Meeting Delay Constraints in DSM by Minimal Repeater Insertion*

I-Min Liu Adnan Aziz
Department of Electrical and Computer Engineering
The University of Texas at Austin

D. F. Wong
Department of Computer Sciences
The University of Texas at Austin

Abstract

We address the problem of inserting repeaters, selected from a library, at feasible locations in a placed and routed network to meet user-specified delay constraints. We use minimal repeater area by taking advantage of slacks available in the network. Specifically, we transform the problem into an unconstrained optimization problem and solve it by iterative local refinement. We show that the optimal repeater locations and sizes that locally minimize the objective function in the unconstrained problem can be efficiently computed. We have implemented our algorithm and tested it on a set of benchmarks; experimental results are promising.

1. Introduction

The difficulties associated with dealing with deep submicron (DSM) effects in designing integrated circuit have been the challenges to the continuation of Moore's law [16, 17]. Among these effects, rising RC delay on on-chip wiring, increasing noise susceptibility due to coupling, delay prediction considering inductance and noise effects, and power and reliability concerns due to increasing current density are commonly mentioned. How to deal with these DSM issues remains to be an active research area. In this paper, we focus on coping with RC delay on global net wiring by minimal repeater insertion.

It is indicated in [17] that RC delay on on-chip wiring is a critical component in determining chip performance in DSM. One major reason is that, with the scaling of process technology, while wire capacitance per unit length is roughly constant, wire resistance per unit length tends to increase. As a result, although device delay tends to decrease, wire delay, particularly on global nets, tends to increase. This has made wire delay more dominant in chip performance than device delay in recent VLSI design. To alleviate this problem, in VLSI design stages, after module assembly and global net routing, it is often imperative to further reduce the delay on global nets to enhance chip performance. Among various interconnect delay optimization methods, repeater insertion is one of the most effective techniques [8].

There has been considerable previous research in repeater insertion. Van Ginneken [18] proposes a dynamic-programming based algorithm which finds an optimal repeater placement in a distributed RC tree that gives minimum delay. It has been since then generalized to other applications, for example, buffer insertion for low power [12], buffered Steiner tree construction [14], buffered maze routing [19], etc. On the other hand, Kang and Dai [11] address the problem of delay bounded buffered tree construction. Chu and Wong [7] propose a quadratic programming approach to size a wire and insert repeaters for efficient delay and area optimization. The authors in [4, 6] integrate moment and driving point admittance matching into [18] for accurate delay and noise computation. Furthermore, on-chip inductance is taken into account in [10] for designing a RLC wire.

Our major concern is that all these papers only considered the optimization of individual nets or wires. An optimal procedure for placing repeaters in RC trees for minimum delay can be used to obtain a design which is timing optimum by inserting repeaters on each individual net. However, in such an approach the repeater area used is usually wasteful. In addition, excess repeaters make the incremental update of existing placement and routing difficult. In [13], we propose an efficient algorithm to repeater insertion in a network. We showed that area saving is significant, taking advantage of slacks available in a network. However, the repeaters are of one size and there is no consideration about the feasibility of repeater locations.

This work is a continuation of [13]. In this paper, we address the problem of inserting repeaters, selected from a library, at feasible locations in a placed and routed network to meet user-specified delay constraints. We use minimal repeater area by taking advantage of slacks available in the network. Specifically, we transform the problem into an unconstrained optimization problem and solve it by *iterative local refinement*. We show that the optimal repeater locations and sizes that *locally* minimize the objective function can be efficiently computed by *quadratic programming* or *dynamic programming*, depending on the restrictions on repeater locations. We have implemented our algorithm and tested it on a set of benchmarks; experimental results are promising.

*This work is supported by grants from NSF, THECB ATP, and IBM.

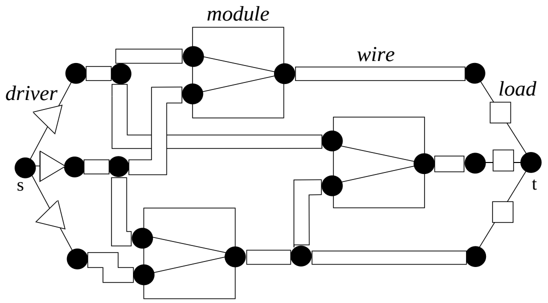

Figure 1. Graph representing a network.

The remainder of this paper is organized as follows. In Section 2, we give notations, definitions and problem formulations. In Section 3, we present algorithms to finding repeater locations and sizes so as to minimize a linear combination of area and delay. In Section 4, we use our algorithms developed in Section 3 to solve the delay constrained minimal repeater insertion problem. We present experimental results in Section 5 and conclude in Section 6.

2. Preliminary

The input to our problem is a placed and routed netlists of modules, with designer specified input drivers and output loads. Our goal is to inserting repeaters, selected from a library, at feasible locations such that total repeater area used is minimal while specified timing constraints are satisfied.

We model a placed and routed network by a directed acyclic graph (DAG), as shown in Figure 1. The nodes in the graph correspond to the primary inputs, primary outputs, tree junctions, and module inputs and outputs in the network. The edges connecting these nodes ($\mathcal{I}$) correspond to the wires connecting the modules ($\mathcal{W}$) and the input-output pairs associated with the modules ($\mathcal{M}$).

We introduce two additional nodes, s and t, in the graph: the source node s is connected to the primary inputs, and the primary outputs are connected to the sink node t. While the edges leaving the source node ($\mathcal{S}$) correspond the input drivers of the network, the edges incident the sink node ($\mathcal{T}$) correspond the output load of the network.

2.1. Repeater Solution

We use Elmore delay [9] and the following RC models. We model wires by their equivalent π-model. Given a wire e, l_e, C_e, and R_e are the length, capacitance, and resistance of e, respectively. Let C and R be the unit length wire capacitance and resistance. We have $C_e = Cl_e$ and $R_e = Rl_e$. We use a switch-level RC model for repeaters. Given a repeater b, C_b, R_b and τ_b are the capacitance, resistance and intrinsic delay of b, respectively. We characterize a module by modeling the critical path between each input-output pair. That is, we model each input-output pair by the capacitance of the input driver, the resistance of the output driver, and the critical path delay between the input-output pair.

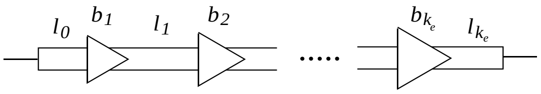

Figure 2. Variables associated with the repeater solution of a wire.

We represent the repeater solution associated with a wire as the illustration in Figure 2. Let k_e be the number of repeaters on e. When $k_e \geq 1$, $b_1, \cdots, b_{k_e}$ are the repeaters from the input to the output spatially. Let $A_{b_1}, \cdots, A_{b_{k_e}}$ be the area of $b_1, \cdots, b_{k_e}$, respectively, and $l_0, l_1, \cdots l_{k_e}$ be the length of wire segments separated by those repeaters. Clearly, the total repeater area A_e is equal to $A_{b_1} + \cdots + A_{b_{k_e}}$ and the wirelength l_e is equal to $l_0 + \cdots + l_{k_e}$.

An element in $\mathcal{E} = \mathcal{W} \cup \mathcal{M} \cup \mathcal{S} \cup \mathcal{T}$ is called a *component*. Given a component u, $\mathcal{D}_u$ is the delay through u and a_u is the arrival time at u, $in(u)$ is the fanins of u and $out(u)$ the fanouts of u. For an input driver, $\mathcal{A}_u$ is the (user-specified) arrival time at the driver. Similarly, for an output load, $\mathcal{Q}_u$ is the (user-specified) required time at the load. Following this, we can express the problem of minimal repeater insertion under input and output timing constraints as below:

$$\begin{array}{lll} & \min \sum_{u \in \mathcal{W}} A_u & \\ \textit{s.t.} & a_u \leq \mathcal{Q}_u & \forall u \in \mathcal{T} \\ & a_{u'} + D_u \leq a_u & \forall u \in \mathcal{I} \text{ and } u' \in in(u) \\ & \mathcal{A}_u + D_u \leq a_u & \forall u \in \mathcal{S} \end{array}$$

We call the above the *primal problem*. Our goal is to solve the problem by finding repeater locations and sizes on all wires in the network. However, the large number of delay constraints make this problem difficult to solve directly.

2.2. Problem Transformation

Following [5, 13], we transform the primal problem into an unconstrained optimization problem (sometimes called a *dual problem*) by Lagrange relaxation [15], Specifically, we relax the constraints by multiplying the amount each constraint is violated by a nonnegative real number (sometimes called a “Lagrangian multiplier”) and adding them to the objective function as penalties. Thus, instead of solving the primal problem, we solve the following dual:

$$\max_{\lambda} L_\lambda$$

where λ is the vector of multipliers and L_λ is the minimum possible value of

$$\sum_{u \in \mathcal{W}} A_u + \sum_{u \in \mathcal{T}} \lambda_{ut}(a_u - \mathcal{Q}_u)$$

$$+\sum_{u\in\mathcal{I}}\sum_{u'\in\, in(u)} \lambda_{u'u}(a_{u'} + D_u - a_u)$$
$$+\sum_{u\in\mathcal{S}} \lambda_{su}(\mathcal{A}_u + D_u - a_u)$$

over all repeater locations and sizes.

Our goal is to find values for the multipliers such that L_λ is maximized. At an optimal solution, the slope with respect to each variable must be zero (This is referred to as Kuhn-Tucker conditions [15]). These conditions can be simplified to

$$\sum_{u'\in\, in(u)} \lambda_{u'u} = \sum_{u''\in\, out(u)} \lambda_{uu''}$$

Let Ω_λ be the set of vectors, λ, which satisfy this constraint. We use the crucial observation in [5, 13] that using these relationships between the multipliers, the objective function can be simplified to

$$\sum_{u\in\mathcal{W}} A_u + \sum_{u\in\mathcal{S}\cup\mathcal{I}} (\sum_{u'\in\, in(u)} \lambda_{u'u} D_u) - \sum_{u\in\mathcal{T}} \lambda_{ut} \mathcal{Q}_u$$

Denoting $\sum_{u'\in\, in(u)} \lambda_{u'u}$ by β_u, and observing that for a fixed vector λ, the last term in the above expression is a constant, L_λ simplifies to

$$\sum_{u\in\mathcal{W}} A_u + \sum_{u\in\mathcal{S}\cup\mathcal{I}} \beta_u D_u + const$$

which is a linear combination of area and delay.

3. Local Refinement

In order to solve the dual problem, we first consider the problem of computing L_λ, i.e., minimizing $F = \sum_{u\in\mathcal{E}}(A_u + \beta_u D_u)$, given a fixed set β. Our procedure for computing L_λ proceeds by iterative local refinement. That is, we iteratively computes the repeater locations and sizes that locally minimize F until we cannot make further improvement.

Focusing on an wire e, the objective can be considered as the sum of two parts: a function of the variable associated with e and terms independent of those parameters. That is, $F = F_e + o.t.$, in which F_e depends on the repeater solution of e but other terms ($o.t.$) do not. When F is being minimized with respect to e, we only need to concentrate on F_e.

3.1. Repeater Insertion — by QP

We first consider repeater insertion using variable repeater sizes for minimizing F_e when there are no restriction (or very limited restrictions) on repeater locations. We show that under this condition, the repeater solution that gives minimum F_e can be computed efficiently via solving a quadratic program.

To simplify to exposition, we use the following abbreviations. We use $\bar{R}_u$ to represent the resistance of u that is not isolated from downstream components, and $\bar{C}_u$ the capacitance of u that is not isolated from upstream components. Take a wire as an example: when $k_e = 0$, we have $\bar{R}_e = Rl_e$ and $\bar{C}_e = Cl_e$; when $k_e \geq 1$, we have $\bar{R}_e = R_{b_{k_e}} + Rl_{k_e}$ and $\bar{C}_e = Cl_0 + C_{b_{k_1}}$.

Let $Des(u)$ be the set of the downstream components not isolated from of u and $Ans(u)$ be the set of upstream components not isolated from u. Let $\hat{R}_u$ be the sum of upstream resistances $\sum_{u'\in\, Ans(u)} \bar{R}_{u'}$ and let $\hat{C}_u$ be the sum of downstream capacitances $\sum_{u'\in\, Des(u)} \bar{C}_{u'}$. Let $\hat{R}'_u$ be the sum of the weighted upstream resistances $\sum_{u'\in\, Ans(u)} \beta_{u'} \bar{R}_{u'}$.

Following the abbreviations, we have

$$\begin{aligned} F_e \;=\; & \sum_{i=1}^{k_e} A_{b_i} \\ & + \hat{R}'_e(Cl_0 + C_{b_1}) + \beta_e \Big[Rl_0(\frac{Cl_0}{2} + C_{b_1}) \\ & + R_{b_1}(Cl_1 + C_{b_2}) + \tau_{b_1} + Rl_1(\frac{Cl_1}{2} + C_{b_2}) \\ & \vdots \\ & + R_{b_{k_e}}(Cl_{k_e} + \hat{C}_e) + \tau_{b_{k_e}} + Rl_{k_e}(\frac{Cl_{k_e}}{2} + \hat{C}_e) \Big] \end{aligned}$$

This expression can be compactly written as

$$F_e = \frac{1}{2} l^T \phi\, l + \rho^T l + const$$

where $\phi = \beta_e RC \cdot \mathcal{I}$, where $\mathcal{I}$ is the identity matrix,

$$\rho = \begin{bmatrix} C\hat{R}'_e & + & \beta_e RC_{b_1} \\ \beta_e CR_{b_1} & + & \beta_e RC_{b_2} \\ & \vdots & \\ \beta_e CR_{b_{k_e}} & + & \beta_e R\hat{C}_e \end{bmatrix} \text{ and } l = \begin{bmatrix} l_0 \\ l_1 \\ \vdots \\ l_{k_e} \end{bmatrix}$$

where $l_0 \geq 0, \cdots, l_{k_e} \geq 0$.

Observe that ϕ is positive definite and so F_e is a convex function in l. It was noted in [7], when the quadratic program is convex, it can be solved extremely fast. Here, the ϕ is even simpler (in fact, it is simply an identity matrix multiplied by a constant, so no matrix inversion will be required). Since the program can be solved efficiently, it can be called with different values of k_e and repeater sizes, so as to pick the repeater solution which gives minimum F_e.

3.2. Repeater Insertion — by DP

In the previous section, we allowed repeaters to be placed anywhere on a wire. However, in many scenarios,

```
Algorithm local-refinement
input: a wire $e$ and $n$ feasible repeater locations
output: repeater locations and sizes
$S_{n+1} = \{(0, \hat{C}_e, \overline{b}, -)\}$;
for $i$ from $n$ down to 1
    for each solution $s = (f, c, b, s^\dagger)$ in $S_{i+1}$
        $S_i = S_i \cup \{(f + \beta_e Rl_i(c + Cl_i/2), c + Cl_i, \overline{b}, s)\}$;
        for each buffer $b_{lib}$ in library $B$
            $S_i = S_i \cup \{(f + A_{b_{lib}} + \beta_e(R_{b_{lib}}(c + Cl_i)$
                $+\tau_{b_{lib}} + Rl_i(c + Cl_i/2)), C_{b_{lib}}, b_{lib}, s)\}$;
    prune inferior solutions in $S_i$;
find $s = (f, c, b, s^\dagger)$ in $S_1$ that gives minimum $f + \hat{R}'_e c$
    and store $s^* = (\hat{R}'_e \cdot c + f, c, \overline{b}, s)$ at $S_0$;
construct an optimal repeater solution for $e$ based on $s^*$;
```

Figure 3. Algorithm *local-refinement.*

after placement and routing, repeaters cannot be inserted arbitrarily. For example, after chip-level module (macro block) assembly, there are regions corresponding to macro blocks. A wire may be run over such regions; however, inserting repeaters in such regions is forbidden. Therefore, macro blocks constitute a routing resource but an obstacle for repeater insertion.

We solve this problem by dynamic programming to find repeater locations and sizes that minimize F_e under location restrictions. We use B to represent the set of library buffers and $\overline{b}$ to represent no buffer. Given a wire $e = (x, y)$, $p_1, \cdots, p_n$ are the feasible repeater locations on e and p_0 and p_{n+1} are the locations of nodes x and y, respectively. We represent a repeater solution by *partial repeater solutions* recursively. A partial repeater solution at p_i is a quadruple $s = (f, c, b, s^\dagger)$, where f is the contribution of the wire segment, from p_i to p_{n+1}, to F_e, c is the total downstream capacitance at p_i, b is the selected buffer from $B \cup \{\overline{b}\}$, and $s^\dagger$ is a partial repeater solution at p_{i+1}.

Theorem 3.1 A partial repeater solution $s = (f, c, b, s^\dagger)$ is inferior to another partial repeater solution $s' = (f', c', b', s^{\dagger'})$ at a feasible repeater location if (1) $f \geq f'$ and $c > c'$ or (2) $f > f'$ and $c \geq c'$.

Let $S_0, \cdots, S_{n+1}$ be the sets of partial repeater solutions, corresponding to $p_0, \cdots, p_{n+1}$, respectively. Given a wire e, we first store a partial repeater solution $(0, \hat{C}_e, \overline{b}, -)$ in S_{n+1} to characterize the initial cost and the downstream capacitance at p_{n+1}. Following this, we bottom-up generate a set of candidate partial repeater solutions S_i for each feasible repeater location p_i. We consider all combinations of the repeater sizes at p_i and the partial repeater solutions stored in S_{i+1}. Specifically, We generate a partial repeater solution for each repeater size and downstream repeater solution combination and store it in S_i. After this, we remove the inferior solutions according to Theorem 3.1 to keep the number of partial repeater solutions tractable. Among all partial repeater solutions in S_1, we select the partial repeater solution s that gives the minimum $\hat{R}'_e \cdot c + f$ value, and then we create the partial repeater solution $s^* = (\hat{R}'_e \cdot c + f, c, \overline{b}, s)$ and store it at S_0. Figure 3 shows our algorithm *local-refinement* which solves repeater locations and sizes that gives minimum F_e under location restrictions.

Theorem 3.2 The partial repeater solution s^* generated by Algorithm *local-refinement* constructs an optimal repeater solution that gives minimum F_e.

4. Algorithm to the Dual Problem

In order to solve the dual problem, we need to compute L_λ, given a fixed set λ. However, we do not know how to solve L_λ exactly. Thus, we approximate L_λ by iterative greedy local refinement. That is, for each wire we greedily compute the repeater solution that locally minimizes F exactly once. Specifically, for each wire, we first compute the downstream capacitance and the weighted upstream resistance, and then use them to compute the optimal repeater solution by Algorithm *local-refinement*. We iterate this process until there is no improvement.

We solve the dual problem as following: we first find a vector of initial multipliers λ^0, which can be any vector in Ω_λ. After this, we inductively deduce multipliers based on previous multiplier values and repeater solutions. That is, we compute λ^{k+1} from λ^k and the repeater solutions for L_{λ^k}. Specifically, we compute λ^{k+1} by (1) for all $u \in \mathcal{T}$, $\lambda^{k+1}_{ut} = \max\{0, \lambda^k_{ut} + \rho_k(a^k_u - \mathcal{Q}_u)\}$, (2) for all $u \in \mathcal{I}$ and $u' \in in(u)$, $\lambda^{k+1}_{u'u} = \max\{0, \lambda^k_{u'u} + \rho_k(a^k_{u'} + D_u - a^k_u)\}$, and (3) for all $u \in \mathcal{S}$, $\lambda^{k+1}_{su} = \max\{0, \lambda^k_{su} + \rho_k(\mathcal{A}_u + D_u - a^k_u)\}$. (This is sometimes called *sub-gradient update* [2]. The step size ρ_k is set to $1/k$.) Following this, we project λ^{k+1} to the nearest vector in Ω_λ. Based on λ^{k+1}, we compute repeater solutions for $L_{\lambda^{k+1}}$. We keep repeating this process until there is no further improvement.

5. Experimental Results

We implemented our algorithm using DP to solve for repeater locations and sizes in C and experimented with it on a 64MB Pentium-II-2.8 running Linux. The experiments are performed on a $0.1\mu m$ technology predicted in NTRS'97 [1]. We have generated a set of placed and routed network to serve as benchmarks and run our experiments on them.

Figure 4 shows the sequence of repeater areas of a network consisting of 681 components, 398 of which are wires, during a delay-constrained repeater area optimization. The areas are in terms of *min-size repeaters*. In this experiment, we observed the repeater area after each updating of multipliers. We see the solutions converge in a stable manners. In fact, most of the optimization is achieved in the first two hundred iterations, taking only 200 seconds.

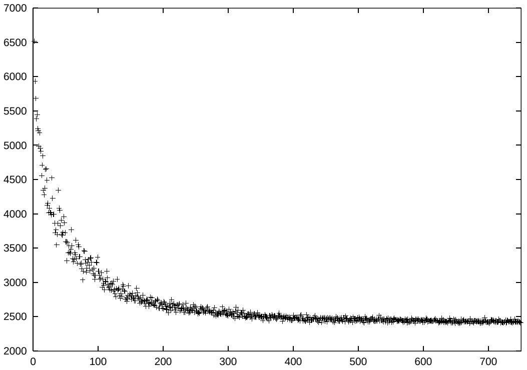

Figure 4. Area vs. # of iterations.

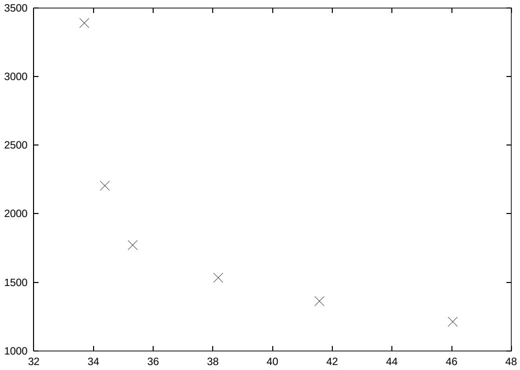

Figure 5. Area-delay tradeoff.

Figure 5 shows the repeater area-delay tradeoff of a network consisting of 343 components, 197 of which are wires. The areas are in terms of *min-size repeaters* and delays are in *nanoseconds*. In this experiment, we varied delay constraints, and observed resulting repeater areas. The results show that a tight constraint yields a result with minimal delay, at cost of a large repeater area. When the delay constraint is more relaxed (by increasing the max-delay bound), we observed that a much less repeater area is required.

6. Conclusion

We address the problem of repeater insertion and sizing in a placed and routed network to meet user-specified delay constraints using minimal repeater area. We transform the problem into an unconstrained optimization problem and solve it by iterative local refinement. We show that the optimal repeater locations and sizes that locally minimize the objective function in the unconstrained problem can be efficiently computed. In the future, we plan to experiment with techniques for meeting delay constraints by minimal restructuring of the physical design, e.g., simultaneous constructing routing tree at the same time as inserting repeaters.

References

[1] *National Technology Roadmap for Semiconductors*. Semiconductor Industry Association, 1997.

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

[3] C. Alpert and A. Devgan. Wire Segmenting for Improved Buffer Insertion. In *Proc. of the Design Automation Conf.*, pages 588–589, 1997.

[4] C. J. Alpert, A. Devgan, and S. T. Quay. Buffer Insertion with Accurate Gate and Interconnect Delay Computation. In *Proc. of the Design Automation Conf.*, pages 479–484, 1999.

[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. Intl. Conf. on Computer-Aided Design*, pages 614–621, 1998.

[6] C. P. Chen and N. Menezes. Noise-aware Repeater Insertion and Wire Sizing for On-Chip Interconnect using Hierarchical Moment-Matching. In *Proc. of the Design Automation Conf.*, pages 502–506, 1999.

[7] C. C. N. Chu and D. F. Wong. A New Approach to Simultaneous Buffer Insertion and Wire Sizing. In *Proc. Intl. Conf. on Computer-Aided Design*, pages 614–621, 1997.

[8] J. Cong, L. He, and C.-K. Koh. Performance Optimization of VLSI Interconnect Layout. In *Integration: the VLSI Journal*, pages 1–94, 1996.

[9] W. C. Elmore. The Transient Response of Damped Linear Network with Particular Regard to Wideband Amplifier. In *Journal of Applied Physics*, pages 55–63, 1948.

[10] Y. I. Ismail and E. G. Friedman. Effects of Inductance on the Propagation Delay and Repeater Insertion in VLSI Circuits. In *Proc. of the Design Automation Conf.*, pages 721–724, 1999.

[11] M. Kang and W. W.-M. Dai. Delay Bounded Buffered Tree Construction for Timing Driven Floorplanning. In *Proc. Intl. Conf. on Computer-Aided Design*, pages 707–712, 1997.

[12] J. Lillis, C.-K. Cheng, and T.-T. Y. Lin. Optimal Wire Sizing and Buffer Insertion for Low Power and a Generalized Delay Model. In *Proc. Intl. Conf. on Computer-Aided Design*, pages 138–143, 1995.

[13] I.-M. Liu, A. Aziz, D. F. Wong, and H. Zhou. An Efficient Buffer Insertion Algorithm for Large Networks Based on Lagrangian Relaxation. In *Proc. Intl. Conf. on Computer Design*, pages 614–621, 1999.

[14] T. Okamoto and J. Cong. Buffered Steiner Tree Construction with Wire Sizing for Interconnect Layout Optimization. In *Proc. Intl. Conf. on Computer-Aided Design*, pages 44–49, 1996.

[15] J. F. Shapiro. *Mathematical Programming: Structures and Algorithms*. John Wiley, 1979.

[16] D. Sylvester and K. Keutzer. Getting to the Bottom of Deep Submicron. In *Proc. Intl. Conf. on Computer-Aided Design*, pages 203–211, 1998.

[17] D. Sylvester and K. Keutzer. Getting to the Bottom of Deep Submicron II: A Global Wiring Paradigm. In *Proc. Intl. Symposium on Physical Design*, pages 193–200, 1999.

[18] L. P. P. P. van Ginneken. Buffer Placement in Distributed RC-tree Networks for Minimal Elmore Delay. In *Proc. Intl. Symposium on Circuits and Systems*, pages 865–868, 1990.

[19] H. Zhou, D. F. Wong, I.-M. Liu, and A. Aziz. Simultaneously Routing and Buffer Insertion with Restrictions on Buffer Locations. In *Proc. of the Design Automation Conf.*, pages 96–99, 1999.