

Crosstalk-Driven Interconnect Optimization by Simultaneous Gate and Wire Sizing

Iris Hui-Ru Jiang, Yao-Wen Chang, *Member, IEEE*, and Jing-Yang Jou, *Member, IEEE*

Abstract—Noise, as well as area, delay, and power, is one of the most important concerns in the design of deep submicrometer integrated circuits. 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 and linear runtime, 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 19.4-min runtime to achieve the precision of within 1% error on a SUN Sparc Ultra-I workstation.

Index Terms—Deep submicrometer, gate sizing, interconnect, performance optimization, physical design, routing.

I. 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 [22], [23]. While power, area, and timing have been extensively discussed in the recent literature, e.g., [3]–[7], [10], [18], and [20], relatively less work has been done on noise.

Noise profoundly affects the performance of a circuit, especially in the deep submicrometer 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 [21], [24], [26], 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 zero. 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 :

$$\begin{aligned} \text{crosstalk}(i, j) &= \text{switching_dissimilarity}(i, j) \\ &\quad \times \text{coupling_capacitance}(i, j). \end{aligned}$$

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 Dissimilarity* 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 Dissimilarity

Manuscript received February 3, 1999; revised January 1, 2000. The work of I. H.-R. Jiang and J.-Y. Jou was supported in part by the National Science Council of Taiwan ROC under Grant NSC88-2215-E-009-070. The work of Y.-W. Chang was supported in part by National Science Council of Taiwan ROC under Grants NSC88-2622-E-009-004 and NSC88-2218-E-009-056. This paper was recommended by Associate Editor M. Sarrafzadeh.

I. H.-R. Jiang and J.-Y. Jou are with the Department of Electronics Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan.

Y.-W. Chang is with the Department of Computer and Information Science, National Chiao Tung University, Hsinchu 30010, Taiwan.

Publisher Item Identifier S 0278-0070(00)07475-3.

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 [14], which can optimally be 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., [3]–[7], etc. In the previous work, Lagrangian relaxation has been proven to be an effective approach for simultaneous performance optimization [4]–[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 and linear runtime, 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 19.4-min. 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 II gives a circuit model and the problem description. The crosstalk modeling is detailed in Section III, in which coupling capacitance and simultaneous switching are discussed. In Section IV, based on Lagrangian relaxation, we propose an algorithm to minimize the total area under noise, power, and delay constraints. Section V shows the experimental results. Concluding remarks are given in Section IV.

II. 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.

A. 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].

Given a combinational circuit with s primary inputs, t primary outputs and n gates/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. Fig. 1 depicts a combinational circuit with three input drivers and one output load.

A *component* is a circuit element which can be a gate, a wire, or an input driver. An input driver is considered as a gate. A

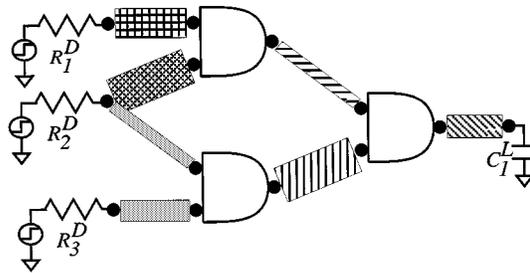


Fig. 1. A combinational circuit with three input drivers, seven wires, three gates, and one output load, in which the gate and wire sizes can be varied for optimization.

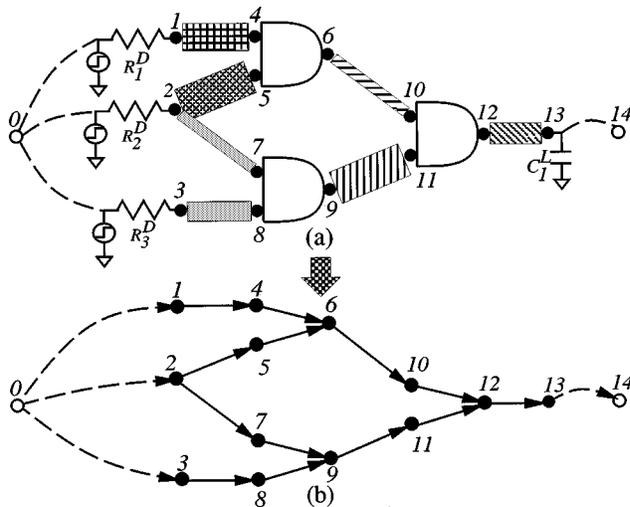


Fig. 2. (a) Two artificial nodes, 0 and 14, are added into the circuit depicted in Fig. 1. (b) The corresponding circuit graph.

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. Fig. 2 illustrates the circuit graph of the circuit given in Fig. 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*, \tilde{s} , connected to every input driver; the other is viewed as the *sink*, \tilde{t} , linked to all output loads. Let $S = \{\tilde{s}\}$ and $T = \{\tilde{t}\}$. 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 zero, 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

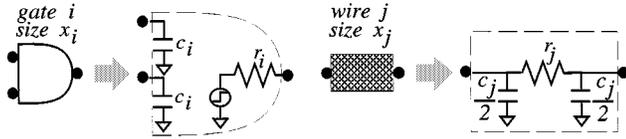


Fig. 3. A gate or a wire is modeled as a combination of resistance–capacitance (RC) elements. A gate is the loading of its upstream, but is the driver of its downstream. A wire is represented by the π model.

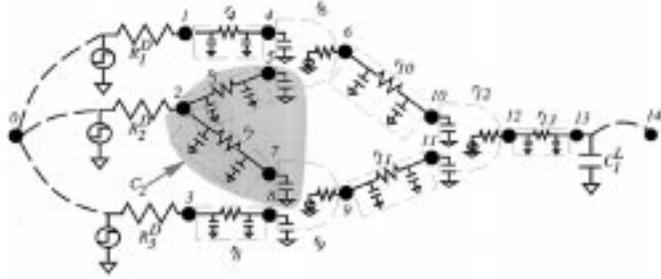


Fig. 4. Before being analyzed, a circuit is transformed to an RC network. Hence, the delay D_i lumped in r_i can be computed by $r_i C_i$. For example, the delay of node 2 is $R_2^P C_2$, where C_2 represents the capacitance for all the capacitors in the shaded area.

$input()$ and $output()$, where $input(i) = \{j | (j, i) \in E, 0 \leq j < i \leq n + s + 1\}$, and $output(i) = \{j | (i, j) \in E, 0 \leq i < j \leq n + s + 1\}$. Furthermore, i belongs to $input(j)$ if and only if j belongs to $output(i)$. For example, in Fig. 2, $input(6) = \{4, 5\}$, $output(0) = \{1, 2, 3\}$.

B. Circuit and Delay Models

In order to explore a circuit, we shall model the circuit elements by analyzable electric components, like resistors and capacitors. Fig. 3 illustrates the gate and wire models used in this paper. For a gate i of 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 of unit size, respectively. In addition, the r_i of an input driver i , $1 \leq i \leq s$, is equal to the input resistor R_i^P . However, same as [5], the intrinsic gate delay is ignored in this model for simplicity. To conquer this problem, we could attach the self-loading capacitance at the output node of each gate. The self-loading capacitance can be approximated by $l c_i$ for a gate i with l inputs. Note that the derivations of Theorems 4–7 remain the same if the intrinsic gate delay is considered, and the corresponding properties still hold. We choose the π model [19] to approximate wire behavior. For a wire j of size x_j , the resistance r_j is \hat{r}_j/x_j , and the capacitance c_j is $\hat{c}_j x_j + f_j + 2C_{c_j}$, where \hat{r}_j and \hat{c}_j are the respective resistance and capacitance of wire j of unit size, f_j is the fringing capacitance of wire j , and C_{c_j} is the coupling capacitance of wire j . Section III-B will detail the coupling capacitance C_{c_j} . The term $2C_{c_j}$ represents the coupling capacitance of wire j in the worst case. By incorporating the coupling capacitance into the wire capacitance, this wire model considers the impacts of crosstalk on delay and power.

With the gate and wire models, a combinational circuit can be transformed to a network with resistors and capacitors. Fig. 4 illustrates the resultant circuit modeling for the circuit shown in Fig. 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 Fig. 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 \bar{R}_i means the weighted upstream resistance of node i in Section IV.

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

C. 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:

\mathcal{M} : Minimize

$$A$$

subject to

$$\begin{aligned} X &\leq X^B, \\ D &\leq D^B, \\ P &\leq P^B. \end{aligned}$$

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

III. 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 I. We compute the physical coupling capacitance between two wires i and j using the model mentioned in Section I

$$\begin{aligned} crosstalk(i, j) &= switching_dissimilarity(i, j) \\ &\quad \times coupling_capacitance(i, j). \end{aligned}$$

We will deal in turn with the two crucial factors which affect the crosstalk—switching behavior and physical coupling capacitance.

A. 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. In real applications, the information of switching behavior can be retrieved during the logic simulation stage or based on the patterns in previous designs. 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:

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

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 , $-1 \leq \text{similarity}(i, j) \leq 1$. The closer to -1 for similarity , the less similar their behavior; the closer to 1 for similarity , the more similar their behavior.

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 $\text{dissimilarity}(i, j)$ equal to $1 - \text{similarity}(i, j)$. An ordering is a sequence composed of all nodes, $\langle w_1, w_2, \dots, w_n \rangle$. Accordingly, the total effective loading between neighboring wires is $\sum_{i=1}^{n-1} \text{dissimilarity}(w_i, w_{i+1})$. Hence, the *Switching Dissimilarity problem* \mathcal{SD} is defined in the following:

SD: Given n wires and their switching behavior.
 Find an ordering for the wires,
 such that the total effective loading between
 neighboring wires is minimized.

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

Theorem 1: The Switching Dissimilarity problem \mathcal{SD} is NP-hard.

The \mathcal{SD} 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 NP-hard. We construct a complete graph $G' = (V, E')$ by adding all nonedges of G , thus $E' = \{(i, j) | i, j \in V\}$. In addition, the weight of each edge (i, j) is assigned as follows:

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

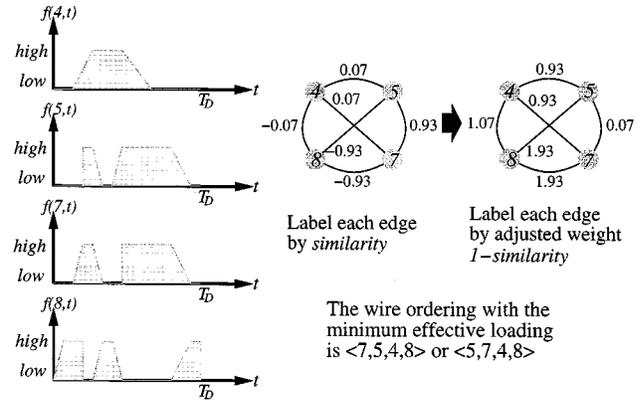


Fig. 5. The waveforms of wires and the similarity between each pair of the wires.

Algorithm: WOSD

(Wire Ordering for the Switching Dissimilarity Problem)

Input: the complete graph K_n for n wires

Output: A wire ordering O

A1. Select a node r to be the root, $1 \leq r \leq n$.

A2. Grow a minimum spanning tree T for K_n from root r .

A3. $O \leftarrow$ the list of nodes visited in a preorder tree walk of T .

Fig. 6. The heuristic of wire ordering for the switching dissimilarity problem.

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{SD} problem is NP-hard.

Since the \mathcal{SD} 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 2: If $P \neq NP$ and $\rho \geq 1$, there is no polynomial-time approximation algorithm with ratio bound ρ for the \mathcal{SD} 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{SD} problem is NP-hard and there exists no efficient approximation algorithm. We propose an efficient minimum spanning tree based heuristic for the \mathcal{SD} problem as shown in Fig. 6. The running time of constructing a minimum spanning tree for a complete graph K_n is $O(n^2)$. A preorder tree walk recursively visits each node in a tree. The walk lists a node when the node is first encountered and before any of its children is visited. The time complexity of a preorder tree walk is $O(n)$. Therefore, the running time of the **WOSD** algorithm is $O(n^2)$. Fig. 7 illustrates the operation of the **WOSD** algorithm on the example shown in Fig. 5.

Solving the switching dissimilarity 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 Fig. 5, if these four wires are routed in the same channel, the geometry ordering

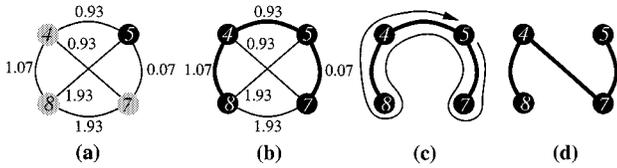


Fig. 7. The execution of the **WOSD** algorithm on the graph from Fig. 5. (a) The complete graph from Fig. 5, where wire 5 is the root. (b) The minimum spanning tree T is identified by the bold lines. (c) The full walk of T is in the order $\{5, 7, 4, 8, 4, 8, 4\}$, yielding the preorder walk $\{5, 7, 4, 8\}$. (d) A wire ordering generated by the **WOSD** algorithm.

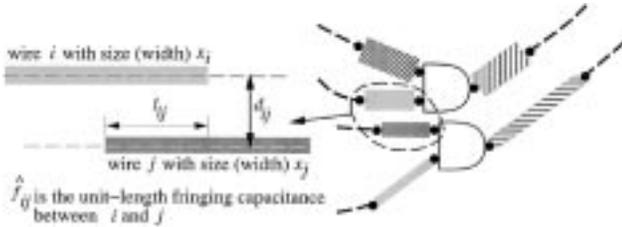


Fig. 8. The physical coupling capacitance between two wires.

is equivalent to track assignment. If we choose $\langle 5, 7, 4, 8 \rangle$ as the resulting track assignment, $N(5) = \{7\}$, $N(7) = \{5, 4\}$, $N(4) = \{7, 8\}$ and $N(8) = \{4\}$; $I(5) = \{7\}$, $I(7) = \emptyset$, $I(4) = \{7, 8\}$ and $I(8) = \emptyset$.

B. Physical Coupling Capacitance

A multiterminal net is decomposed into wire segments. Each line between two junction is treated as a wire. Fig. 8 depicts a case where two wires i and j , belonging to different nets, have coupling capacitance.

According to Fig. 8, the physical coupling capacitance c_{ij} between two neighboring wires i and j can be calculated as follows:

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

where

- x_i and x_j sizes of wires i and j ($x_i, x_j > 0$);
- \hat{f}_{ij} unit-length fringing capacitance between wires i and j ;
- l_{ij} overlap length of wires i and j ;
- d_{ij} distance from the center line of wire i to that of wire j .

In (1), the first term, $(\hat{f}_{ij} l_{ij}/d_{ij})$, is a constant which can be computed by technology files, and the second term, $(1/(1 - (x_i + x_j)/2d_{ij}))$, is what we are concerned. Let $x = ((x_i + x_j)/2d_{ij})$, the second term of (1) becomes $(1/(1 - x))$. 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 $(1/(1 - x))$, we have the following properties.

Theorem 3: Let $f(x) = (1/(1 - x))$, $|x| < 1$.

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

The proof is simple which can be done by the expansion of Taylor series. Theorem 3 reveals $(1/(1 - x))$ can be approximated by $\sum_{n=0}^{k-1} x^n$, the first k terms in the summation. The error ratio is small;¹ for example, for the case $x = 0.25$

- $k = 2$, error ratio $< 6.3\%$;
- $k = 3$, error ratio $< 1.6\%$;
- $k = 4$, error ratio $< 0.4\%$;
- $k = 5$, error ratio $< 0.1\%$.

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, (1) can be approximated as follows:

$$c_{ij} \approx \frac{\hat{f}_{ij} l_{ij}}{d_{ij}} \left(1 + \frac{x_i + x_j}{2d_{ij}}\right) = \tilde{c}_{ij} \left(1 + \frac{x_i + x_j}{2d_{ij}}\right) \quad (2)$$

where $\tilde{c}_{ij} = (\hat{f}_{ij} l_{ij}/d_{ij})$ is a constant. Note that (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 IV.

Recall that, in Section II-B, the capacitance c_i of wire i is $\hat{c}_i x_i + f_i + 2C_{c_i}$. The coupling capacitance C_{c_i} of wire i can be computed by (2) as follows:

$$C_{c_i} = \sum_{j \in N(i)} c_{ij} = \sum_{j \in N(i)} \tilde{c}_{ij} \left(1 + \frac{x_i + x_j}{2d_{ij}}\right).$$

Hence, c_i can be calculated in the following:

$$\begin{aligned} c_i &= \hat{c}_i x_i + f_i + 2 \sum_{j \in N(i)} c_{ij} \\ &= \hat{c}_i x_i + f_i + 2 \sum_{j \in N(i)} \tilde{c}_{ij} \left(1 + \frac{x_i + x_j}{2d_{ij}}\right) \\ &= \left(\hat{c}_i + 2 \sum_{j \in N(i)} \tilde{c}_{ij}\right) x_i + f_i + 2 \sum_{j \in N(i)} \tilde{c}_{ij} \left(1 + \frac{x_j}{2d_{ij}}\right). \end{aligned}$$

IV. 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.

¹In our experiments, the average of x is about 0.12 (ranging from 0.09–0.15), which gives the error ratios of less than 1.5% and 0.2% for $k = 2$ and 3, respectively; thus, the empirical errors are very small.

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

Minimize

$$\mathbf{c}\mathbf{x}$$

subject to

$$\begin{aligned} \mathbf{A}\mathbf{x} &\leq \mathbf{b}, \\ \mathbf{x} &\in \mathbf{X}. \end{aligned}$$

The decision variables \mathbf{x} lie in a given constraint set \mathbf{X} . The Lagrangian relaxation method relaxes the set of constraints $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ to the objective function by introducing *Lagrange multipliers*, $\boldsymbol{\lambda}$, resulting in the *Lagrangian subproblem*

Minimize

$$\mathbf{c}\mathbf{x} + \boldsymbol{\lambda}(\mathbf{A}\mathbf{x} - \mathbf{b})$$

subject to

$$\mathbf{x} \in \mathbf{X}.$$

Since the constraints are relaxed, the Lagrangian subproblem is easier to be solved. By the *Lagrangian Bounding Principle* [1], the *Lagrangian function* $L(\boldsymbol{\lambda}) = \min\{\mathbf{c}\mathbf{x} + \boldsymbol{\lambda}(\mathbf{A}\mathbf{x} - \mathbf{b}) : \mathbf{x} \in \mathbf{X}\}$ 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 $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ or in $\mathbf{A}\mathbf{x} = \mathbf{b}$ form, and the objective and constraints are in a *posynomial* form [14].

Section IV-A formulates the primal problem in the linear programming form. Section IV-B 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 IV-C. In Section IV-D, we present the Lagrangian dual problem and solve it by the subgradient optimization technique.

A. 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 \square_i , the area of component i is $\square_i x_i$; the total area of a circuit is thus $\sum_{i=s+1}^{n+s} \square_i x_i$. The areas occupied by input drivers and output loads are ignored because their areas are fixed. If the respective crosstalk, power, and delay bounds of a circuit are X_B , P_B and A_B , we have

$$\begin{aligned} \sum_{i \in W} \sum_{j \in I(i)} c_{ij} &\leq X_B, \\ V_{DD}^2 f \sum_{i=s+1}^{n+s} \alpha_i c_i &\leq P_B, \\ \sum_{i \in \delta} D_i &\leq A_B, \quad \forall \delta \in \Delta, \end{aligned}$$

where

- V_{DD} supply voltage;
- f working frequency;
- α_i switching activity of component i ;
- δ path in the path set Δ .

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 or a crosstalk bound on the sum of the square of each crosstalk. Further, all corresponding theorems and properties still hold for the extended formulation. Therefore, the optimization problem addressed here can be formulated as follows.

\mathcal{P} : Minimize

$$\sum_{i=s+1}^{n+s} \square_i x_i$$

subject to

$$\sum_{i \in \delta} D_i \leq A_B, \quad \forall \delta \in \Delta, \quad /* Delay */$$

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

$$V_{DD}^2 f \sum_{i=s+1}^{n+s} \alpha_i c_i \leq P_B, \quad /* Power */$$

$$L_i \leq x_i \leq U_i, \quad \forall s+1 \leq i \leq n+s.$$

By the delay constraint in Problem \mathcal{P} , the delay for each source-to-sink path cannot exceed the delay bound A_B . The crosstalk and power constraints mean that the total crosstalk (coupling capacitance) for all nets and total power consumption for all gates and wires cannot exceed the crosstalk and power bounds. From Section III-B, the crosstalk between two adjacent wires i and j is their inter-wire physical coupling capacitance, $\tilde{c}_{ij}(1 + (x_i + x_j)/2d_{ij})$, 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

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

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_{DD} and frequency f are fixed. The power constraint can be simplified by dividing both sides by $V_{DD}^2 f$. Let P_0 be $(P_B/V_{DD}^2 f)$. The power constraint becomes

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

Since the interconnect densities of a circuit can be very high in deep submicrometer technology, the circuit graph could be very dense. Hence, the path set Δ 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

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

Consequently, the problem \mathcal{P} can be modified as follows.

\mathcal{PP} : Minimize

$$\sum_{i=s+1}^{n+s} \square_i x_i$$

subject to

$$\begin{aligned} a_j &\leq A_0, & j \in \text{input}(m), \\ a_j + D_i &\leq a_i, & i = s + 1, \dots, n + s \\ & & \text{and } \forall j \in \text{input}(i), \\ D_i &\leq a_i, & i = 1, \dots, s, \\ \sum_{i=s+1}^{n+s} \alpha_i c_i &\leq P_0, \\ \sum_{i \in W} \sum_{j \in I(i)} \hat{c}_{ij}(x_i + x_j) &\leq X_0, \\ L_i &\leq x_i \leq U_i, \\ & & \forall s + 1 \leq i \leq n + s. \end{aligned}$$

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.

Note that in the formulations for Problems \mathcal{P} and \mathcal{PP} , we did not consider the switching conditions mentioned in Section III-A. To incorporate switching behavior, we can simply multiply c_{ij} by *dissimilarity*(i, j) in the formulations.

B. 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 $\mathbf{x} = (x_{s+1}, \dots, x_{n+s})$ and $\mathbf{a} = (a_1, \dots, a_{n+s})$. The Lagrangian function, therefore, is

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

The corresponding Lagrangian relaxation subproblem is

\mathcal{LRS} 1: Minimize

$$L_{\lambda, \beta, \gamma}(\mathbf{x}, \mathbf{a})$$

subject to

$$L_i \leq x_i \leq U_i, \quad \forall s + 1 \leq i \leq n + s.$$

To solve the Lagrangian relaxation subproblem, we derive the optimality conditions by Kuhn–Tucker conditions [25].

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

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

Proof: By Kuhn–Tucker conditions [25], if the optimal solution $(\mathbf{x}^*, \mathbf{a}^*)$ of the Lagrangian relaxation subproblem $\mathcal{LRS1}$ is the optimal solution of primal problem \mathcal{PP} , then $(\mathbf{x}^*, \mathbf{a}^*)$ must satisfy

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

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

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

By checking Kuhn–Tucker conditions, this theorem thus follows. ■

Theorem 4 reveals 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 (3) in Theorem 4, solving $\mathcal{LRS1}$ is equivalent to solving

$\mathcal{LRS2}$: Minimize

$$L_{\mu, \beta, \gamma}(\mathbf{x})$$

subject to

$$L_i \leq x_i \leq U_i, \quad \forall s + 1 \leq i \leq n + s$$

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

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

Proof: Applying the optimality condition, we get

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

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

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

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

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

C. Lagrangian Relaxation Subproblem

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

Theorem 6: Let $\tilde{\mathbf{x}} = (\tilde{x}_{s+1}, \dots, \tilde{x}_{n+s})$ be a solution, then the optimal resizing of component i is given by

$$x_i^* = \min \left(U_i, \max \left(L_i, \sqrt{\frac{O_1}{O_2 + O_3}} \right) \right)$$

where

$$O_1 = \mu_i \hat{r}_i C_i', \quad O_2 = \square_i + (\beta \alpha_i + R_i) \hat{c}_i, \\ O_3 = \sum_{j \in N(i)} \hat{c}_{ij} (2\beta \alpha_i + \gamma + 2R_i + 2\mu_j r_j).$$

Proof: C_i' is the portion of downstream capacitance C_i which is independent of the size x_i . Hence, C_i' is defined as follows:

$$C_i' = \begin{cases} C_i - \left(\frac{\hat{c}_i}{2} + \sum_{j \in N(i)} \hat{c}_{ij} \right) x_i, & \text{if } i \in W; \\ C_i, & \text{otherwise.} \end{cases}$$

We have

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

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

$$R_i = \sum_{k \in \text{upstream}(i)} \mu_k r_k.$$

Rewriting (4), we have

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

The minimum occurs when

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

Therefore,

$$x_i = \sqrt{\frac{O_1}{O_2 + O_3}},$$

where

$$\begin{aligned}
O_1 &= \mu_i \hat{r}_i C'_i, \quad O_2 = \square_i + (\beta \alpha_i + R_i) \hat{c}_i, \\
O_3 &= \sum_{j \in N(i)} \hat{c}_{ij} (2\beta \alpha_i + \gamma + 2R_i + 2\mu_j r_j).
\end{aligned}$$

Considering the upper bound and lower bound of x_i , we have the optimal resizing for component i

$$x_i^* = \min \left(U_i, \max \left(L_i, \sqrt{\frac{O_1}{O_2 + O_3}} \right) \right),$$

where

$$\begin{aligned}
O_1 &= \mu_i \hat{r}_i C'_i, \quad O_2 = \square_i + (\beta \alpha_i + R_i) \hat{c}_i, \\
O_3 &= \sum_{j \in N(i)} \hat{c}_{ij} (2\beta \alpha_i + \gamma + 2R_i + 2\mu_j r_j). \quad (5)
\end{aligned}$$

Subroutine: LRS (Lagrangian Relaxation Subroutine)
Input: the circuit graph H and Lagrange multipliers μ, β, γ
Output: $\mathbf{x} = (x_{s+1}, \dots, x_{n+s})$ 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
 $x_i \leftarrow \min \left(U_i, \max \left(L_i, \sqrt{\frac{O_1}{O_2 + O_3}} \right) \right)$, where
 $O_1 = \mu_i \hat{r}_i C'_i$,
 $O_2 = \square_i + (\beta \alpha_i + R_i) \hat{c}_i$,
 $O_3 = \sum_{j \in N(i)} \hat{c}_{ij} (2\beta \alpha_i + \gamma + 2R_i + 2\mu_j r_j)$.
S5. Repeat **S2-S4** until no improvement.

Fig. 9. Lagrangian relaxation subroutine.

By (5), the optimal resizing for a gate is mainly determined by its upstream and downstream; that for a wire is dominated by not only the upstream and downstream but also its neighborhood.

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 $\boldsymbol{\lambda}^* = (\lambda_{01}^*, \dots, \lambda_{m-1m}^*)$, β^* , and γ^* such that

- 1) $\sum_{k \in \text{output}(i)} \lambda_{ik}^* = \sum_{j \in \text{input}(i)} \lambda_{ji}^*, \forall 1 \leq i \leq n+s$;
- 2) $\lambda_{jm}^* (a_j - A_0) = 0, \forall j \in \text{input}(m)$;
- 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) $\beta^* (\sum_{s+1}^{n+s} \alpha_i c_i - P_0) = 0$;
- 6) $\gamma^* (\sum_{i \in W} \sum_{j \in I(i)} \hat{c}_{ij}(x_i^* + x_j^*) - X_0) = 0$;
- 7) $a_j^* \leq A_0, \forall j \in \text{input}(m)$;
- 8) $a_j^* + D_i \leq a_i^*, \forall s+1 \leq i \leq n+s$;
- 9) $D_i \leq a_i^*, \forall 1 \leq i \leq s$;
- 10) $\sum_{s+1}^{n+s} \alpha_i c_i \leq P_0$;
- 11) $\sum_{i \in W} \sum_{j \in I(i)} \hat{c}_{ij} \leq X_0$;
- 12) $\lambda_{ji}^* \geq 0, 1 \leq i \leq m$;
- 13) $\beta^* \geq 0$;
- 14) $\gamma^* \geq 0$;
- 15)

$$x_i^* = \min \left(U_i, \max \left(L_i, \sqrt{\frac{O_1}{O_2 + O_3}} \right) \right)$$

where

$$\begin{aligned}
O_1 &= \mu_i \hat{r}_i C'_i, \quad O_2 = \square_i + (\beta \alpha_i + R_i) \hat{c}_i \\
O_3 &= \sum_{j \in N(i)} \hat{c}_{ij} (2\beta \alpha_i + \gamma + 2R_i + 2\mu_j r_j).
\end{aligned}$$

In the above theorem, 1) is the optimality condition; 2)–6) are the complementary slackness conditions; 7)–11) are constraints; 12)–14) restrict nonnegative multipliers; 15) is the optimal sizing.

We propose a greedy algorithm **LRS** in Fig. 9 to optimally solve the Lagrangian relaxation subproblem $\mathcal{LRS2}$ (and equivalently to solve $\mathcal{LRS1}$). 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. ■

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. $\mu = (\mu_1, \dots, \mu_{n+s+1})$, where $\mu_i = \sum_{j \in \text{input}(i)} \lambda_{ji}$.

A3. Call **LRS**;
 Compute a_1, \dots, a_{n+s} .

A4. Adjust multipliers λ_{ji} 's, β , γ :
for $i = 1$ **to** $n + s + 1$ **do**
 forall $j \in \text{input}(i)$ **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 $\lim_{k \rightarrow \infty} \theta_k = 0$
 and $\sum_{j=1}^k \theta_j \rightarrow \infty$.

A5. Project λ onto the nearest point in optimality condition.

A6. $k \leftarrow k + 1$.

A7. Repeat **A2-A6** until
 $(\sum_{i=s+1}^{n+s} \square_i x_i - L_{\lambda, \beta, \gamma}(\mathbf{x})) \leq \text{error bound}$.

Fig. 10. Optimal gate and wire sizing algorithm.

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

D. Lagrangian Dual Problem

It can be shown that there exists a vector of Lagrange multipliers such that the optimal solution of $\mathcal{LRS1}$ is also the optimal solution of the original problem \mathcal{PP} . The problem of finding such a vector is the Lagrangian dual problem described as follows.

\mathcal{LDP} : Maximize

$$D(\lambda, \beta, \gamma)$$

subject to

$$\lambda \text{ in optimal condition,}$$

where

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

We present Algorithm **OGWS** listed in Fig. 10 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 subgradient optimization method. It is well-known that if the step-size sequence $\langle \theta_k \rangle$ satisfies the condition $\lim_{k \rightarrow \infty} \theta_k = 0$ and $\sum_{k=1}^{\infty} \theta_k = \infty$ [e.g., $\theta_k = (1/k)$], the subgradient 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.

TABLE I
RC PARAMETERS AND SIZE BOUNDS OF A
GATE OR A WIRE

	\hat{c}_i (fF/ μm)	\hat{r}_i ($\Omega \cdot \mu\text{m}$)	L_i (μm)	U_i (μm)
gate	8.8	4.73×10^3	0.36	5.0
wire	0.206	0.53	0.36	1.8

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

V. 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 640–9656. The supply voltage was set to 2.5 V, and the working frequency was set to 400 MHz. Listed in Table I, the unit-sized resistance and capacitance of a gate were $4.73 \times 10^3 \Omega \cdot \mu\text{m}$ and 8.8 fF/ μm , and those of a wire were $5.3 \Omega \cdot \mu\text{m}$ and 2.06 fF/ μm , respectively. The respective lower and upper bounds for a gate were 0.36 μm and 5 μm ; those of a wire were 0.36 μm and 1.8 μm . Initially, the sizes of gates and wires were set to 0.36 μm and 1.8 μm , respectively. Table II 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 $((\text{Init} - \text{Fin})/\text{Init}) \times 100\%$.

Our algorithm is effective and efficient. The results show that our algorithm, on the average, improved the respective area, noise, power, and delay by 79.98%, 80.00%, 16.02%, and 1.77% after wire and gate sizing. For the largest circuit, c7552, with 3512 gates and 6144 wires, our algorithm needed only 19.4-min runtime 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 not 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 Fig. 11, 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 and the circuit size is depicted in Fig. 12. Figs. 11 and 12 show that the runtime and the storage requirements of our algorithm approach linear in the total number of gates and wires. As revealed by Fig. 12, 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.

TABLE II
EXPERIMENTAL RESULTS IN NOISE, DELAY, POWER, AND AREA

Ckt Name	Ckt Size			Noise (fF)		Delay (ns)		Power (mW)		Area (kum^2)		ite	time (sec)	mem (KB)
	#G	#W	tot	Init	Fin	Init	Fin	Init	Fin	Init	Fin			
c1355	546	1064	1610	3.69	0.74	274.81	272.89	3.73	3.08	1915.59	383.49	2	34.2	1096
c1908	880	1498	2378	5.19	1.04	280.15	277.20	5.34	4.40	2696.96	539.90	2	70.8	1184
c2670	1193	2076	3269	7.21	1.44	280.59	275.95	7.37	6.06	3737.55	748.19	2	134	1320
c3540	1669	2939	4608	10.20	2.04	283.38	279.92	10.43	8.54	5291.24	1059.21	2	265.5	1472
c432	214	426	640	1.47	0.29	280.15	272.62	1.49	1.26	766.96	153.59	2	4	976
c499	514	928	1442	3.22	0.64	273.24	271.58	3.28	2.71	1670.75	334.48	2	25.2	1072
c5315	2307	4386	6693	15.22	3.05	282.62	279.19	15.43	12.58	7896.31	1580.67	2	596.8	1752
c6288	2416	4800	7216	16.66	3.33	321.82	303.15	16.81	13.82	8641.53	1730.02	2	716.0	1808
c7552	3512	6144	9656	21.33	4.26	282.21	277.49	21.81	17.79	11061.42	2214.19	2	1163.5	2120
c880	383	729	1112	2.53	0.51	273.91	270.94	2.56	2.12	1312.47	262.75	2	14.4	1032
Impr(%)	-			80.00%		1.77%		16.02%		79.98%		-		

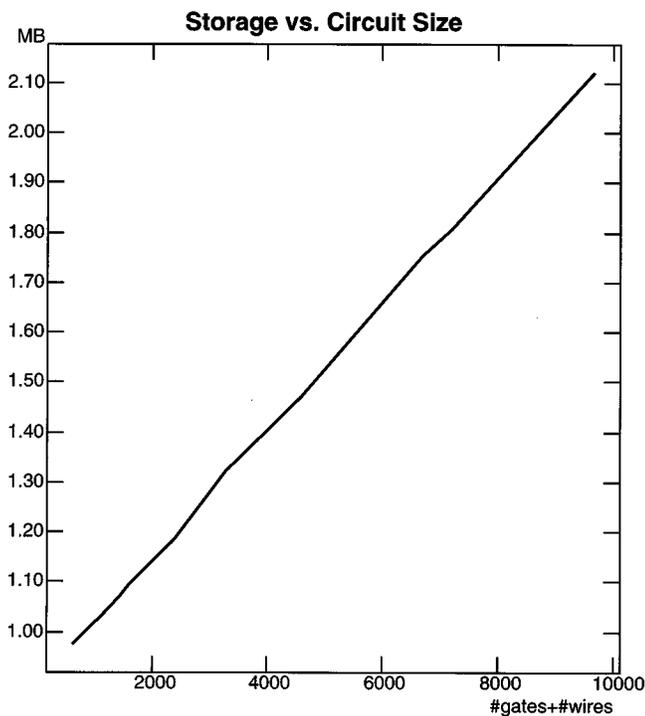


Fig. 11. The storage requirement of our algorithm versus circuit size.

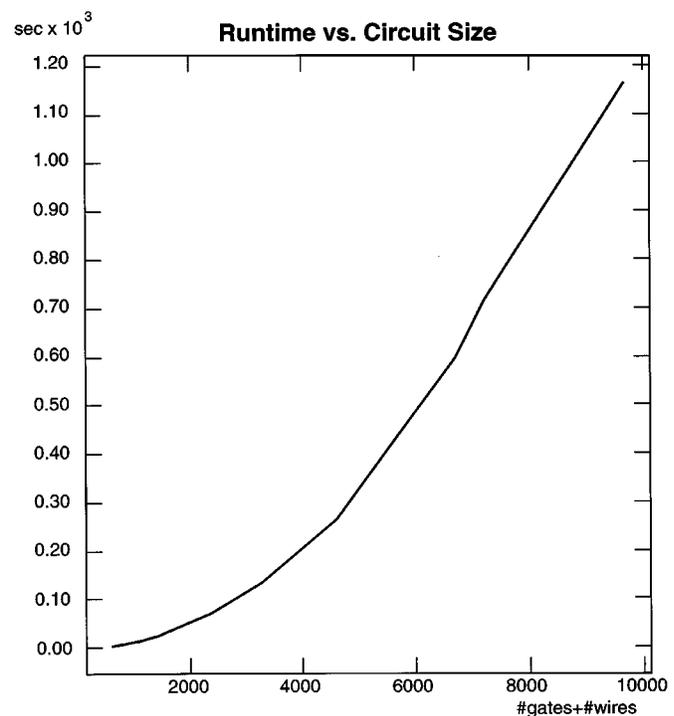


Fig. 12. The runtime requirement of our algorithm versus circuit size.

VI. CONCLUDING REMARKS

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

ACKNOWLEDGMENT

The authors would like to thank Prof. C.-P. Chen of University of Wisconsin at Madison for his suggestions and help. They would also like to thank the anonymous reviewers for their constructive comments.

REFERENCES

- [1] R. K. Ahuja, T. L. Magnati, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [2] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Reading, MA: Addison-Wesley, 1990.
- [3] O. Coudert, "Gate sizing: A general purpose optimization approach," in *Proc. Eur. Design and Test Conf.*, Paris, France, Mar. 1996, pp. 214–218.

- [4] C.-P. Chen, Y.-W. Chang, and D. F. Wong, "Fast performance-driven optimization for buffered clock trees based on lagrangian relaxation," in *Proc. 33rd Design Automation Conf.*, Las Vegas, NV, June 1996, pp. 405–408.
- [5] C.-P. Chen, C. C. N. Chu, and D. F. Wong, "Fast and exact simultaneous gate and wire sizing by lagrangian relaxation," in *Proc. Int Conf. Computer-Aided Design*, Santa Clara, CA, November 1998, pp. 617–624.
- [6] C.-P. Chen and D. F. Wong, "Optimal wire-sizing function with fringing capacitance consideration," in *Proc. 34th Design Automation Conf.*, Anaheim, CA, June 1997, pp. 604–607.
- [7] D.-S. Chen and M. Sarrafzadeh, "An exact algorithm for low power library-specific gate re-sizing," in *33rd Design Automation Conf.*, Las Vegas, NV, June 1996, pp. 783–788.
- [8] L. O. Chua, C. A. Desoer, and E. S. Kuh, *Linear and Nonlinear Circuits*. New York: McGraw Hill, 1987.
- [9] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA: The MIT Press, 1990.
- [10] H. Eisenmann and F. M. Johannes, "Generic global placement and floorplanning," in *Proc. 35th Design Automation Conf.*, San Francisco, CA, June 1998, pp. 269–274.
- [11] W. C. Elmore, "The transient response of damped linear networks with particular regard to wide band amplifiers," *J. Appl. Phys.*, vol. 19, no. 1, 1948.
- [12] T. Gao and C. L. Liu, "Minimum crosstalk channel routing," in *Proc. Int Conf. Computer-Aided Design*, Santa Clara, CA, Nov. 1993, pp. 692–696.
- [13] —, "Minimum crosstalk switchbox routing," in *Proc. Int Conf. Computer-Aided Design*, San Jose, CA, November 1994, pp. 610–615.
- [14] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*, 5th ed. New York: McGraw Hill, 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," in *Proc. Int Symp. Circuits and Systems*, vol. 2, Monterey, CA, May 1998, pp. 248–251.
- [16] M. Marek-Sadowska, Impact of deep sub-micron technologies on physical design, Lecture notes and Private Communication, Aug. 1998.
- [17] Y. Massoud, S. Majors, T. Bustami, and J. White, "Layout techniques for minimizing on-chip interconnect self inductance," in *Proc. 35th Design Automation Conf.*, San Francisco, CA, June 1998, pp. 566–571.
- [18] M. Nemani and F. N. Najm, "High-level area and power estimation for VLSI circuits," in *Proc. Int Conf. Computer-Aided Design*, San Jose, CA, November 1997, pp. 114–119.
- [19] J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [20] M. Sarrafzadeh, D. Knol, and G. Tellez, "Unification of budgeting and placement," in *Proc. 34th Design Automation Conf.*, Anaheim, CA, June 1997, pp. 758–761.
- [21] P. Saxena and C. L. Liu, "Crosstalk minimization using wire perturbations," in *Proc. 36th Design Automation Conf.*, New Orleans, LA, June 1999, pp. 100–103.
- [22] K. L. Shepard, "Design methodologies for noise in digital integrated circuits," in *Proc. 35th Design Automation Conf.*, San Francisco, CA, June 1998, pp. 94–99.
- [23] K. L. Shepard and V. Narayanan, "Conquering noise in deep-submicron digital ICs," *IEEE Design Test Comput.*, pp. 51–62, Jan.–Mar. 1998.
- [24] H.-P. Tseng, L. Scheffer, and C. Sechen, "Timing and crosstalk driven area routing," in *Proc. 35th Design Automation Conf.*, San Francisco, CA, June 1998, pp. 378–381.
- [25] W. L. Winston, *Operations Research: Applications and Algorithms*, 3rd ed: Int Thomson, 1994.
- [26] T. Xue, E. S. Kuh, and D. Wang, "Post global routing crosstalk risk estimation and reduction," in *Proc. Int Conf. Computer-Aided Design*, San Jose, CA, Nov. 1996, pp. 302–309.

- [27] G. Yee, R. Chandra, V. Ganesan, and C. Sechen, "Wire delay in the presence of crosstalk," in *Int Workshop Timing Issues in the Specification and Synthesis of Digital Systems (TAU)*, Dec. 1997, pp. 170–175.



Iris Hui-Ru Jiang received the B.S. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1995. She is currently pursuing the Ph.D. degree in electronics at the same university.

Her research interests focus on interconnect optimization in deep submicrometer technology.

Ms. Jiang is a member of ACM and ACM/SIGDA.



Yao-Wen Chang (S'94–M'99) received the B.S. degree in computer science and information engineering from National Taiwan University, Taiwan, in 1988, and the M.S. and the Ph.D. degrees in computer science from the University of Texas at Austin in 1993 and 1996, respectively.

He was with IBM T. J. Watson Research Center, Yorktown Heights, NY, in the VLSI group during the summer of 1994. Currently, he is an Associate Professor in the Department of Computer and Information Science at National Chiao Tung University,

Hsinchu, Taiwan, where he received an inaugural all-university Excellent Teacher Award (#1 in the department) in 2000. His research interests lie in design automation, architectures, and systems for VLSI and combinatorial optimization.

Dr. Chang received the Best Paper Award of the CAD track at the 1995 IEEE International Conference on Computer Design (ICCD-95) for his work on FPGA routing. He is a member of IEEE Circuits and Systems Society, ACM, and ACM/SIGDA.



Jing-Yang Jou (S'82–M'83) received the B.S. degree in electrical engineering from National Taiwan University, Taiwan, and the M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana-Champaign.

He is a Professor in the Department of Electronics Engineering at National Chiao Tung University, Hsinchu, Taiwan. He has worked in the GTE Laboratories and in the Bell Laboratories. His research interests include behavioral and logic synthesis, VLSI designs and CAD for low power, design

verification, and hardware/software codesign. He has published more than 80 journal and conference papers.

Dr. Jou is a member of Tau Beta Pi, and the recipient of the distinguished paper award of the IEEE International Conference on Computer-Aided Design, 1990. He served as the Technical Program Chair of the Asia-Pacific Conference on Hardware Description Languages (APCHDL'97)