

Greedy Wire-Sizing Is Linear Time

Chris C. N. Chu and Martin D. F. Wong, *Member, IEEE*

Abstract—The greedy wire-sizing algorithm (GWSA) has been experimentally shown to be very efficient, but 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 and the minimization of area subject to maximum delay bound using Lagrangian relaxation can be sped up by more than 50%.

Index Terms—Interconnect, performance optimization, wire-sizing.

I. INTRODUCTION

WITH the evolution of very large scale integration (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 the clock cycle are consumed by interconnect delay [8]. 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; see [8] for a survey of recent works. In particular, the problem of minimizing weighted sink delay by wire sizing 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 nonnegative 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 delays from the source to the sinks is minimized. Solving this problem is a key to solving problems with many other important objectives such as minimizing maximum sink delay or minimizing total area subject to bounds on sink delays. It is because [1] and [3] have shown that those problems can all be reduced by Lagrangian relaxation to a sequence of weighted sink delay problems. Thus having efficient algorithms for the weighted sink delay problem is very important for interconnect optimization.

Manuscript received September 24, 1998. This work was supported in part by Texas Advanced Research Program under Grant 003658288 and by a grant from the Intel Corporation. This paper was recommended by Associate Editor M. Sarrafzadeh.

The authors are with the Department of Computer Sciences, University of Texas at Austin, Austin, TX 78712 USA.

Publisher Item Identifier S 0278-0070(99)02321-0.

For the problem of minimizing weighted sink delay under the Elmore delay model [13], a widely used technique is optimal local resizing. The basic idea is to iteratively and greedily resize 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 resized optimally while keeping the widths of all other segments fixed. This technique was first introduced in [11] and was later extended to many other wire, buffer, gate, driver and/or transistor sizing problems [1], [2], [4]–[6], [9], [10].

In [11], 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 greedy wire-sizing algorithm (GWSA). GWSA does not give the optimal solution directly as it can converge to nonoptimal 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, if an arbitrary starting solution is used, it is not known whether GWSA will still converge. Moreover, even for min-width (or max-width) starting solution, the convergence rate is not known.

In this paper, we analyze the convergence of GWSA for continuous wire sizing. We prove that no matter what starting solution is used, GWSA always converges to the optimal solution. Furthermore, we prove that the convergence rate of GWSA is linear for any starting solution. This implies that the run time of GWSA is $O(n \log \frac{1}{\epsilon})$, where n is the number of wire segments and ϵ specifies the precision of the solution (see Theorem 2). Hence, GWSA runs in time linear to n for a fixed precision.

Being able to use an arbitrary starting solution is particularly useful in optimizing other objectives (e.g., minimizing maximum sink delay or minimizing total area subject to bounds on sink delays) 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

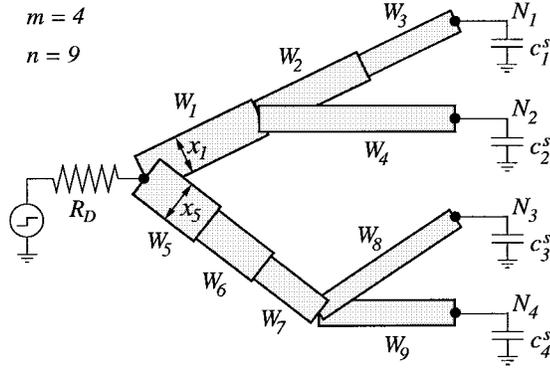


Fig. 1. An example of a routing tree.

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 resetting is much better than the previous approach of resetting each time. We show that our approach can speed up the minimization of maximum delay and minimization of area subject to maximum delay bound using Lagrangian relaxation by more than 50%.

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

II. THE WEIGHTED SINK DELAY PROBLEM AND THE ALGORITHM GWSA

In this section, we 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 $\mathcal{W} = \{W_1, W_2, \dots, W_n\}$, and a set of m sinks $\mathcal{N} = \{N_1, N_2, \dots, N_m\}$ (at the leaves) with load capacitance c_k^s , $1 \leq k \leq m$. Associated with each sink N_k is a nonnegative 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 Fig. 1 for an example of a routing tree.

Let $\text{des}(W_i)$ be the set of descendant wire segments or sinks of W_i (excluding W_i). Let $\text{anc}(W_i)$ be the set of ancestor wire segments of W_i (excluding W_i). Let $\text{path}(N_k)$ 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 Fig. 1, $\text{des}(W_1) = \{W_2, W_3, W_4, N_1, N_2\}$, $\text{anc}(W_1) =$

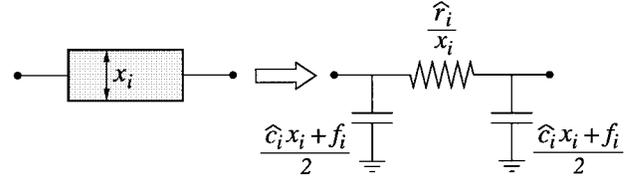


Fig. 2. The model of wire segment W_i by a π -type RC circuit. Note that the resistance and capacitance of the segment are \hat{r}_i/x_i and $\hat{c}_i x_i + f_i$, respectively, where \hat{r}_i is the unit width wire resistance, \hat{c}_i is the unit width wire area capacitance, and f_i is the wire fringing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

$\{W_5, W_6, W_7, W_8\}$, $\text{des}(W_8) = \{N_3\}$, $\text{anc}(W_8) = \{W_5, W_6, W_7\}$, and $\text{path}(N_3) = \{W_5, W_6, W_7, W_8\}$.

For $1 \leq i \leq n$, 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, $L_i \leq x_i \leq U_i$ for $1 \leq i \leq n$. Let $\mathbf{x} = (x_1, x_2, \dots, x_n)$, which is referred to as a wire-sizing solution. A wire segment is modeled as a π -type resistor-capacitor (RC) circuit as shown in Fig. 2. The resistance and capacitance of wire segment W_i are \hat{r}_i/x_i and $\hat{c}_i x_i + f_i$, respectively, where \hat{r}_i is the unit width wire resistance, \hat{c}_i is the unit width wire area capacitance, and f_i is the wire fringing capacitance of W_i .

Let

$$\mu_i = \sum_{N_k \in \text{des}(W_i)} \lambda_k$$

i.e., μ_i is the total downstream sink weight of segment W_i .

Let

$$R_i(\mathbf{x}) = \sum_{W_j \in \text{anc}(W_i)} \mu_j \hat{r}_j / x_j$$

i.e., $R_i(\mathbf{x})$ is the weighted upstream wire resistance of segment W_i .

Let

$$C_i(\mathbf{x}) = \sum_{W_j \in \text{des}(W_i)} \hat{c}_j x_j$$

i.e., $C_i(\mathbf{x})$ is the total downstream wire area capacitance of segment W_i .

Let

$$C_i^{fs} = \sum_{W_j \in \text{des}(W_i)} f_j + \sum_{N_k \in \text{des}(W_i)} c_k^s$$

i.e., C_i^{fs} is the total downstream wire fringing capacitance and sink capacitance of segment W_i .

Let

$$C_T(\mathbf{x}) = \sum_{W_j \in \mathcal{W}} \hat{c}_j x_j$$

i.e., $C_T(\mathbf{x})$ is the total wire area capacitance of the routing tree T .

Let

$$C_T^{fs} = \sum_{W_j \in \mathcal{W}} f_j + \sum_{N_k \in \mathcal{N}} c_k^s$$

i.e., C_T^{fs} is the total wire fringing capacitance and sink capacitance of the routing tree T .

The Elmore delay model [13] is used for delay calculation. For a wire-sizing solution \mathbf{x} , the Elmore delay from the source to the sink N_k is given by

$$D_k(\mathbf{x}) = R_D(C_T(\mathbf{x}) + C_T^{fs}) + \sum_{W_i \in \text{path}(N_k)} \frac{\hat{r}_i}{x_i} \left(\frac{\hat{c}_i x_i}{2} + C_i(\mathbf{x}) + \frac{f_i}{2} + C_i^{fs} \right) \quad (1)$$

Then the weighted sink delay problem can be written as follows:

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

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 resizing the wire segments. In each iteration, the wire segments are examined one by one. When a wire segment W_i is examined, it is resized optimally while keeping the widths of all other segments fixed. This operation is called an optimal local resizing of W_i .

The following lemma gives a formula for optimal local resizing. This lemma is similar to [2, Lemma 1 and Lemma 2]. However, wire fringing capacitance was not considered in the original proof and the notation used in [2] were quite different from those in this paper. So the proof is included for completeness.

Lemma 1: For a wire-sizing solution $\mathbf{x} = (x_1, x_2, \dots, x_n)$, the optimal local resizing of W_i is given by changing the width of W_i to

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

Proof:

$$\begin{aligned} D(\mathbf{x}) &= \sum_{k=1}^m \lambda_k D_k(\mathbf{x}) \\ &= \sum_{k=1}^m \lambda_k \left(R_D(C_T(\mathbf{x}) + C_T^{fs}) + \sum_{W_i \in \text{path}(N_k)} \frac{\hat{r}_i}{x_i} \left(\frac{\hat{c}_i x_i}{2} + C_i(\mathbf{x}) + \frac{f_i}{2} + C_i^{fs} \right) \right) \\ &\quad \text{by (1)} \\ &= \sum_{W_i \in \mathcal{W}} \left(\sum_{N_k \in \text{des}(W_i)} \lambda_k \right) \frac{\hat{r}_i}{x_i} \\ &\quad \cdot \left(\frac{\hat{c}_i x_i}{2} + C_i(\mathbf{x}) + \frac{f_i}{2} + C_i^{fs} \right) + \left(\sum_{k=1}^m \lambda_k \right) \\ &\quad \cdot R_D(C_T(\mathbf{x}) + C_T^{fs}) \end{aligned}$$

$$\begin{aligned} &= \sum_{W_i \in \mathcal{W}} \mu_i \frac{\hat{r}_i}{x_i} \left(\frac{\hat{c}_i x_i}{2} + C_i(\mathbf{x}) + \frac{f_i}{2} + C_i^{fs} \right) \\ &\quad + R_D(C_T(\mathbf{x}) + C_T^{fs}) \\ &= \sum_{W_i \in \mathcal{W}} \mu_i \frac{\hat{r}_i}{x_i} \left(\frac{\hat{c}_i x_i}{2} + \sum_{W_j \in \text{des}(W_i)} \hat{c}_j x_j + \frac{f_i}{2} + C_i^{fs} \right) \\ &\quad + R_D \left(\sum_{W_i \in \mathcal{W}} \hat{c}_i x_i + C_T^{fs} \right) \\ &= \hat{c}_i x_i \left(\sum_{W_j \in \text{anc}(W_i)} \mu_j \frac{\hat{r}_j}{x_j} + R_D \right) \\ &\quad + \mu_i \frac{\hat{r}_i}{x_i} \left(\sum_{W_j \in \text{des}(W_i)} \hat{c}_j x_j + \frac{f_i}{2} + C_i^{fs} \right) \\ &\quad + \text{terms independent of } x_i \\ &= \hat{c}_i x_i (R_i(\mathbf{x}) + R_D) + \frac{\mu_i \hat{r}_i}{x_i} \left(C_i(\mathbf{x}) + \frac{f_i}{2} + C_i^{fs} \right) \\ &\quad + \text{terms independent of } x_i \end{aligned}$$

Note that $R_i(\mathbf{x})$ and $C_i(\mathbf{x})$ are also independent of x_i . So by the Kuhn–Tucker conditions [14], the optimal value of x_i between L_i and U_i which minimizes $\hat{c}_i x_i (R_i(\mathbf{x}) + R_D) + \frac{\mu_i \hat{r}_i}{x_i} (C_i(\mathbf{x}) + \frac{f_i}{2} + C_i^{fs})$ is

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

Let $\text{children}(W_i)$ 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 at the bottom of the next page. Note that since $C_i(\mathbf{x})$ and $R_i(\mathbf{x})$ are computed incrementally in step S3 and S4, each iteration of GWSA takes only $O(n)$ time.

III. CONVERGENCE ANALYSIS OF GWSA

We discuss the convergence of GWSA in this section. In the past, in order to guarantee convergence, GWSA always sets all segments to their minimum (or maximum) possible widths for the starting solution. For example, the original GWSA in [2] sets x_i to L_i for all i in the starting solution. With this starting solution, it can be proved inductively that segment widths are nondecreasing after each optimal local resizing operation. Since each segment width x_i is also upper bounded (by U_i), the wire sizing solution must converge. We can prove similarly that the wire sizing solution will also converge if x_i is set to U_i for all i in the starting solution. However, if an arbitrary starting solution is used, it is previously not known whether GWSA will still converge. Moreover, even for min-width (or max-width) starting solution, the convergence rate of GWSA is not known.

In the rest of this section, we first prove that the algorithm GWSA always converges to the optimal solution for any start-

ing solution (Theorem 1). Then we prove that the convergence rate for any starting solution is always linear with convergence ratio upper bounded by the parameter α defined as follows:

$$\alpha = \max_{1 \leq k \leq n} \left\{ 1 / \left(1 + \frac{R_D}{\sum_{W_i \in \text{anc}(W_k)} \hat{r}_i / L_i} \right), \right. \\ \left. 1 / \left(1 + \frac{\frac{f_k}{2} + C_k^{fs}}{\sum_{W_i \in \text{anc}(W_k)} \hat{c}_i U_i} \right) \right\}.$$

Note that α is a constant such that $0 < \alpha < 1$. This implies that 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 focus on segment W_k for some fixed k . Note that during the n optimal local resizing operations just before the local resizing of W_k at a particular iteration (except the first iteration), each wire segment is resized exactly once. Intuitively, the following two lemmas show that during these n resizing operations, if the changes in all the segment widths are small, then the change in the width x_k during the local resizing of W_k at that iteration will be even smaller.

For some $t \geq 1$, let $\mathbf{x} = (x_1, \dots, x_n)$, $\mathbf{x}' = (x'_1, \dots, x'_n)$ and $\mathbf{x}'' = (x''_1, \dots, x''_n)$ be, respectively, the wire-sizing solutions just before the local resizing of W_k at iteration t , $t+1$ and $t+2$ of GWSA. Let

$$q'_k = \sqrt{\frac{\mu_k \hat{r}_k}{\hat{c}_k} \cdot \frac{C_k(\mathbf{x}) + \frac{f_k}{2} + C_k^{fs}}{R_k(\mathbf{x}) + R_D}}$$

and

$$q''_k = \sqrt{\frac{\mu_k \hat{r}_k}{\hat{c}_k} \cdot \frac{C_k(\mathbf{x}') + \frac{f_k}{2} + C_k^{fs}}{R_k(\mathbf{x}') + R_D}}.$$

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 $\rho > 0$, if $\frac{1}{1+\rho} \leq \frac{x'_i}{x_i} \leq 1 + \rho$ for all i , then $\frac{1}{1+\rho\alpha} \leq \frac{q''_k}{q'_k} \leq 1 + \rho\alpha$.

Proof: If $\frac{1}{1+\rho} x_i \leq x'_i \leq (1+\rho)x_i$ for all i , we have

$$\frac{1}{1+\rho} R_k(\mathbf{x}) \leq R_k(\mathbf{x}') \leq (1+\rho) R_k(\mathbf{x})$$

and

$$\frac{1}{1+\rho} C_k(\mathbf{x}) \leq C_k(\mathbf{x}') \leq (1+\rho) C_k(\mathbf{x}).$$

Since $0 \leq \mu_i \leq 1$ and $x_i \geq L_i$ for all i , we have

$$R_k(\mathbf{x}) = \sum_{W_i \in \text{anc}(W_k)} \mu_i \hat{r}_i / x_i \leq \sum_{W_i \in \text{anc}(W_k)} \hat{r}_i / L_i.$$

By the definition of α , $\alpha \geq 1/(1 + R_D/R_k(\mathbf{x}))$, or equivalently,

$$R_k(\mathbf{x}) \leq \alpha(R_k(\mathbf{x}) + R_D).$$

Hence

$$\begin{aligned} R_k(\mathbf{x}') + R_D &\leq (1+\rho)R_k(\mathbf{x}) + R_D \\ &= \rho R_k(\mathbf{x}) + (R_k(\mathbf{x}) + R_D) \\ &\leq \rho\alpha(R_k(\mathbf{x}) + R_D) + (R_k(\mathbf{x}) + R_D) \\ &= (1+\rho\alpha)(R_k(\mathbf{x}) + R_D) \end{aligned} \quad (2)$$

and

$$\begin{aligned} R_k(\mathbf{x}') + R_D &\geq \frac{1}{1+\rho} R_k(\mathbf{x}) + R_D \\ &= R_k(\mathbf{x}) + R_D - \frac{\rho}{1+\rho} R_k(\mathbf{x}) \\ &\geq R_k(\mathbf{x}) + R_D - \frac{\rho\alpha}{1+\rho} (R_k(\mathbf{x}) + R_D) \\ &= \left(1 - \frac{\rho\alpha}{1+\rho}\right) (R_k(\mathbf{x}) + R_D) \\ &> \frac{1}{1+\rho\alpha} (R_k(\mathbf{x}) + R_D) \\ &\quad \text{as } \rho > 0 \text{ and } 0 < \alpha < 1. \end{aligned} \quad (3)$$

ALGORITHM GWSA:

S1. Let \mathbf{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:

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

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

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

$$C_i(\mathbf{x}) := \sum_{W_j \in \text{children}(W_i)} (C_j(\mathbf{x}) + \hat{c}_j x_j)$$

S4. Perform a top-down traversal of T :

For each W_i ,

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

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

S5. Repeat Step S3–S4 until no improvement.

Similarly, since $x_i \leq U_i$ for all i , we have $C_k(\mathbf{x}) = \sum_{W_i \in \text{anc}(W_k)} \hat{c}_i x_i \leq \sum_{W_i \in \text{anc}(W_k)} \hat{c}_i U_i$. By the definition of α , $\alpha \geq 1/(1 + (f_k/2 + C_k^{fs})/C_k(\mathbf{x}))$, or equivalently

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

Hence, we can prove similarly that

$$C_k(\mathbf{x}') + \frac{f_k}{2} + C_k^{fs} \leq (1 + \rho\alpha) \left(C_k(\mathbf{x}) + \frac{f_k}{2} + C_k^{fs} \right) \quad (4)$$

and

$$C_k(\mathbf{x}') + \frac{f_k}{2} + C_k^{fs} > \frac{1}{1 + \rho\alpha} \left(C_k(\mathbf{x}) + \frac{f_k}{2} + C_k^{fs} \right). \quad (5)$$

By definitions of q'_k and q''_k , and by (3) and (4), we have

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

Similarly, by (2) and (5), we can prove that $q''_k \geq \frac{1}{1 + \rho\alpha} q'_k$.

As a result, $\frac{1}{1 + \rho\alpha} \leq \frac{q''_k}{q'_k} \leq 1 + \rho\alpha$. \square

Lemma 3: For any $\rho > 0$, if $\frac{1}{1 + \rho} \leq \frac{x'_i}{x_i} \leq 1 + \rho$ for all i , then $\frac{1}{1 + \rho\alpha} \leq \frac{x''_k}{x'_k} \leq 1 + \rho\alpha$.

Proof: By Lemma 2, if $\frac{1}{1 + \rho} x_i \leq x'_i \leq (1 + \rho)x_i$ for all i , then $\frac{1}{1 + \rho\alpha} q'_k \leq q''_k \leq (1 + \rho\alpha) q'_k$. 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 $\frac{1}{1 + \rho\alpha} x'_k \leq x''_k$, we consider three cases.

Case 1) $q'_k < L_k$.

Then $x'_k = L_k$. So $\frac{1}{1 + \rho\alpha} x'_k = \frac{1}{1 + \rho\alpha} L_k < L_k \leq x''_k$.

Case 2) $q''_k > U_k$.

Then $x''_k = U_k$. So $\frac{1}{1 + \rho\alpha} x'_k \leq \frac{1}{1 + \rho\alpha} U_k < U_k = x''_k$.

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 $\frac{1}{1 + \rho\alpha} x'_k \leq \frac{1}{1 + \rho\alpha} q'_k \leq q''_k \leq x''_k$.

In order to prove $x''_k \leq (1 + \rho\alpha) x'_k$, we consider another three cases.

Case 1) $q'_k > U_k$.

Then $x'_k = U_k$. So $x''_k \leq U_k < (1 + \rho\alpha) U_k = (1 + \rho\alpha) x'_k$.

Case 2) $q''_k < L_k$.

Then $x''_k = L_k$. So $x''_k = L_k < (1 + \rho\alpha) L_k \leq (1 + \rho\alpha) x'_k$.

Case 3) $q'_k \leq U_k$ and $q''_k \geq L_k$.

Then $q'_k \leq U_k \Rightarrow q'_k \leq x'_k$ and $q''_k \geq L_k \Rightarrow x''_k \leq q''_k$. So $x''_k \leq q''_k \leq (1 + \rho\alpha) q'_k \leq (1 + \rho\alpha) x'_k$.

As a result, $\frac{1}{1 + \rho\alpha} \leq \frac{x''_k}{x'_k} \leq 1 + \rho\alpha$. \square

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

Lemma 4: For any $t \geq 0$ and $\rho > 0$, if $\frac{1}{1 + \rho} \leq \frac{x_i^{(t+1)}}{x_i^{(t)}} \leq 1 + \rho$ for all i , then $\frac{1}{1 + \rho\alpha} \leq \frac{x_i^{(t+2)}}{x_i^{(t+1)}} \leq 1 + \rho\alpha$ for all i .

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, \dots, W_n . The lemma can be proved by induction on i .

Base Case)

Consider the wire segment W_1 . Before the local resizing of W_1 , the wire-sizing solutions at iteration $t + 1$ and $t + 2$ are $(x_1^{(t)}, x_2^{(t)}, \dots, x_n^{(t)})$ and $(x_1^{(t+1)}, x_2^{(t+1)}, \dots, x_n^{(t+1)})$, respectively. Since $\frac{1}{1 + \rho} \leq \frac{x_i^{(t+1)}}{x_i^{(t)}} \leq 1 + \rho$ for all i , by Lemma 3, we have $\frac{1}{1 + \rho\alpha} \leq \frac{x_1^{(t+2)}}{x_1^{(t+1)}} \leq 1 + \rho\alpha$.

Induction Step)

Assume that the induction hypothesis is true for $i = 1, \dots, k - 1$. Before the local resizing of W_k , the wire-sizing solutions at iteration $t + 1$ and $t + 2$ are $(x_1^{(t+1)}, \dots, x_{k-1}^{(t+1)}, x_k^{(t)}, \dots, x_n^{(t)})$ and $(x_1^{(t+2)}, \dots, x_{k-1}^{(t+2)}, x_k^{(t+1)}, \dots, x_n^{(t+1)})$, respectively. By induction hypothesis, $\frac{1}{1 + \rho\alpha} \leq \frac{x_i^{(t+2)}}{x_i^{(t+1)}} \leq 1 + \rho\alpha$ for $i = 1, \dots, k - 1$.

Hence, $\frac{1}{1 + \rho} \leq \frac{x_i^{(t+2)}}{x_i^{(t+1)}} \leq 1 + \rho$ for $i = 1, \dots, k - 1$, as $\alpha < 1$. Also, it is given that $\frac{1}{1 + \rho} \leq \frac{x_i^{(t+1)}}{x_i^{(t)}} \leq 1 + \rho$ for $i = k, \dots, n$.

So by Lemma 3, $\frac{1}{1 + \rho\alpha} \leq \frac{x_k^{(t+2)}}{x_k^{(t+1)}} \leq 1 + \rho\alpha$.

Hence, the lemma is proved. \square

Let $\Delta = \max_{1 \leq i \leq n} \left\{ \frac{U_i - L_i}{L_i} \right\}$.

Lemma 5: For any $t \geq 0$, $\frac{1}{1 + \Delta\alpha^t} \leq \frac{x_i^{(t+1)}}{x_i^{(t)}} \leq 1 + \Delta\alpha^t$ for all i .

Proof: This can be proved by induction on t .

Base Case)

Consider $t = 0$. Note that for any wire-sizing solution $\mathbf{x} = (x_1, \dots, x_n)$, $L_i \leq x_i \leq U_i$ for all i . For all i , $\frac{x_i^{(1)}}{x_i^{(0)}} \leq \frac{U_i}{L_i} \leq 1 + \frac{U_i - L_i}{L_i} \leq 1 + \Delta$. Similarly, we can prove that for all i , $\frac{x_i^{(1)}}{x_i^{(0)}} \geq \frac{1}{1 + \Delta}$.

Induction Step)

Assume that the induction hypothesis is true for t . Therefore, $\frac{1}{1 + \Delta\alpha^t} \leq \frac{x_i^{(t+1)}}{x_i^{(t)}} \leq 1 + \Delta\alpha^t$ for all i . By Lemma 4, $\frac{1}{1 + \Delta\alpha^{t+1}} \leq \frac{x_i^{(t+2)}}{x_i^{(t+1)}} \leq 1 + \Delta\alpha^{t+1}$ for all i .

Hence, the lemma is proved. \square

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

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

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

Lemma 6: For any $t \geq 0$, $|\frac{x_i^{(t)} - x_i^*}{x_i^*}| \leq \frac{(1+\Delta)\Delta\alpha^t}{1-\alpha}$ for all i .

Proof: For any $t \geq 0$ and for any i

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

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

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

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, $\frac{1}{P} \leq$

$\frac{x_i^{(t)}}{x_i^*} \leq P$ where $P = \prod_{k=t}^{\infty} (1 + \Delta\alpha^k)$.

$$\begin{aligned} \ln P &= \sum_{k=t}^{\infty} \ln(1 + \Delta\alpha^k) \\ &= \sum_{k=t}^{\infty} \left(\Delta\alpha^k - \frac{1}{2}\Delta^2\alpha^{2k} + \frac{1}{3}\Delta^3\alpha^{3k} \right. \\ &\quad \left. - \frac{1}{4}\Delta^4\alpha^{4k} + \dots \right) \end{aligned} \quad (6)$$

$$\begin{aligned} &\leq \sum_{k=t}^{\infty} \left(\sum_{j=1}^{\infty} \frac{1}{j} \Delta^j \alpha^{jk} \right) \\ &= \sum_{j=1}^{\infty} \frac{\Delta^j}{j} \sum_{k=t}^{\infty} (\alpha^j)^k \\ &= \sum_{j=1}^{\infty} \frac{\Delta^j}{j} \frac{\alpha^{jt}}{1 - \alpha^j} \\ &\leq \sum_{j=1}^{\infty} \frac{\Delta^j}{j} \frac{\alpha^{jt}}{(1 - \alpha)^j} \end{aligned} \quad (7)$$

$$= \ln \frac{1}{1 - \frac{\Delta\alpha^t}{1-\alpha}} \quad (8)$$

where (6) is because $\ln(1+x) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}x^4 + \dots$, (7) is because $0 < \alpha < 1$, which implies $0 < (1-\alpha)^j < 1-\alpha < 1-\alpha^j$ for $j \geq 1$, and (8) is because $0 < \frac{\Delta\alpha^t}{1-\alpha} < \frac{(1+\Delta)\alpha^t}{1-\alpha} < 1$ and $\ln \frac{1}{1-x} = x + \frac{1}{2}x^2 + \frac{1}{3}x^3 + \dots$ if $0 < x < 1$. So

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

Hence

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

Therefore, for both cases

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

It is easy to see that

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

So for any $t \geq 0$ and for all i

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

□

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 \mathbf{x}^* , the final solution \mathbf{x} satisfies $|(x_i^* - x_i)/x_i^*| \leq \epsilon$ for all i).

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

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

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

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

or equivalently

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

In other words, at most $\lceil \log_{\frac{1}{\alpha}} \frac{(1+\Delta)\Delta}{(1-\alpha)\epsilon} \rceil$ 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 $O(n)$. As suggested by the proof of Theorem 2, the number of iterations required for convergence depends on the parameters α , Δ , and ϵ . The effect of these parameters on the number of iterations is discussed in the next section. We also demonstrate in Section IV that GWSA usually takes only a few iterations in practice.

IV. EXPERIMENTAL RESULTS AND CONCLUDING REMARKS

In this section, we 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 a PC with a 200-MHz Pentium Pro processor and 32 MB of memory.

Fig. 3 shows the linearity of the run time of GWSA. We use the clock trees obtained by applying the routing algorithm in [12] to the data $r1-r5$ in [15]. The number of segments in these trees range from 533 to 6201. In order to have more data

TABLE I
DEMONSTRATION OF THE ADVANTAGE 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

Circuit		Objective to minimize	CPU time (s)		Improvement
Name	Size		Old approach	Our approach	
r1	533	Max. delay	1.95	0.88	54.9%
		Area with delay bound	3.09	1.12	63.8%
r2	1195	Max. delay	7.85	3.32	57.7%
		Area with delay bound	12.23	5.31	56.6%
r3	1723	Max. delay	11.97	5.09	57.5%
		Area with delay bound	12.93	6.24	51.7%
r4	3805	Max. delay	55.34	22.54	59.3%
		Area with delay bound	41.52	20.58	50.4%
r5	6201	Max. delay	71.59	29.41	58.9%
		Area with delay bound	78.43	39.18	50.0%
Average:					56.1%

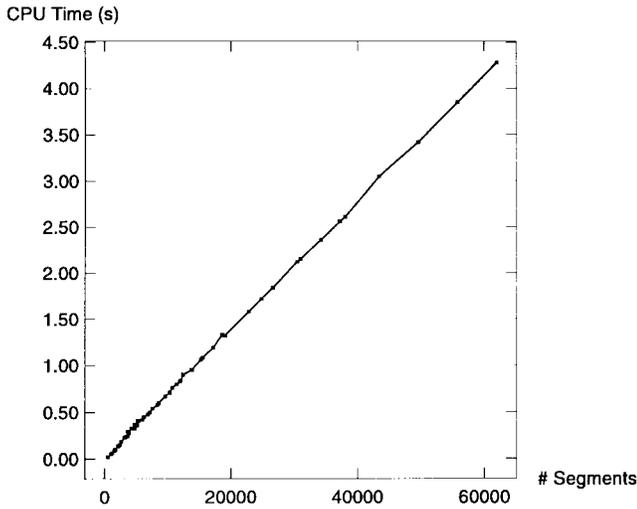


Fig. 3. Run time of GWSA versus number of segments. Run time of GWSA is linear.

points, we construct ten trees from each tree by dividing each tree edge into k segments where $k = 1, \dots, 10$. So we have 50 trees with the number of segments ranging from 533 to 62010. The run time is plotted against the number of segment in Fig. 3. It can be seen that the run time of GWSA is linear in practice.

The number of iterations required for convergence depends on the parameters α , Δ , and ϵ as suggested by the proof of Theorem 2. For the experiment above, Δ equals nine and ϵ equals 10^{-5} . α is defined at the beginning of Section III and its value may be large (close to one) if it is calculated according to the definition. However, α is defined in such a way that it can be conveniently used in the proofs. It is just a loose upper bound on the convergence ratio. For the data shown in Fig. 3, we observe the actual convergence ratio is only around 0.015–0.03. The number of iterations is just six for most cases. In fact, even for some very extreme parameters like $\alpha = 0.7$, $\Delta = 99$, and $\epsilon = 10^{-7}$, the number of iterations is at most $\lceil \log_{\frac{1}{\alpha}}((1 + \Delta)\Delta)/((1 - \alpha)\epsilon) \rceil = 75$ only.

We also investigate the dependency of the convergence ratio on the technology parameters. If we use more advanced technology, the ratio of the wire fringing capacitances and sink capacitances to the wire area capacitance will become larger,

which will result in faster convergence. On the other hand, the ratio of the driver resistance to the wire resistances will become smaller, which will result in slower convergence. To find out which of these two factors is dominating, we run GWSA with different sets of technology parameters listed in [7]. We observe that for more advanced technology, the convergence ratio is smaller. For example, for the $0.13 \mu\text{m}$ technology, the actual convergence ratio is only around 0.006–0.015.

To demonstrate the advantage of being able to use an arbitrary starting wire-sizing solution, we optimize the clock trees $r1$ – $r5$ using two different objectives. The first objective is to minimize the maximum sink delay. The second objective is to minimize the total wire area subject to a delay bound of 10% more than the minimized maximum sink delay. These problems are 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 segment 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 I. For the old approach, each weighted sink delay problem takes 3.67 iterations of GWSA on average. For our approach, each weighted sink delay problem takes only 1.12 iterations of GWSA on average. The overall improvement on the run time is 56.1% on average.

For future research, an interesting problem to look at is whether our work can be extended to discrete wire sizing. Another interesting problem is to analyze the convergence of other local resizing based algorithms by our analysis technique. The local resizing idea has been successfully applied to many other wire, buffer, gate, driver and/or transistor sizing problems [1], [4]–[6], [9], [10]. All these algorithms have been shown experimentally to be very efficient. However, just like the case of wire sizing alone, these algorithms are known to converge only for min-width/size or max-width/size starting solution, and the convergence rates are not known even for min-width/size or max-width/size starting solution. In the future, we would like to investigate the convergence rates of these algorithms with arbitrary starting solution.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their helpful comments and thank C.-P. Chen for his help in carrying out the experiments.

REFERENCES

- [1] C.-P. Chen, Y.-W. Chang, and D. F. Wong, "Fast performance-driven optimization for buffered clock trees based on Lagrangian relaxation," in *Proc. ACM/IEEE Design Automation Conf.*, 1996, pp. 405–408.
- [2] C.-P. Chen and D. F. Wong, "A fast algorithm for optimal wire-sizing under Elmore delay model," in *Proc. IEEE ISCAS*, 1996, vol. 4, pp. 412–415.
- [3] C.-P. Chen, H. Zhou, and D. F. Wong, "Optimal nonuniform wire-sizing under the Elmore delay model," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 1996, pp. 38–43.
- [4] C. C. N. Chu, C.-P. Chen, and D. F. Wong, "Fast and exact simultaneous gate and wire sizing by Lagrangian relaxation," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 1998, pp. 617–624.
- [5] J. Cong and L. He, "An efficient approach to simultaneous transistor and interconnect sizing," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 1996, pp. 181–186.
- [6] ———, "Optimal wiresizing for interconnects with multiple sources," *ACM Trans. Design Automation of Electronic Systems*, vol. 1, no. 4, Oct. 1996.
- [7] J. Cong, L. He, K.-Y. Khoo, C.-K. Koh, and Z. Pan, "Interconnect design for deep submicron IC's," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 1997, pp. 478–485.
- [8] J. Cong, L. He, C.-K. Koh, and P. H. Madden, "Performance optimization of VLSI interconnect layout," *INTEGRATION, the VLSI J.*, vol. 21, pp. 1–94, 1996.
- [9] J. Cong and C.-K. Koh, "Simultaneous driver and wire sizing for performance and power optimization," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 1994, pp. 206–212.
- [10] J. Cong, C.-K. Koh, and K.-S. Leung, "Simultaneous buffer and wire sizing for performance and power optimization," in *Proc. Int. Symp. on Low-Power Electronics and Design*, Aug. 1996, pp. 271–276.
- [11] J. Cong and K.-S. Leung, "Optimal wiresizing under the distributed Elmore delay model," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 1993, pp. 634–639.
- [12] M. Edahiro, "A clustering-based optimization algorithm in zero-skew routings," in *Proc. ACM/IEEE Design Automation Conf.*, 1993, pp. 612–616.
- [13] W. C. Elmore, "The transient response of damped linear network with particular regard to wideband amplifiers," *J. Appl. Phys.*, vol. 19, pp. 55–63, 1948.
- [14] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed. Reading, MA: Addison Wesley, 1984.
- [15] R. S. Tsay, "An exact zero-skew clock routing algorithm," *IEEE Trans. Computer-Aided Design*, vol. 12, no. 2, pp. 242–249, Feb. 1993.



Chris C. N. Chu received the B.S. degree in computer science from the University of Hong Kong, Hong Kong, in 1993, and the M.S. degree in computer science from the University of Texas at Austin, Austin, in 1994. He is now pursuing the Ph.D. degree at the computer sciences department of the University of Texas at Austin.

His research interests include design and analysis of algorithms, computer-aided design of VLSI physical design, and performance-driven interconnect optimization.

Martin D. F. Wong (M'88) for a photograph and biography, see this issue, p. 374.