GREEDY WIRE-SIZING IS LINEAR TIME*

Chris C. N. Chu and D. F. Wong
cnchu@cs.utexas.edu and wong@cs.utexas.edu
Department of Computer Sciences, University of Texas at Austin, Austin, TX 78712.

ABSTRACT

In interconnect optimization by wire-sizing, minimizing weighted sink delay has been shown to be the key problem. Wire-sizing with many important objectives such as minimizing total area subject to delay bounds or minimizing maximum delay can all be reduced to solving a sequence of weighted sink delay problems by Lagrangian relaxation [1, 3]. GWSA, first introduced in [10] for discrete wire-sizing and later extended in [2] to continuous wire-sizing, is a greedy wire-sizing algorithm for the weighted sink delay problem. Although GWSA has been experimentally shown to be very efficient, no mathematical analysis on its convergence rate has ever been reported. In this paper, we consider GWSA for continuous wire sizing. We prove that GWSA converges linearly to the optimal solution, which implies that the run time of GWSA is linear with respect to the number of wire segments for any fixed precision of the solution. Moreover, we also prove that this is true for any starting solution. This is a surprising result because previously it was believed that in order to guarantee convergence, GWSA had to start from a solution in which every wire segment is set to the minimum (or maximum) possible width. Our result implies that GWSA can use a good starting solution to achieve faster convergence. We demonstrate this point by showing that the minimization of maximum delay using Lagrangian relaxation can be speed up by 57.7%.

1. INTRODUCTION

With the evolution of VLSI fabrication technology, interconnect delay has become the dominant factor in deep submicron design. In many systems designed today, as much as 50% to 70% of clock cycle are consumed by interconnect delay [7]. As technology continues to scale down, we expect the significance of interconnect delay will further increase in the near future. Wire-sizing has been shown to be an effective technique for interconnect optimization. Many works have been done during the past few years. See [7] for a survey.

In particular, the problem of minimizing weighted sink delay has drawn a lot of attention. Basically, a routing tree with a source, a set of sinks and a set of wire segments is given. Associated with each sink is a non-negative weight representing the criticality of the sink. The problem is to determine the width of each wire segment so that the weighted sum of the delay from the source to the sinks is minimized. Solving this problem is a key to solve problems with many other important objectives such as minimizing total area subject to delay bounds or minimizing maximum delay. It is because [1, 3] have shown that those problems can all be reduced by Lagrangian relaxation to a sequence of weighted sink delay problems. So having efficient algorithms for the weighted sink delay problem is very important for interconnect optimization.

For the problem of minimizing weighted sink delay under Elmore delay model [11], a widely used technique is optimal local re-sizing. The basic idea is to iteratively and greedily re-size the wire segments. In each iteration, the wire segments in the tree are examined one by one. When a wire segment is examined, it is re-sized optimally while keeping the widths of all other segments fixed. This technique was first introduced in [10] and was later extended to many other wire, buffer, gate, driver and/or transistor sizing problems [1, 2, 4, 5, 6, 8, 9].

In [10], discrete wire-sizing (i.e. the segment widths must be chosen from a given set of discrete choices) was considered. The proposed algorithm was called GWSA (Greedy Wire-Sizing Algorithm). GWSA does not give the optimal solution directly as it can converge to non-optimal solutions. Rather, GWSA is used to get lower and upper bounds on the segment widths of the optimal solution. Then dynamic programming technique is used to find the optimal solution among all the possible solutions satisfying the lower and upper bounds. As the lower and upper bounds obtained by GWSA are close to each other in most cases, the dynamic programming step is usually very efficient.

In [2], GWSA was extended to continuous wire-sizing (i.e. the segment widths can be from a continuous range of real numbers). It was proved in [2] that for continuous wire-sizing, GWSA always converges to the optimal solution, provided that all segments are set to their minimum (or maximum) possible widths for the starting solution. However, the convergence rate of GWSA is not known.

In this paper, we analyze the convergence of GWSA for continuous wire-sizing. One of our contributions is we prove that the convergence rate of GWSA is linear. This implies that the run time of GWSA is $O(n \log \frac{1}{\epsilon})$ where n is the number of wire segmspecifiesd the precision of the solution (see Theorem 2). So GWSA runs in time linear to n for a fixed precision.

For all previous algorithms using optimal local re-sizing, the convergence always depends on the fact that the solution of optimal local re-sizing satisfies a special *dominance property* [10]. That is if a wire-sizing solution is dominated by the optimal solution (i.e. the width of every segment in the solution is smaller than or equal to that in the optimal solution), then the solution after an optimal local re-sizing of any segment will still be dominated by the optimal so-

*This work was partially supported by the Texas Advanced Research Program under Grant No. 003658288 and by a grant from the Intel Corporation.

ISPD 98, April 6-8, 1998, Monterey, CA USA
© 2000 ACM ISBN 1-58113-021-x/98/04...$5.00

lution. So if we start from a solution with every segment set to its minimum possible width (this solution is obviously dominated by the optimal solution), then after any number of optimal local re-sizing, the solution will still be dominated by the optimal solution. In other words, for any segment, the optimal width is always an upper bound to the width by optimal local re-sizing. Since segment widths are non-decreasing during optimal local re-sizing and are upper bounded, the solution must converge (to a lower bound of the optimal solution for discrete wire-sizing, and to the optimal solution for continuous wire-sizing). A similar property holds for wire-sizing solutions which dominate the optimal solution.

Therefore, previously in order to guarantee convergence, GWSA always sets all segments to their minimum (or maximum) possible widths for the starting solution. Another contribution of this paper is we prove that for continuous wire-sizing, GWSA always converges to the optimal solution from any starting solution. This is done by proving the convergence of GWSA without using dominance property. So by using a good starting solution for GWSA, faster convergence can be achieved.

This result on starting solution is particularly useful in optimizing other objectives (e.g. minimizing total area subject to delay bounds or minimizing maximum delay) by Lagrangian relaxation. A problem with other objective can be solved optimally by reducing it to a sequence of weighted sink delay problems using the Lagrangian relaxation technique. Previously, before solving each weighted sink delay problem, in order to guarantee convergence, all segments are reset to their minimum (or maximum) possible widths to form the starting solution for GWSA. However, since two consecutive weighted sink delay problems in the sequence are almost the same (except that the sink weights are changed by a little bit), the optimal solution of the first weighted sink delay problem is close to the optimal solution of the second one, and hence a good starting solution to the second one. So it is better not to reset the wire-sizing solution before solving each weighted sink delay problem. We experimentally verify that our new approach of not reseting is much better than the previous approach of reseting each time. We show that our approach can speed up the minimization of maximum delay using Lagrangian relaxation by 57.7%.

The rest of this paper is organized as follows. In Section 2, we will present the weighted sink delay problem and the algorithm GWSA considered in [2]. In Section 3, we will analyze the convergence of GWSA. In Section 4, experimental results to show the linearity of the run time of GWSA and the speedup on optimizing other objectives using Lagrangian relaxation are presented.

2. THE WEIGHTED SINK DELAY PROBLEM AND THE ALGORITHM GWSA

In this section, we will first present the continuous wire-sizing problem with weighted sink delay objective and then the algorithm GWSA considered in [2].

Assume that we are given a routing tree T implementing a signal net which consists of a source (at the root) with driver resistance R_D, a set of n wire segments $\mathcal{W} = \{W_1, W_2, \ldots, W_n\}$, and a set of m sinks $\mathcal{N} = \{N_1, N_2, \ldots, N_m\}$ (at the leaves) with load capacitance c_k^s, $1 \le k \le m$. Associated with each sink N_k is a non-negative weight λ_k representing the criticality of the sink. Assume without loss of generality that $\sum_{k=1}^{m} \lambda_k = 1$. Basically, the problem is to minimize the weighted sink delay for the routing tree by changing the widths of the wire segments. See Figure 1 for an example of a routing tree.

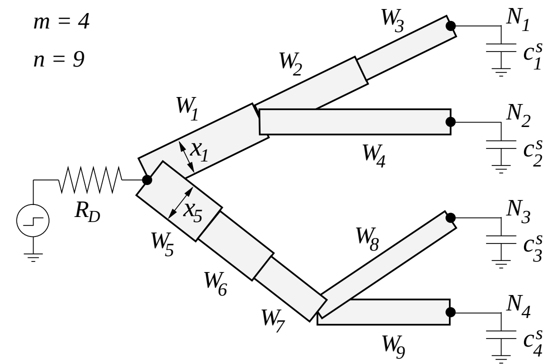

Figure 1. *An example of a routing tree.*

Let $dec(W_i)$ be the set of descendant wire segments or sinks of W_i (excluding W_i). Let $ans(W_i)$ be the set of ancestor wire segments of W_i (excluding W_i). Let $path(N_k)$ be the set of wire segments on the path from the driver to the sink N_k. For example, for the routing tree as shown in Figure 1, $dec(W_1) = \{W_2, W_3, W_4, N_1, N_2\}$, $ans(W_1) = \{\}$, $dec(W_8) = \{N_3\}$, $ans(W_8) = \{W_5, W_6, W_7\}$, and $path(N_3) = \{W_5, W_6, W_7, W_8\}$.

For $1 \le i \le n$, let x_i be the width of wire segment W_i, and L_i and U_i be respectively the lower bound and the upper bound on the width of W_i. Therefore, $L_i \le x_i \le U_i$ for $1 \le i \le n$. Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, which will be referred to as a wire-sizing solution. A wire segment is modeled as a π-type RC circuit as shown in Figure 2. The resistance and capacitance of wire segment W_i are $\widehat{r}_i/x_i$ and $\widehat{c}_i x_i + f_i$ respectively, where $\widehat{r}_i$ is the unit width wire resistance, $\widehat{c}_i$ is the unit width wire area capacitance, and f_i is the wire fringing capacitance of W_i.

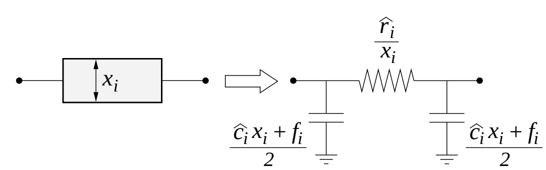

Figure 2. *The model of wire segment W_i by a π-type RC circuit. Note that the resistance and capacitance of the segment are $\widehat{r}_i/x_i$ and $\widehat{c}_i x_i + f_i$ respectively, where $\widehat{r}_i$ is the unit width wire resistance, $\widehat{c}_i$ is the unit width wire area capacitance, and f_i is the wire fringing capacitance of W_i.*

Let $\mu_i = \sum_{N_k \in dec(W_i)} \lambda_k$, i.e. μ_i is the total downstream sink weights of segment W_i.

Let $R_i(\boldsymbol{x}) = \sum_{W_j \in ans(W_i)} \mu_j \widehat{r}_j / x_j$, i.e. $R_i(\boldsymbol{x})$ is a weighted upstream wire resistance of segment W_i.

Let $C_i(\boldsymbol{x}) = \sum_{W_j \in dec(W_i)} \widehat{c}_j x_j$, i.e. $C_i(\boldsymbol{x})$ is the total downstream wire area capacitance of segment W_i.

Let $C_i^{fs} = \sum_{W_j \in dec(W_i)} f_j + \sum_{N_k \in dec(W_i)} c_k^s$, i.e. C_i^{fs} is the total downstream wire fringing capacitance and sink capacitance of segment W_i.

Elmore delay model [11] is used for delay calculation. For a wire-sizing solution $\boldsymbol{x}$, the Elmore delay from the source to the sink N_k is given by

$$\begin{aligned} D_k(\boldsymbol{x}) &= R_D(\sum_{W_j \in \mathcal{W}} \widehat{c}_j x_j + \sum_{W_j \in \mathcal{W}} f_j + \sum_{N_k \in \mathcal{N}} c_k^s) \\ &+ \sum_{W_i \in path(N_k)} \frac{\widehat{r}_i}{x_i} \left(\frac{\widehat{c}_i x_i}{2} + C_i(\boldsymbol{x}) + \frac{f_i}{2} + C_i^{fs} \right) \end{aligned}$$

Then the weighted sink delay problem can be written as:

$$\begin{aligned} &\text{Minimize} \quad D(\boldsymbol{x}) = \sum_{k=1}^{m} \lambda_k D_k(\boldsymbol{x}) \\ &\text{Subject to} \quad L_i \le x_i \le U_i, \ 1 \le i \le n. \end{aligned}$$

Now we present the algorithm GWSA proposed in [2] for solving the weighted sink delay problem. The algorithm GWSA is a greedy algorithm based on iteratively re-sizing the wire segments. In each iteration, the wire segments are examined one by one. When a wire segment W_i is examined, it is re-sized optimally while keeping the widths of all other segments fixed. This operation is called an optimal local re-sizing of W_i. The following lemma gives a formula for optimal local re-sizing.

Lemma 1 *For a wire-sizing solution $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, the optimal local re-sizing of W_i is given by changing the width of W_i to*

$$x_i = \min \left\{ U_i, \max \left\{ L_i, \sqrt{\frac{\mu_i \widehat{r}_i}{\widehat{c}_i} \cdot \frac{C_i(\boldsymbol{x}) + \frac{f_i}{2} + C_i^{fs}}{R_i(\boldsymbol{x}) + R_D}} \right\} \right\}$$

Proof outline: By extending the proof of Lemma 1 in [2] (which did not consider wire fringing capacitance), we can show that

$$\begin{aligned} D(\boldsymbol{x}) &= \widehat{c}_i x_i \left(R_i(\boldsymbol{x}) + R_D\right) + \frac{\mu_i \widehat{r}_i}{x_i} \left(C_i(\boldsymbol{x}) + \frac{f_i}{2} + C_i^{fs} \right) \\ &+ \text{ terms independent of } x_i \end{aligned}$$

Note that $R_i(\boldsymbol{x})$ and $C_i(\boldsymbol{x})$ are also independent of x_i. Hence by Lemma 2 of [2], the result follows. □

Let $children(W_i)$ be the set of all children wire segments of W_i and let p_i be the index of the parent wire segment of W_i. Then the algorithm GWSA is given below. Note that since $C_i(\boldsymbol{x})$ and $R_i(\boldsymbol{x})$ are computed incrementally in step S3 and S4, each iteration of GWSA takes only $O(n)$ time.

For the original GWSA in [2], in S1, x_i is set to L_i for all i. Then dominance property can be applied to show that the algorithm converges. However, the convergence rate is not known. Also, if some other starting wire-sizing solution is used in S1, it it not clear whether the algorithm will still converge. In the next section, we will show that GWSA always converges linearly for any starting solution.

3. CONVERGENCE ANALYSIS OF GWSA

In this section, we will first prove that the algorithm GWSA always converges to the optimal solution for any starting solution (Theorem 1). Then we will prove that the convergence rate for any starting solution is always linear. This implies the run time of GWSA is $O(n \log \frac{1}{\epsilon})$ for any starting solution, where ϵ specifies the precision of the solution (Theorem 2).

For the following two lemmas, we will focus on segment W_k for some fixed k. Note that during the n optimal local re-sizing operations just before the local re-sizing of W_k at a particular iteration (except the first iteration), each wire segment is re-sized exactly once. Intuitively, the following two lemmas show that during these n re-sizing operations, if the changes in all the segment widths are small, then the change in the width x_k during the local re-sizing of W_k at that iteration will be even smaller.

ALGORITHM GWSA:

S1. Let $\boldsymbol{x}$ be some starting wire-sizing solution.

S2. Compute μ_i's and C_i^{fs}'s by a bottom-up traversal of T using the following formula:

$$\mu_i := \begin{cases} \lambda_k, & \text{if } W_i \text{ connects directly to sink } N_k \\ \sum_{W_j \in children(W_i)} \mu_j, & \text{otherwise} \end{cases}$$

$$C_i^{fs} := \begin{cases} c_k^s, & \text{if } W_i \text{ connects directly to sink } N_k \\ \sum_{W_j \in children(W_i)} (f_j + C_j^{fs}), & \text{otherwise} \end{cases}$$

S3. Compute all C_i's by a bottom-up traversal of T using the following formula:

$$C_i(\boldsymbol{x}) := \textstyle\sum_{W_j \in children(W_i)} (\widehat{c}_j x_j + C_j(\boldsymbol{x}))$$

S4. Perform a top-down traversal of T:
For each W_i,

$$R_i(\boldsymbol{x}) := R_{p_i}(\boldsymbol{x}) + \mu_{p_i} \widehat{r}_{p_i} / x_{p_i}$$

$$x_i = \min \left\{ U_i, \max \left\{ L_i, \sqrt{\frac{\mu_i \widehat{r}_i}{\widehat{c}_i} \frac{C_i(\boldsymbol{x}) + \frac{f_i}{2} + C_i^{fs}}{R_i(\boldsymbol{x}) + R_D}} \right\} \right\}$$

S5. Repeat S3–S4 until no improvement.

For some $t \ge 1$, let $\boldsymbol{x} = (x_1, \ldots, x_n)$, $\boldsymbol{x}' = (x_1', \ldots, x_n')$ and $\boldsymbol{x}'' = (x_1'', \ldots, x_n'')$ be respectively the wire-sizing solution just before the local re-sizing of W_k at iteration t, $t+1$ and $t+2$ of GWSA. Let $q_k' = \sqrt{\frac{\mu_k \widehat{r}_k}{\widehat{c}_k} \cdot \frac{C_k(\boldsymbol{x}) + \frac{f_k}{2} + C_k^{fs}}{R_k(\boldsymbol{x}) + R_D}}$ and $q_k'' = \sqrt{\frac{\mu_k \widehat{r}_k}{\widehat{c}_k} \cdot \frac{C_k(\boldsymbol{x}') + \frac{f_k}{2} + C_k^{fs}}{R_k(\boldsymbol{x}') + R_D}}$. So by Lemma 1, $x_k' = \min\{U_k, \max\{L_k, q_k'\}\}$ and $x_k'' = \min\{U_k, \max\{L_k, q_k''\}\}$.

Lemma 2 *For any $\delta > 0$, if $\frac{1}{1+\delta} \le \frac{x_i'}{x_i} \le 1 + \delta$ for all i, then $\frac{1}{1+\delta\alpha} \le \frac{q_k''}{q_k'} \le 1 + \delta\alpha$ for some constant $0 < \alpha < 1$.*

Proof: If $\frac{1}{1+\delta} x_i \le x_i' \le (1+\delta) x_i$ for all i, we have

$$\frac{1}{1+\delta} R_k(\boldsymbol{x}) \ \le \ R_k(\boldsymbol{x}') \ \le \ (1+\delta) R_k(\boldsymbol{x})$$

and

$$\frac{1}{1+\delta} C_k(\boldsymbol{x}) \ \le \ C_k(\boldsymbol{x}') \ \le \ (1+\delta) C_k(\boldsymbol{x}).$$

Let $\alpha = \max\limits_{1 \le k \le n} \left\{ \begin{array}{l} 1/\left(1 + \frac{R_D}{\sum_{W_i \in ans(W_k)} \widehat{r}_i / L_i}\right), \\ 1/\left(1 + \frac{\frac{f_k}{2} + C_k^{fs}}{\sum_{W_i \in ans(W_k)} \widehat{c}_i U_i}\right) \end{array} \right\}$.

Note that α is a constant such that $0 < \alpha < 1$.
Since $0 \le \mu_i \le 1$ and $x_i \ge L_i$ for all i, we have $R_k(\boldsymbol{x}) = \sum_{W_i \in ans(W_k)} \mu_i \widehat{r}_i / x_i \le \sum_{W_i \in ans(W_k)} \widehat{r}_i / L_i$.
So $\alpha \ge 1/\left(1 + R_D / R_k(\boldsymbol{x})\right)$, or equivalently,

$$R_k(\boldsymbol{x}) \quad \le \quad \alpha(R_k(\boldsymbol{x}) + R_D).$$

Hence

$$\begin{aligned} R_k(\boldsymbol{x}') + R_D &\leq (1+\delta)R_k(\boldsymbol{x}) + R_D \\ &= \delta R_k(\boldsymbol{x}) + (R_k(\boldsymbol{x}) + R_D) \\ &\leq \delta\alpha(R_k(\boldsymbol{x}) + R_D) + (R_k(\boldsymbol{x}) + R_D) \\ &= (1+\delta\alpha)(R_k(\boldsymbol{x}) + R_D) \qquad (1) \end{aligned}$$

and

$$\begin{aligned} R_k(\boldsymbol{x}') + R_D &\geq \frac{1}{1+\delta}R_k(\boldsymbol{x}) + R_D \\ &= R_k(\boldsymbol{x}) + R_D - \frac{\delta}{1+\delta}R_k(\boldsymbol{x}) \\ &\geq R_k(\boldsymbol{x}) + R_D - \frac{\delta\alpha}{1+\delta}(R_k(\boldsymbol{x}) + R_D) \\ &= (1 - \frac{\delta\alpha}{1+\delta})(R_k(\boldsymbol{x}) + R_D) \\ &> \frac{1}{1+\delta\alpha}(R_k(\boldsymbol{x}) + R_D) \qquad (2) \\ &\quad \text{as } \delta > 0 \text{ and } 0 < \alpha < 1 \end{aligned}$$

Similarly, since $x_i \leq U_i$ for all i, we have $C_k(\boldsymbol{x}) = \sum_{W_i \in ans(W_k)} \widehat{c}_i x_i \leq \sum_{W_i \in ans(W_k)} \widehat{c}_i U_i$.
So $\alpha \geq 1/\left(1 + (\frac{f_k}{2} + C_k^{fs})/C_k(\boldsymbol{x})\right)$, or equivalently,

$$C_k(\boldsymbol{x}) \leq \alpha(C_k(\boldsymbol{x}) + \frac{f_k}{2} + C_k^{fs}).$$

Hence we can prove similarly that

$$C_k(\boldsymbol{x}') + \frac{f_k}{2} + C_k^{fs} \leq (1+\delta\alpha)(C_k(\boldsymbol{x}) + \frac{f_k}{2} + C_k^{fs}) \qquad (3)$$

and

$$C_k(\boldsymbol{x}') + \frac{f_k}{2} + C_k^{fs} > \frac{1}{1+\delta\alpha}(C_k(\boldsymbol{x}) + \frac{f_k}{2} + C_k^{fs}) \qquad (4)$$

By definitions of q_k' and q_k'', and by (2) and (3), we have

$$\begin{aligned} q_k'' &= \sqrt{\frac{\mu_k \widehat{r}_k}{\widehat{c}_k} \cdot \frac{C_k(\boldsymbol{x}') + \frac{f_k}{2} + C_k^{fs}}{R_k(\boldsymbol{x}') + R_D}} \\ &\leq \sqrt{\frac{\mu_k \widehat{r}_k}{\widehat{c}_k} \cdot \frac{(1+\delta\alpha)(C_k(\boldsymbol{x}) + \frac{f_k}{2} + C_k^{fs})}{\frac{1}{1+\delta\alpha}(R_k(\boldsymbol{x}) + R_D)}} \\ &= (1+\delta\alpha)q_k'. \end{aligned}$$

Similarly, by (1) and (4), we can prove that $q_k'' \geq \dfrac{1}{1+\delta\alpha}q_k'$.
As a result, $\dfrac{1}{1+\delta\alpha} \leq \dfrac{q_k''}{q_k'} \leq 1+\delta\alpha$. □

Lemma 3 *For any* $\delta > 0$, *if* $\dfrac{1}{1+\delta} \leq \dfrac{x_i'}{x_i} \leq 1+\delta$ *for all* i, *then* $\dfrac{1}{1+\delta\alpha} \leq \dfrac{x_k''}{x_k'} \leq 1+\delta\alpha$ *for some constant* $0 < \alpha < 1$.

Proof: By Lemma 2, if $\frac{1}{1+\delta}x_i \leq x_i' \leq (1+\delta)x_i$ for all i, then $\frac{1}{1+\delta\alpha}q_k' \leq q_k'' \leq (1+\delta\alpha)q_k'$ where α is the constant as defined in Lemma 2. By Lemma 1, $x_k' = \min\{U_k, \max\{L_k, q_k'\}\}$ and $x_k'' = \min\{U_k, \max\{L_k, q_k''\}\}$.
In order to prove $\frac{1}{1+\delta\alpha}x_k' \leq x_k''$, we consider three cases:

Case 1) $q_k' < L_k$.
Then $x_k' = L_k$. So $\dfrac{1}{1+\delta\alpha}x_k' = \dfrac{1}{1+\delta\alpha}L_k < L_k \leq x_k''$.

Case 2) $q_k'' > U_k$.
Then $x_k'' = U_k$. So $\dfrac{1}{1+\delta\alpha}x_k' \leq \dfrac{1}{1+\delta\alpha}U_k < U_k = x_k''$.

Case 3) $q_k' \geq L_k$ and $q_k'' \leq U_k$.
Then $q_k' \geq L_k \Rightarrow x_k' \leq q_k'$ and $q_k'' \leq U_k \Rightarrow q_k'' \leq x_k''$. So $\dfrac{1}{1+\delta\alpha}x_k' \leq \dfrac{1}{1+\delta\alpha}q_k' \leq q_k'' \leq x_k''$.

Similarly, by considering the cases $q_k' > U_k$, $q_k'' < L_k$ and ($q_k' \leq U_k$ and $q_k'' \geq L_k$), we can prove $x_k'' \leq (1+\delta\alpha)x_k'$.
As a result, $\dfrac{1}{1+\delta\alpha} \leq \dfrac{x_k''}{x_k'} \leq 1+\delta\alpha$. □

The following two lemmas give bounds on the changes of segment widths after each iteration of GWSA. Let $\boldsymbol{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)})$ be the starting wire-sizing solution, and for $t \geq 1$, let $\boldsymbol{x}^{(t)} = (x_1^{(t)}, x_2^{(t)}, \ldots, x_n^{(t)})$ be the wire-sizing solution just after iteration t of GWSA.

Lemma 4 *For any* $t \geq 0$ *and* $\delta > 0$, *if* $\dfrac{1}{1+\delta} \leq \dfrac{x_i^{(t+1)}}{x_i^{(t)}} \leq 1+\delta$ *for all* i, *then* $\dfrac{1}{1+\delta\alpha} \leq \dfrac{x_i^{(t+2)}}{x_i^{(t+1)}} \leq 1+\delta\alpha$ *for all* i *and for some constant* $0 < \alpha < 1$.

Proof: Assume without loss of generality that the wire segments are indexed in such a way that a top-down traversal of T is in the order of $W_1, W_2, \ldots, W_n$. The lemma can be proved by induction on i.

Base case: Consider the wire segment W_1.
At iteration $t+1$, before local re-sizing of W_1, the wire-sizing solution is $(x_1^{(t)}, x_2^{(t)}, \ldots, x_n^{(t)})$.
At iteration $t+2$, before local re-sizing of W_1, the wire-sizing solution is $(x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_n^{(t+1)})$.
Since $\frac{1}{1+\delta} \leq \frac{x_i^{(t+1)}}{x_i^{(t)}} \leq 1+\delta$ for all i, by Lemma 3, we have $\frac{1}{1+\delta\alpha} \leq \frac{x_1^{(t+2)}}{x_1^{(t+1)}} \leq 1+\delta\alpha$.

Induction step: Assume that the induction hypothesis is true for $i = 1, \ldots, k$.
At iteration $t+1$, before local re-sizing of W_k, the wire-sizing solution is $(x_1^{(t+1)}, \ldots, x_k^{(t+1)}, x_{k+1}^{(t)}, \ldots, x_n^{(t)})$.
At iteration $t+2$, before local re-sizing of W_k, the wire-sizing solution is $(x_1^{(t+2)}, \ldots, x_k^{(t+2)}, x_{k+1}^{(t+1)}, \ldots, x_n^{(t+1)})$.
By induction hypothesis, $\frac{1}{1+\delta\alpha} \leq \frac{x_i^{(t+2)}}{x_i^{(t+1)}} \leq 1+\delta\alpha$, and hence $\frac{1}{1+\delta} \leq \frac{x_i^{(t+2)}}{x_i^{(t+1)}} \leq 1+\delta$ (as $\alpha < 1$) for $i = 1, \ldots, k$.
Also, it is given that $\frac{1}{1+\delta} \leq \frac{x_i^{(t+1)}}{x_i^{(t)}} \leq 1+\delta$ for $i = k+1, \ldots, n$. So by Lemma 3, $\frac{1}{1+\delta\alpha} \leq \frac{x_k^{(t+2)}}{x_k^{(t+1)}} \leq 1+\delta\alpha$.

Hence the lemma is proved. □

Let $\Delta = \max\limits_{1 \leq i \leq n} \left\{ \dfrac{U_i - L_i}{L_i} \right\}$.

Lemma 5 *For any* $t \geq 0$, $\dfrac{1}{1+\Delta\alpha^t} \leq \dfrac{x_i^{(t+1)}}{x_i^{(t)}} \leq 1+\Delta\alpha^t$ *for all* i *and for some constant* $0 < \alpha < 1$.

Proof: This can be proved by induction on t.

Base case: Consider $t = 0$.

Note that for any wire-sizing solution $\boldsymbol{x} = (x_1, \ldots, x_n)$, $L_i \le x_i \le U_i$ for all i. So $\frac{x_i^{(1)}}{x_i^{(0)}} \le \frac{U_i}{L_i} \le 1 + \frac{U_i - L_i}{L_i} \le 1 + \Delta$.

Similarly, we can prove that for all i, $\frac{x_i^{(1)}}{x_i^{(0)}} \ge \frac{1}{1+\Delta}$.

Induction step: Assume that the induction hypothesis is true for t. Therefore, $\frac{1}{1+\Delta\alpha^t} \le \frac{x_i^{(t+1)}}{x_i^{(t)}} \le 1 + \Delta\alpha^t$ for all i. So by Lemma 4, $\frac{1}{1+\Delta\alpha^{t+1}} \le \frac{x_i^{(t+2)}}{x_i^{(t+1)}} \le 1 + \Delta\alpha^{t+1}$ for all i.

Hence the lemma is proved. □

Theorem 1 *GWSA always converges to the optimal wire-sizing solution for any starting solution.*

Proof: For any constant $0 < \alpha < 1$, $1 + \Delta\alpha^t \to 1$ as $t \to \infty$. So by Lemma 5, it is obvious that the algorithm GWSA always converges for any starting wire-sizing solution. Theorem 1 of [2] proved that if GWSA converges, then the wire-sizing solution is optimal. So the theorem follows. □

Let $\boldsymbol{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$ be the optimal wire-sizing solution. The following lemma proves that the convergence rate of GWSA is linear with convergence ratio α.

Lemma 6 *For any $t \ge 0$, $\left|\frac{x_i^* - x_i^{(t)}}{x_i^*}\right| \le \frac{(1+\Delta)\Delta\alpha^t}{1-\alpha}$ for all i.*

Proof: For any $t \ge 0$ and for any i,

Case 1) $\frac{(1+\Delta)\alpha^t}{1-\alpha} \ge 1$.

Then $\frac{x_i^{(t)}}{x_i^*} \le \frac{U_i}{L_i} \le 1 + \Delta \le 1 + \Delta\frac{(1+\Delta)\alpha^t}{1-\alpha}$.

Similarly, we can prove $\frac{x_i^{(t)}}{x_i^*} \ge \frac{1}{1+\frac{(1+\Delta)\Delta\alpha^t}{1-\alpha}}$.

Case 2) $\frac{(1+\Delta)\alpha^t}{1-\alpha} < 1$.

Then $\frac{x_i^{(t)}}{x_i^*} = \prod_{k=t}^{\infty} \frac{x_i^{(k)}}{x_i^{(k+1)}}$. So by Lemma 5, $\frac{1}{P} \le \frac{x_i^{(t)}}{x_i^*} \le P$ where $P = \prod_{k=t}^{\infty}(1 + \Delta\alpha^k)$.

$$\begin{aligned}
\ln P &= \sum_{k=t}^{\infty} \ln(1 + \Delta\alpha^k) \\
&= \sum_{k=t}^{\infty} \left(\Delta\alpha^k - \frac{1}{2}\Delta^2\alpha^{2k} + \frac{1}{3}\Delta^3\alpha^{3k} + \cdots\right) \quad (5) \\
&\le \sum_{k=t}^{\infty} \left(\sum_{j=1}^{\infty} \frac{1}{j}\Delta^j\alpha^{jk}\right) \\
&= \sum_{j=1}^{\infty} \frac{\Delta^j}{j} \sum_{k=t}^{\infty} (\alpha^j)^k \\
&= \sum_{j=1}^{\infty} \frac{\Delta^j}{j} \frac{\alpha^{jt}}{1-\alpha^j} \\
&\le \sum_{j=1}^{\infty} \frac{\Delta^j}{j} \frac{\alpha^{jt}}{(1-\alpha)^j} \quad (6) \\
&= \ln \frac{1}{1 - \frac{\Delta\alpha^t}{1-\alpha}} \quad (7)
\end{aligned}$$

where (5) is because $\ln(1+x) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 + \cdots$, (6) is because $0 < \alpha < 1$, which implies $0 < (1-\alpha)^j < 1 - \alpha < 1 - \alpha^j$ for $j \ge 1$, and (7) is because $0 < \frac{\Delta\alpha^t}{1-\alpha} < \frac{(1+\Delta)\alpha^t}{1-\alpha} < 1$ and $\ln\frac{1}{1-x} = x + \frac{1}{2}x^2 + \frac{1}{3}x^3 + \cdots$ if $0 < x < 1$. So

$$\begin{aligned}
P &\le \frac{1}{1 - \frac{\Delta\alpha^t}{1-\alpha}} \\
&= 1 + \frac{\Delta\alpha^t}{1-\alpha} \Big/ \left(1 - \frac{\Delta\alpha^t}{1-\alpha}\right) \\
&\le 1 + \frac{\Delta\alpha^t}{1-\alpha} \Big/ \left(1 - \frac{\Delta}{1+\Delta}\right) \\
&= 1 + \frac{(1+\Delta)\Delta\alpha^t}{1-\alpha}.
\end{aligned}$$

Hence $\frac{1}{1+\frac{(1+\Delta)\Delta\alpha^t}{1-\alpha}} \le \frac{x_i^{(t)}}{x_i^*} \le 1 + \frac{(1+\Delta)\Delta\alpha^t}{1-\alpha}$.

Therefore for both cases,

$$\frac{1}{1 + \frac{(1+\Delta)\Delta\alpha^t}{1-\alpha}} \le \frac{x_i^{(t)}}{x_i^*} \le 1 + \frac{(1+\Delta)\Delta\alpha^t}{1-\alpha}.$$

It is easy to see that $1 - \frac{(1+\Delta)\Delta\alpha^t}{1-\alpha} \le \frac{1}{1+\frac{(1+\Delta)\Delta\alpha^t}{1-\alpha}}$. So for any $t \ge 0$ and for all i, $\left|\frac{x_i^* - x_i^{(t)}}{x_i^*}\right| \le \frac{(1+\Delta)\Delta\alpha^t}{1-\alpha}$. □

Since the convergence rate of GWSA is linear and the run time of each GWSA iteration is $O(n)$, we have the following theorem.

Theorem 2 *The total run time of GWSA for any starting solution is $O(n \log \frac{1}{\epsilon})$, where ϵ specifies the precision of the final wire-sizing solution (i.e. for the optimal solution $\boldsymbol{x}^*$, the final solution $\boldsymbol{x}$ satisfies $|(x_i^* - x_i)/x_i^*| \le \epsilon$ for all i).*

Proof: By Lemma 6, for any $t \ge 0$ and for all i,

$$\left|\frac{x_i^* - x_i^{(t)}}{x_i^*}\right| \le \frac{(1+\Delta)\Delta\alpha^t}{1-\alpha}.$$

In order to guarantee that $|(x_i^* - x_i^{(t)})/x_i^*| \le \epsilon$ for all i, the number of iterations t must satisfy

$$\frac{(1+\Delta)\Delta\alpha^t}{1-\alpha} \le \epsilon,$$

or equivalently,

$$t \ge \log_{\frac{1}{\alpha}} \frac{(1+\Delta)\Delta}{(1-\alpha)\epsilon}.$$

In other words, at most $\left\lceil \log_{\frac{1}{\alpha}} \frac{(1+\Delta)\Delta}{(1-\alpha)\epsilon} \right\rceil$ iterations are enough. Since each iteration of GWSA takes $O(n)$ time, the total run time is $O(n \log \frac{1}{\epsilon})$. □

Therefore, to obtain a solution with any fixed precision, only a constant number of GWSA iterations are needed. This implies that the run time of GWSA is $O(n)$. In practice, even for very accurate solutions, GWSA usually takes only a few iterations. So, as we will demonstrate in the next section, GWSA is very efficient in practice.

4. EXPERIMENTAL RESULTS

In this section, we will demonstrate the linearity of the run time of GWSA in practice and the use of better starting solutions to speed up the optimization of other objectives using Lagrangian relaxation. We run the algorithm GWSA on an IBM PC with a 200 MHz Pentium Pro processor.

Figure 3 shows the linearity of the run time of GWSA. We are using the clock trees r1–r5 in [12]. The number of segments in these trees range from 533 to 6201. In order to have more data points, we construct 10 trees from each tree by dividing each tree edge into k segments where $k = 1, \ldots, 10$. So we have 50 trees with the number of segments ranging from 533 to 62010. For each tree, we run GWSA with ϵ equals 10^{-5}. The run time is plotted against the number of segment in Figure 3. It can be seen that the run time of GWSA is linear in practice.

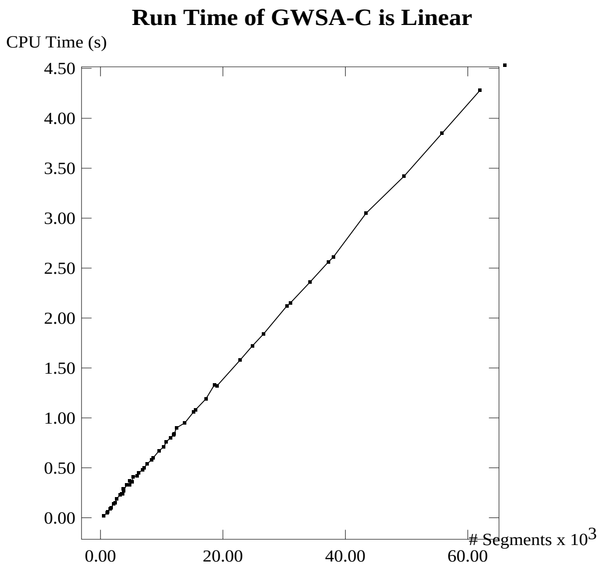

Figure 3. *Run time of GWSA verses number of segments.*

To demonstrate the usefulness of being able to use any starting wire-sizing solution, we solve the problem of minimizing the maximum sink delay of the clock trees r1–r5. This problem is reduced by Lagrangian relaxation to a sequence of weighted sink delay problems. Previously, before solving each weighted sink delay problem, all segments are reset to their minimum possible widths to form the starting solution of GWSA. Our result implies that GWSA will still converge even if we do not reset the segments widths. So in our new approach, we do not reset, and therefore the optimal solution of a weighted sink delay problem is used as a better starting solution to the next one in the sequence. The run time of the previous approach and our new approach are listed in Table 1. For the old approach, each weighted sink delay problem takes about 4 iterations of GWSA. For our approach, each weighted sink delay problem takes only 1.16 iterations of GWSA on average. The overall improvement on the run time is 57.7% on average.

ACKNOWLEDGMENT

We thank Chung-Ping Chen for his help in carrying out the experiments.

Circuit		CPU time (s)		
Name	Size	Old approach	Our approach	Improv.
r1	533	1.95	0.88	54.9%
r2	1195	7.85	3.32	57.7%
r3	1723	11.97	5.09	57.5%
r4	3805	55.34	22.54	59.3%
r5	6201	71.59	29.41	58.9%
			Average:	57.7%

Table 1. *Demonstration of the usefulness of being able to use any starting solution. The run time for the old approach (reset to min-width before each call to GWSA) and our new approach (do not reset) are listed.*

REFERENCES

[1] Chung-Ping Chen, Yao-Wen Chang, and D. F. Wong. Fast performance-driven optimization for buffered clock trees based on Lagrangian relaxation. In *Proc. IEEE Intl. Conf. on Computer-Aided Design*, pages 405–408, 1996.

[2] Chung-Ping Chen and D. F. Wong. A fast algorithm for optimal wire-sizing under Elmore delay model. In *Proc. IEEE ISCAS*, volume 4, pages 412–415, 1996.

[3] Chung-Ping Chen, Hai Zhou, and D. F. Wong. Optimal non-uniform wire-sizing under the Elmore delay model. In *Proc. IEEE Intl. Conf. on Computer-Aided Design*, pages 38–43, 1996.

[4] Chris C. N. Chu, Chung-Ping Chen, and D. F. Wong. Fast and exact simultaneous gate and wire sizing by Lagrangian relaxation. Technical Report TR98–06, Department of Computer Science, University of Texas at Austin, February 1998.

[5] Jason Cong and Lei He. An efficient approach to simultaneous transistor and interconnect sizing. In *Proc. IEEE Intl. Conf. on Computer-Aided Design*, pages 181–186, 1996.

[6] Jason Cong and Lei He. Optimal wiresizing for interconnects with multiple sources. *ACM Trans. Design Automation of Electronic Systems*, 1(4), October 1996.

[7] Jason Cong, Lei He, Cheng-Kok Koh, and Patrick H. Madden. Performance optimization of VLSI interconnect layout. *INTEGRATION, the VLSI Journal*, 21:1–94, 1996.

[8] Jason Cong and Cheng-Kok Koh. Simultaneous driver and wire sizing for performance and power optimization. In *Proc. IEEE Intl. Conf. on Computer-Aided Design*, pages 206–212, 1994.

[9] Jason Cong, Cheng-Kok Koh, and Kwok-Shing Leung. Simultaneous buffer and wire sizing for performance and power optimization. In *Proc. Intl. Symp. on Low Power Electronics and Design*, pages 271–276, August 1996.

[10] Jason Cong and Kwok-Shing Leung. Optimal wiresizing under the distributed Elmore delay model. In *Proc. IEEE Intl. Conf. on Computer-Aided Design*, pages 634–639, 1993.

[11] W. C. Elmore. The transient response of damped linear network with particular regard to wideband amplifiers. *J. Applied Physics*, 19:55–63, 1948.

[12] R. S. Tsay. An exact zero-skew clock routing algorithm. *IEEE Trans. Computer-Aided Design*, 12(2):242–249, February 1993.