. 302–309, San Jose, CA, November 1996.

[26] G. Yee, R. Chandra, V. Ganesan, and C. Sechen, "Wire Delay in the Presence of Crosstalk," *Int Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (TAU),* pp. 170–175, December 1997.