

Gate Sizing by Lagrangian Relaxation Revisited^{*}

Jia Wang, Debasish Das, and Hai Zhou
Electrical Engineering and Computer Science
Northwestern University
Evanston, IL 60208

Abstract—In this paper, we formulate the Generalized Convex Sizing (GCS) problem that unifies and generalizes the sizing problems. We revisit the approach to solve the sizing problem by Lagrangian relaxation, point out several misunderstandings in the previous works, and extend the approach to handle general convex delay functions in the GCS problems. We identify a class of proper GCS problems whose objective functions in the simplified dual problem are differentiable and show many practical sizing problems, including the simultaneous sizing and clock skew optimization problem, are proper. We design an algorithm based on the method of feasible directions to solve proper GCS problems. The algorithm will provide evidences for infeasible GCS problems according to a condition derived by us. Experimental results confirm the efficiency and the effectiveness of our algorithm when the Elmore delay model is used.

I. INTRODUCTION

The transistor sizing, gate sizing, and wire sizing problems [1], [2], [3], [4], [5], [6] are important problems in VLSI design because they allow to explore the trade-offs between the performance and the cost of the system. Since all these problems share the same structure that the timing constraints are formulated as a system of difference inequalities involving the delays of each individual components and the arrival times, we call them collectively as the sizing problems.

Most research works on the sizing problem use a convex delay model for individual components. It had been shown in the work TILOS by Fishburn et al. [1] that a few problem formulations concerning the total size and the clock period are convex programming problems under such delay model. Convex programming problems have the advantage that a local optimum is a global one and they have been studied for decades (see [7], [8] for references). In TILOS, the Elmore delay model [9] was used for transistor delays and a heuristic that sizes the transistors iteratively according to the sensitivities of the critical path delay to the transistor sizes was proposed to find an optimum. The Elmore delays are special cases of posynomials, which are a class of convex functions under the logarithm variable transformation. Sapatnekar et al. [2] applied the algorithms that solve general convex programming problems to solve the sizing problem for the delays as the posynomials of the sizes. Kasamsetty et al. [10] proposed to use the generalized posynomials, which are also convex under the logarithm variable transformation, to approximate the delays more accurately than the Elmore delay model and solved the sizing problem with this model in the optimization framework of the work [2]. However, the experimental results in the works [2] and [10] showed that the general algorithms were not efficient for the sizing problems with even less than 1000 sizable components.

The special structure in the sizing problem that the timing constraints are formulated as a system of difference inequalities had been exploited by Chen et al. [4] to design an algorithm that solves the gate and wire sizing problem by Lagrangian relaxation. The structure allows to simplify the Lagrangian dual problem using the Karush-Kuhn-Tucker (KKT) conditions for an optimal solution. The dual problem was then solved by subgradient optimization. The gate and wire delay model used in this work was the Elmore delay model and

thus the Lagrangian subproblem was a convex optimization problem with simple constraints, which can be solved efficiently. Although the approach was efficient for sizing adders in the work [4], Tennakoon et al. [6] showed that for general circuits, it is very hard to choose proper initial solution and step sizes for subgradient optimization to converge practically. Heuristics were developed in [6] to obtain a good initial solution and to speed up the convergence of subgradient optimization. However, it is not clear whether the heuristics can be extended to handle more sophisticated and accurate convex delay models, e.g. the ones in the work [10].

The ever-increasing complexity in modern VLSI systems demands efficient and effective sizing algorithms to handle the sophisticated convex delay models and the tremendous number of sizable components. The difficulties in the previous works motivate us to design new algorithms for the sizing problem. We revisit the Lagrangian relaxation based approach [4]. This particular approach is of our interests because the special structure of the sizing problem is exploited. Our contributions in this work include:

- 1) We formulate the Generalized Convex Sizing (GCS) problem that unifies and generalizes the sizing problem.
- 2) We revisit the approach to formulate the Lagrangian dual problem by Lagrangian relaxation and to simplify the dual problem by the KKT conditions. Several misunderstandings are corrected and the approach is extended to handle general convex delay models.
- 3) We identify a class of the GCS problems called the proper GCS problems that most practical sizing problems belong to. We propose a problem formulation technique to formulate the simultaneous sizing and clock skew optimization problem as a proper GCS problem.
- 4) We prove that the objective function in the simplified dual problem of a proper GCS problem is differentiable and then design the DualFD algorithm to solve the proper GCS problems by the method of feasible directions.
- 5) We derive the necessary and sufficient condition for the GCS problem to be feasible. It allows to check the feasibility in the DualFD algorithm and to identify the troublesome part of the circuit that makes it infeasible.

The rest of this paper is organized as follows. In Section II, we examine the convex gate delay model, propose the Generalized Convex Sizing (GCS) problem, and then formulate the simultaneous sizing and clock skew optimization problem as a proper GCS problem. In Section III, Lagrangian relaxation based approaches that formulate the dual problems are revisited. In Section IV, after the methods to solve the dual problem by subgradient optimization are reviewed, we present our DualFD algorithm that solves the proper GCS problems. Experimental results are reported in Section V. Section VI concludes the paper.

II. PROBLEM FORMULATION

A. Convex Gate Delay Modeling

Previous gate sizing works [4], [6] assumed a single delay function for the timing arcs that correspond to the input-to-output delays in

^{*}This work was supported in part by NSF under CCR-0238484 and CNS-0613967 and in part by a grant from Intel.

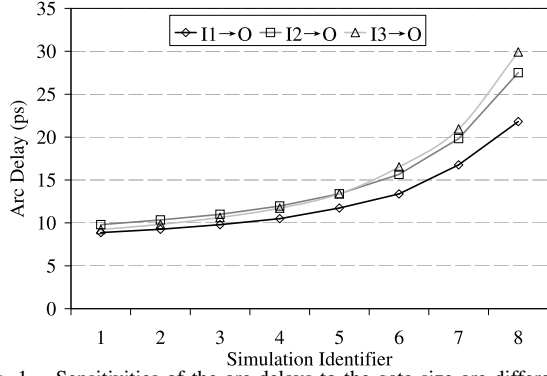


Fig. 1. Sensitivities of the arc delays to the gate size are different.

a gate. However, this assumption is not accurate even with uniform NMOS transistor sizes, PMOS transistor sizes, and input slews. As an example, we simulate the SPICE netlist of a 3-input NAND gate from Faraday 90nm [11] standard cell library. The input ports are I1, I2, and I3 while the output port is O. The three timing arcs are I1 → O, I2 → O, and I3 → O. Eight simulations are performed with decreasing gate sizes, fixed input slews, and fixed output capacitance. In each simulation, the input slews at all the input ports are assumed to be the same. We also assume that the sizes of the PMOS transistors are the same and so are those of the NMOS transistors. This assumption was successfully applied in [6] to establish the delay model. For the i th simulation, the size of the PMOS transistors x_p and that of the NMOS transistors x_n satisfy that $x_p = x_{p0} - (i - 1)\delta$ and $x_n = x_{n0} - (i - 1)\delta$ where x_{p0} , x_{n0} , and δ are constants. The simulation results are shown in Figure 1. It is clear that the sensitivities of the arc delays to the gate size are different. Therefore, we prefer to assign different delay functions to different timing arcs in a gate. Note that such modeling technique was applied when a more accurate delay model considering input slews was proposed in [12].

B. The Generalized Convex Sizing Problem

Although gate sizing is commonly applied to the combinational part of a circuit, for a sequential circuit with flip-flops (FF) as the storage units, the clock period and the clock skews can be incorporated into the timing constraints as the following system of difference inequalities.

$$\begin{aligned}
 t_i + d_{i,j} &\leq t_j, & \forall (i,j) \in E, \\
 t_{pi} &= a_{pi}, t_{po} = r_{po}, & \forall pi \in PI, po \in PO, \\
 t_{Q_k} &= s_k, t_{D_k} = s_k + T, & \forall k \in FF.
 \end{aligned} \quad (1)$$

Here the topology of the combinational part of the circuit is represented by a directed acyclic graph (DAG) $G = (V, E)$. The vertices represent ports and the edges represent interconnects and timing arcs. The variable t_v is the arrival time at the vertex v . The function $d_{i,j}$ is the delay of the edge (i, j) . It is either the delay of a wire or a timing arc in a gate. The constants a_{pi} and r_{po} are the arrival times and the required arrival times of the primary input and output ports respectively. The clock period is T . The constants s_k are the clock skews of the FFs while D_k and Q_k are the data input and output ports of the FFs respectively.

Eq. (1) can be expressed in a more consistent manner by extending G as follows. Two vertices I and O are added to V . The edges (I, pi) and (po, O) are added to E with $d_{I,pi} = a_{pi}$ and $d_{po,O} = T - r_{po}$ for all $pi \in PI$ and $po \in PO$. Then the edges (I, Q_k) and (D_k, O) are added to E with $d_{I,Q_k} = s_k$ and $d_{D_k,O} = -s_k$ for each $k \in FF$.

Finally the edge (O, I) is added to E with $d_{O,I} = -T$. We will refer the extended graph as G for the ease of presentation when there is no ambiguity. The extended graph G of an example circuit is shown in Figure 2.

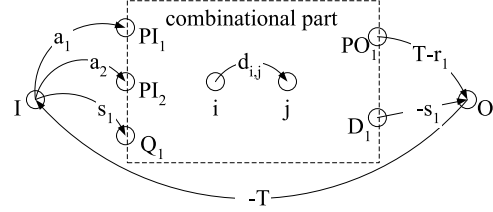


Fig. 2. Timing constraints of a sequential circuit.

In the extended graph G , Eq. (1) becomes $t_i + d_{i,j} \leq t_j$, $\forall (i, j) \in E$. This motivates us to formulate the *Generalized Convex Sizing* (GCS) problem as follows.

Problem 1 (Generalized Convex Sizing): Let $G = (V, E)$ be a directed graph representing the structure of a system with the parameters $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ belonging to the set $\Omega \triangleq \{\mathbf{x} : l_k \leq x_k \leq u_k, \forall 1 \leq k \leq n\}$ where l_k and u_k , $\forall 1 \leq k \leq n$, are constants. The edge delays $d_{i,j}$, $\forall (i, j) \in E$, and the objective function C are twice differentiable convex function for $\mathbf{x} \in \Omega$. Solve that

$$\begin{aligned}
 \text{Minimize} \quad & C(\mathbf{x}) \\
 \text{s.t.} \quad & t_i + d_{i,j}(\mathbf{x}) - t_j \leq 0, \forall (i, j) \in E, \\
 & \mathbf{x} \in \Omega.
 \end{aligned} \quad (2)$$

Let \mathbf{t} be the vector of all the $t_i, \forall i \in V$. The decision variables in the GCS problem are (\mathbf{x}, \mathbf{t}) . The GCS problem is a convex programming problem (see Chapter 4.2 [7]) since the objective function C and the left-hand-side of Eq. (2) are all convex functions and the set Ω is a convex set. Define (\mathbf{x}, \mathbf{t}) to be feasible if Eq. (2) and (3) are satisfied, \mathbf{t} to be feasible with respect to \mathbf{x} if (\mathbf{x}, \mathbf{t}) is feasible, and \mathbf{x} to be feasible if there exists a feasible \mathbf{t} with respect to \mathbf{x} . Denote the set of all the feasible \mathbf{x} by \mathcal{X} .

It is clear that the GCS problems are not restricted to the sizing problem where the delay functions are the posynomials of gate sizes and the convexity is established through geometric programming. One trivial but important group of the convex delay functions that are not posynomials of sizes is the linear combination of \mathbf{x} with negative coefficients.

We define the *proper* GCS problems as follows.

Definition 1: A GCS problem is proper iff the Hessian matrix of its objective function for any $\mathbf{x} \in \Omega$ is positive definite.

We are interested in the proper GCS problems because their simplified dual functions are differentiable as proved later, which is a much desirable property for designing optimization algorithms. This definition only depends on the property of the objective function but not that of the delay functions. Many practical sizing problems are the proper GCS problems as shown in the following theorem and corollary.

Theorem 1: Assume that the objective function C of a GCS problem is a posynomial of the variables $e^{x_k}, \forall 1 \leq k \leq n$, i.e. $C(\mathbf{x}) = \sum_{i=1}^l c_i e^{\mathbf{b}_i^T \mathbf{x}}$ where $c_i > 0, \forall 1 \leq i \leq l$. Then the GCS problem is proper iff $\text{rank}(B) = n$, where $B = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_l)$ is a matrix with n rows and l columns.

Corollary 1: For a GCS problem, if $C(\mathbf{x}) = \sum_{k=1}^n w_k e^{x_k}$ where $w_k > 0, \forall 1 \leq k \leq n$, then the problem is proper.

C. Simultaneous Sizing and Clock Skew Optimization as a Proper GCS Problem

The simultaneous sizing and clock skew optimization problem was studied by Chuang et al. [3]. They proposed an algorithm to

solve the problem considering both the long path (setup) and the short path (hold) conditions by formulating a linear programming problem using the piece-wise-linear (PWL) approximations of the convex delays. However, if we consider a path p from Q_i to D_j for the FF i and j with the non-linear convex delay d_p , the short path condition $-s_i - d_p + s_j + \text{hold-time} \leq 0$ is not convex because the left-hand-side is not convex. Thus such PWL approximation may result in suboptimal solutions. We propose to consider the long path conditions only to maintain the convexity of the problem and assume that a post-processing, e.g. [13], will repair the violated short path conditions.

Suppose that each clock skew s_k to be optimized belongs to a pre-defined range $[s_k^-, s_k^+]$ where s_k^- and s_k^+ are constants. Assume that the objective function represents the positive weighted summation of the sizes. It is straightforward that the simultaneous sizing and clock skew optimization problem is a GCS problem where the clock skews to be optimized are among the variables \mathbf{x} and the pre-defined ranges are part of the set Ω . However, the problem is not proper. We show how to transform the problem into a proper one by eliminating the clock skew variables. There are two advantages: first, the number of the variables are reduced; second, since the result problem is a proper GCS problem, the algorithm presented later is applicable.

There is a valid skew assignment for the FF k iff there exists t_l , t_o , t_{Q_k} , t_{D_k} , and s_k satisfying

$$(t_l + s_k \leq t_{Q_k}) \wedge (t_{D_k} - s_k \leq t_o) \wedge (s_k^- \leq s_k \leq s_k^+).$$

That is $[t_{D_k} - t_o, t_{Q_k} - t_l] \cap [s_k^-, s_k^+] \neq \emptyset$, which is equivalent to

$$(t_{D_k} - s_k^+ \leq t_o) \wedge (t_l + s_k^- \leq t_{Q_k}) \wedge (t_{D_k} - t_{Q_k} \leq t_o - t_l). \quad (4)$$

Modify the graph G by adding an edge (D_k, Q_k) with $d_{D_k, Q_k} = -T$, setting $d_{D_k, O} = -s_k^+$, and setting $d_{l, Q_k} = s_k^-$. Recall that $t_o - t_l \leq T$. According to Eq. (4), any feasible \mathbf{t} with the unmodified G will be feasible with the modified G . On the other hand, for any feasible \mathbf{t} with the modified graph G , t_o can be increased to $t_l + T$ without violating the constraints and changing the cost. Then Eq. (4) holds and the s_k exists. So we no longer has the clock skew variables. We have,

Theorem 2: The simultaneous sizing and clock skew optimization problem can be formulated as a proper GCS problem without introducing variables representing the clock skews to be optimized.

In Figure 3, the modified graph of the same example circuit as shown in Figure 2 is shown, which allows optimizing the clock skew of the FF 1.

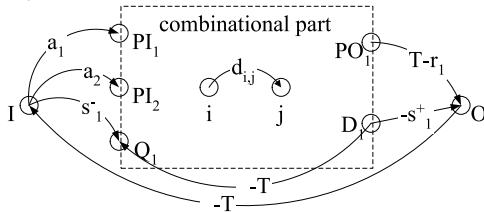


Fig. 3. Optimize the clock skew of the FF 1 without a clock skew variable.

III. SOLVING GCS VIA LAGRANGIAN DUAL PROBLEMS

We follow Chen et al. [4] to formulate the Lagrangian dual problem of the GCS problem and to simplify the dual problem. Although the formulations are similar, we revisit the assumptions and correct the misunderstandings for better understanding of the approach.

A. The Lagrangian Dual Problem

Let $f_{i,j}$ be the Lagrangian multipliers associated with each inequality in Eq. (2). Let \mathbf{f} be the vector for all the $f_{i,j}$. Let $L^*(\mathbf{x}, \mathbf{t}, \mathbf{f})$, $L(\mathbf{f})$,

and \mathcal{N} be the Lagrangian function, the Lagrangian dual function, and the set of the non-negative multipliers respectively, i.e.,

$$\begin{aligned} L^*(\mathbf{x}, \mathbf{t}, \mathbf{f}) &\triangleq C(\mathbf{x}) + \sum_{(i,j) \in E} f_{i,j} (t_i + d_{i,j}(\mathbf{x}) - t_j), \\ L(\mathbf{f}) &\triangleq \inf\{L^*(\mathbf{x}, \mathbf{t}, \mathbf{f}) : \mathbf{x} \in \Omega, \mathbf{t} \in R^{|V|}\}, \\ \mathcal{N} &\triangleq \{\mathbf{f} : f_{i,j} \geq 0, \forall (i,j) \in E\}. \end{aligned}$$

The Lagrangian dual problem *D-GCS* is formulated as follows.

Problem 2 (D-GCS):

$$\begin{aligned} \text{Maximize} \quad & L(\mathbf{f}) \\ \text{s.t.} \quad & \mathbf{f} \in \mathcal{N}. \end{aligned}$$

Recall that \mathcal{X} is the set of all the feasible \mathbf{x} . The weak duality theorem (see Chapter 6.2 [8]) states that the duality gap is non-negative, i.e.,

$$\inf\{C(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\} \geq \sup\{L(\mathbf{f}) : \mathbf{f} \in \mathcal{N}\}. \quad (5)$$

The approach that solves the GCS problem by solving the D-GCS problem requires a zero duality gap, i.e.,

$$\inf\{C(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\} = \sup\{L(\mathbf{f}) : \mathbf{f} \in \mathcal{N}\}. \quad (6)$$

If there is a strictly feasible solution for the GCS problem, i.e. if there is a feasible solution such that the inequalities in Eq. (2) all hold strictly, then we can apply the strong duality theorem (see Chapter 6.2 [8]), which is also known as Slater's constraint qualification, to obtain the following theorem.

Theorem 3: If there is a strictly feasible solution to the GCS problem, then the duality gap is zero and there exists a saddle point $(\mathbf{x}, \mathbf{t}, \mathbf{f})$ such that: first, \mathbf{f} is the optimal solution of the D-GCS problem; second, (\mathbf{x}, \mathbf{t}) is the optimal solution of the GCS problem; third, $C(\mathbf{x}) = L^*(\mathbf{x}, \mathbf{t}, \mathbf{f}) = L(\mathbf{f})$.

Theorem 3 guarantees that in the presence of a strictly feasible solution, the GCS problem can be solved by solving the D-GCS problem. For a sizing problem with the constraint on the clock period, there is a strictly feasible solution if the clock period is not tight as shown by the following corollary.

Corollary 2: If there is a feasible solution for the GCS problem formulated from a sizing problem under the clock period T_0 , then if the clock period is substituted by any $T > T_0$, Theorem 3 applies.

There are two misunderstandings when Chen et al. [4] applied the strong duality theorem. First, if transformations are necessary to convert a problem into a convex programming problem, the Lagrangian dual problem should be derived from the transformed problem instead of the original problem. More specifically, since Chen et al. [4] claimed the convexity through geometric programming, the Lagrangian dual problem should be derived from the geometric programming formulation. Here we show that it is not necessary to establish the convexity through geometric programming and thus the Lagrangian dual problem can be formulated as the work [4]. Second, the strong duality theorem requires the existence of a strictly feasible solution. For the GCS problems without a strictly feasible solution, Theorem 3 will not apply and Eq. (6) should be established through other theories. Rockafellar [14] defined a class of convex functions as follows.

Definition 2 (Rockafellar [14]): A convex function $g(\mathbf{x})$ satisfies the regularity condition iff $g(\mathbf{x}) = \mathbf{b}^\top \mathbf{x} + c$ or $g(\mathbf{x}) = h(A\mathbf{x}) + \mathbf{b}^\top \mathbf{x} + c$ for some finite strictly convex function $h(\mathbf{y})$, matrix A , vector \mathbf{b} , and scalar c .

Applying the result of Rockafellar [14], we obtain the following theorem and corollary.

Theorem 4: If the objective function C and the delay functions $d_{i,j}$ all satisfy the regularity condition, then the duality gap is zero.

Corollary 3: If each of the objective function and the delay functions can be written as $p(e^{x_1}, \dots, e^{x_n}) + \mathbf{b}^\top \mathbf{x} + c$ for some posynomial p , vector \mathbf{b} , and scalar c , then the duality gap is zero.

Although Theorem 4 guarantees a zero duality gap without requiring a strictly feasible solution, it does not guarantee any saddle point as in Theorem 3. For a GCS problem without a strictly feasible solution, the D-GCS problem may have no finite solution, i.e. $L(\mathbf{f}') \neq \sup\{L(\mathbf{f}) : \mathbf{f} \in \mathcal{N}\}, \forall \mathbf{f}' \in \mathcal{N}$.

B. Simplifying the Lagrangian Dual Problem

The Lagrangian function L^* can be rewritten as

$$L^*(\mathbf{x}, \mathbf{t}, \mathbf{f}) = C(\mathbf{x}) + \sum_{(i,j) \in E} f_{i,j} d_{i,j}(\mathbf{x}) + \sum_{k \in V} \left(\sum_{(k,j) \in E} f_{k,j} - \sum_{(i,k) \in E} f_{i,k} \right) t_k,$$

Let \mathcal{F} be the set of the multipliers satisfying the flow conservation constraints, i.e.,

$$\mathcal{F} \triangleq \{ \mathbf{f} : \sum_{(i,k) \in E} f_{i,k} = \sum_{(k,j) \in E} f_{k,j}, \forall k \in V \}.$$

If $\sum_{(k,j) \in E} f_{k,j} \neq \sum_{(i,k) \in E} f_{i,k}$ for some $k \in V$, then $\forall M \in R$, t_k can be chosen such that $L^*(\mathbf{x}, \mathbf{t}, \mathbf{f}) < M$. In such cases, $L(\mathbf{f}) = -\infty$. Therefore, the dual problem D-GCS can be simplified by introducing the flow conservation constraints, which is stated as the following *FD-GCS* problem.

Problem 3 (FD-GCS):

$$\begin{aligned} & \text{Maximize} && L(\mathbf{f}) \\ & \text{s.t.} && \mathbf{f} \in \mathcal{F} \cap \mathcal{N}. \end{aligned}$$

The objective function $L(\mathbf{f})$ can be further simplified given $\mathbf{f} \in \mathcal{F}$. For $\mathbf{f} \in \mathcal{N}$, define

$$P_{\mathbf{f}}(\mathbf{x}) \triangleq C(\mathbf{x}) + \sum_{(i,j) \in E} f_{i,j} d_{i,j}(\mathbf{x}),$$

$$Q(\mathbf{f}) \triangleq \inf\{P_{\mathbf{f}}(\mathbf{x}) : \mathbf{x} \in \Omega\}.$$

We formulate the *SD-GCS* problem as follows.

Problem 4 (SD-GCS):

$$\begin{aligned} & \text{Maximize} && Q(\mathbf{f}) \\ & \text{s.t.} && \mathbf{f} \in \mathcal{F} \cap \mathcal{N}. \end{aligned}$$

We call the SD-GCS problem proper if the GCS problem is proper. We call the \mathbf{f} *dual feasible* if $\mathbf{f} \in \mathcal{F} \cap \mathcal{N}$. The FD-GCS problem and the SD-GCS problem are different since their objective functions are different, although both objective function take the same value for all dual feasible \mathbf{f} . Both problems are equivalent to the D-GCS problem as stated in the following theorem.

Theorem 5: The D-GCS problem, the FD-GCS problem, and the SD-GCS problem are equivalent.

Our reasoning in simplifying the Lagrangian dual problem is different from the work [4]. They derived the flow conservation constraints of the multipliers from the KKT conditions of the optimal solutions. However, since the KKT conditions are sufficient but not necessary for an optimal solution, it is possible that there is no feasible solution satisfying such conditions for some sizing problems. We circumvent the difficulty by excluding the \mathbf{f} satisfying $L(\mathbf{f}) = -\infty$.

C. A Trivial Example

One could be deceived to think that the situations mentioned in the previous sections, i.e. there is no saddle point or there is no

feasible solution satisfying the KKT conditions, only happen for some ‘‘corner’’ cases of the sizing problems. However, such situations may happen for extremely trivial examples. Consider the following problem that might be formulated from optimizing a single inverter with the size a .

$$\begin{aligned} & \text{Minimize} && a \\ & \text{s.t.} && t_1 + a \leq t_2, \quad t_2 + \frac{1}{a} \leq t_3, \quad t_3 \leq t_1 + 2, \\ & && \frac{1}{2} \leq a \leq 2. \end{aligned}$$

The problem is a GCS problem of the variable $x = \ln a$. There is only one feasible solution $x = 0$ which is also the optimal solution while there is no strictly feasible solution. Part of the KKT conditions, $(e^x + f_{1,2}e^x - f_{2,3}e^{-x} = 0) \wedge (f_{1,2} = f_{2,3})$, cannot be satisfied by the only feasible solution $x = 0$.

On the other hand, for the dual feasible \mathbf{f} , there should exist $\beta \geq 0$ such that $\beta = f_{1,2} = f_{2,3} = f_{3,1}$. Define

$$q(\beta) \triangleq \inf\{e^x + \beta(e^x + e^{-x} - 2) : -\ln 2 \leq x \leq \ln 2\},$$

$$x_\beta \triangleq \operatorname{argmin}_{-\ln 2 \leq x \leq \ln 2} (e^x + \beta(e^x + e^{-x} - 2)).$$

For the SD-GCS problem, it is simplified as maximizing $q(\beta)$ for $\beta \geq 0$. The $q(\beta)$ and x_β can be computed as

$$q(\beta), x_\beta = \begin{cases} \left(\frac{1+\beta}{2}, -\ln 2\right), & \text{if } 0 \leq \beta < \frac{1}{3} \\ \left(\frac{2}{\sqrt{1+1/\beta+1}}, \ln \sqrt{\frac{\beta}{\beta+1}}\right). & \text{if } \beta \geq \frac{1}{3} \end{cases}$$

So $Q(\mathbf{f}) < 1$ for any finite \mathbf{f} and thus there is no saddle point. Note that for any $\beta \geq 0$, there is no x_β being feasible.

IV. SOLVING THE SIMPLIFIED DUAL PROBLEMS

A. Solving the Lagrangian Subproblem

Solving either the FD-GCS problem or the SD-GCS problem requires solving the Lagrangian subproblem first, i.e. to compute $L(\mathbf{f})$ or equivalently $Q(\mathbf{f})$ for a given dual feasible \mathbf{f} , which is in turn equivalent to minimize $P_{\mathbf{f}}(\mathbf{x})$ for $\mathbf{x} \in \Omega$. Let $\mathbf{x}_{\mathbf{f}} \triangleq \operatorname{argmin}_{\mathbf{x} \in \Omega} P_{\mathbf{f}}(\mathbf{x})$. Note that $\mathbf{x}_{\mathbf{f}}$ is not necessarily unique. Recall that the objective function $C(\mathbf{x})$ and the delay functions $d_{i,j}(\mathbf{x})$ are all twice differentiable convex functions. It is straightforward that $P_{\mathbf{f}}(\mathbf{x})$ is a twice differentiable convex function for $\mathbf{x} \in \Omega$. Many convex programming algorithms (e.g. Chapter 8 [8]) can be applied to compute the optimal solution $\mathbf{x}_{\mathbf{f}}$.

For the Elmore delay model, Chen et al. [4] proposed a greedy algorithm that iteratively sizes each gate and wire segment to solve the LRS/ μ problem, which is a simplified version of the Lagrangian subproblem. Generally speaking, this algorithm is a descent method that uses coordinate axes as the search directions. Chen et al. proved that if the algorithm starts with all the gates and wire segments at their minimum sizes, then it converges to the optimal solution. Since $P_{\mathbf{f}}(\mathbf{x})$ does not increase during the iterations and the algorithm stops only if $P_{\mathbf{f}}(\mathbf{x})$ is the minimum on every coordinate direction, we prove the following stronger result.

Theorem 6: For the Elmore delay model, the algorithm that minimizes $P_{\mathbf{f}}(\mathbf{x})$ for $\mathbf{x} \in \Omega$ based on the descent method using the coordinate axes as the search directions converges to an optimal solution $\mathbf{x}_{\mathbf{f}}$ for any initial solution belonging to Ω .

When the simplified dual problems are solved iteratively by improving the dual feasible \mathbf{f} , intuitively \mathbf{f} would be changed by a small amount from an iteration to another and then the changes in $\mathbf{x}_{\mathbf{f}}$ would be small. Therefore, the algorithm may converge faster if it starts with the previous $\mathbf{x}_{\mathbf{f}}$, while the convergence is guaranteed by Theorem 6.

B. Solving FD-GCS by Subgradient Optimization

Since $L(\mathbf{f})$ is not differentiable in general, the FD-GCS problem was solved by subgradient optimization in [4], [6]. We introduce the algorithm with some modifications as the SubGrad algorithm below.

Starting from any dual feasible \mathbf{f} , the SubGrad algorithm solves the Lagrangian subproblem to compute \mathbf{x}_f and then iteratively improves \mathbf{f} and \mathbf{x}_f until convergence. In each iteration, first the Lagrangian multipliers \mathbf{f} are updated as suggested in the work [4] using the step size ρ_k and the subgradient $t_i + d_{i,j}(\mathbf{x}_f) - t_j$, $\forall (i, j) \in E$, where k is the iteration number and $t_i, \forall i \in V$, are the arrival times. We do not use the heuristic in the work [6] since the relationship between it and the standard subgradient optimization method is not clear. Since after such updating, \mathbf{f} would not necessarily be dual feasible, it is then projected to the closest one that is dual feasible. The projection method is the one presented in the work [6] that distributes incoming flows proportionally to the outgoing edges according to the existing outgoing flows. Finally, the Lagrangian subproblem is solved to compute \mathbf{x}_f for the current \mathbf{f} . To claim convergence, Chen et al. [4] suggested to terminate the iterations when $C(\mathbf{x}_f) - L(\mathbf{f})$ is less than a pre-defined small positive error bound. However, this is only correct if \mathbf{x}_f is feasible since the duality gap is defined between a feasible \mathbf{x} and a dual feasible \mathbf{f} as shown in Eq. (5) and (6). In the SubGrad algorithm, when \mathbf{x}_f is not feasible, the convergence is claimed if the changes in \mathbf{x}_f , \mathbf{f} , and $L(\mathbf{f})$ are marginal.

Although the convergence is claimed for both works [4], [6], it is not clear how to prove such convergence since the convergence of subgradient optimization depends on the choice of the step sizes and the projection method (see Chapter 8.9 [8]). Moreover, the proportional projection method only applies to the GCS problems in which removing one edge from G , e.g. the O to I edge, results in a DAG. Since the SubGrad algorithm is used as a comparison to our algorithm and is not the focus of this paper, we leave further improvements to future researches.

C. Properties of Proper SD-GCS Problems

Assume both \mathbf{f} and \mathbf{f}' belong to \mathcal{N} . Let \mathbf{d} be the vector of all the $d_{i,j}$. Then

$$\begin{aligned} Q(\mathbf{f}') &= \inf\{P_{\mathbf{f}'}(\mathbf{x}) : \mathbf{x} \in \Omega\} \leq P_{\mathbf{f}'}(\mathbf{x}_f) \\ &= P_{\mathbf{f}}(\mathbf{x}_f) + \mathbf{d}(\mathbf{x}_f)^\top (\mathbf{f}' - \mathbf{f}) \\ &= Q(\mathbf{f}) + \mathbf{d}(\mathbf{x}_f)^\top (\mathbf{f}' - \mathbf{f}) \end{aligned}$$

Therefore, the following theorem holds.

Theorem 7: $Q(\mathbf{f})$ is a concave function for $\mathbf{f} \in \mathcal{N}$.

Recall that the SD-GCS problem is proper if the GCS problem is proper. We prove the following lemma which is essential for $Q(\mathbf{f})$ to be differentiable.

Lemma 1: In a proper GCS problem, for any $\mathbf{f} \in \mathcal{N}$, there is a single $\mathbf{x}_f \in \Omega$ satisfying that $Q(\mathbf{f}) = P_{\mathbf{f}}(\mathbf{x}_f)$.

Proof: We prove the lemma by contradiction.

For any twice differentiable function h of \mathbf{x} , let H_h be the Hessian matrix of h . Assume $Q(\mathbf{f}) = P_{\mathbf{f}}(\mathbf{x}') = P_{\mathbf{f}}(\mathbf{x}'')$ for some $\mathbf{x}' \neq \mathbf{x}''$, $\mathbf{x}' \in \Omega$, $\mathbf{x}'' \in \Omega$, and $\mathbf{f} \in \mathcal{N}$. Let $\mathbf{y}^\gamma = (1 - \gamma)\mathbf{x}' + \gamma\mathbf{x}''$. Then $\mathbf{y}^\gamma \in \Omega$, $\forall 0 \leq \gamma \leq 1$. Recall that the function $P_{\mathbf{f}}(\mathbf{x})$ is convex and twice differentiable. We have,

$$Q(\mathbf{f}) = (1 - \gamma)P_{\mathbf{f}}(\mathbf{x}') + \gamma P_{\mathbf{f}}(\mathbf{x}'') \geq P_{\mathbf{f}}(\mathbf{y}^\gamma), \forall 0 \leq \gamma \leq 1.$$

Since $Q(\mathbf{f}) \leq P_{\mathbf{f}}(\mathbf{y}^\gamma)$, we must have $P_{\mathbf{f}}(\mathbf{y}^\gamma) = Q(\mathbf{f})$, $\forall 0 \leq \gamma \leq 1$.

Let $\mathbf{z}^0 = \mathbf{y}^{\frac{1}{2}}$ and $\mathbf{z} = \mathbf{x}'' - \mathbf{x}'$. Then $\mathbf{z}^0 \in \Omega$ and $\mathbf{z} \neq 0$. Therefore,

$$\begin{aligned} \mathbf{z}^\top \nabla P_{\mathbf{f}}(\mathbf{z}^0) &= \lim_{\lambda \rightarrow 0} \frac{P_{\mathbf{f}}(\mathbf{z}^0 + \lambda \mathbf{z}) - P_{\mathbf{f}}(\mathbf{z}^0)}{\lambda} \\ &= \lim_{\lambda \rightarrow 0} \frac{P_{\mathbf{f}}(\mathbf{y}^{\frac{1}{2} + \lambda}) - P_{\mathbf{f}}(\mathbf{y}^{\frac{1}{2}})}{\lambda} = 0, \\ \mathbf{z}^\top H_{P_{\mathbf{f}}}(\mathbf{z}^0) \mathbf{z} &= \lim_{\lambda \rightarrow 0} 2 \frac{P_{\mathbf{f}}(\mathbf{z}^0 + \lambda \mathbf{z}) - P_{\mathbf{f}}(\mathbf{z}^0) - \lambda \mathbf{z}^\top \nabla P_{\mathbf{f}}(\mathbf{z}^0)}{\lambda^2} \quad (7) \\ &= \lim_{\lambda \rightarrow 0} 2 \frac{P_{\mathbf{f}}(\mathbf{y}^{\frac{1}{2} + \lambda}) - P_{\mathbf{f}}(\mathbf{y}^{\frac{1}{2}}) - 0}{\lambda^2} = 0. \end{aligned}$$

On the other hand, it is straightforward that because of the convexity, $H_{d_{i,j}}(\mathbf{x})$, $\forall (i, j) \in E$, are positive semidefinite for any $\mathbf{x} \in \Omega$. So $\mathbf{z}^\top H_{d_{i,j}}(\mathbf{z}^0) \mathbf{z} \geq 0$. Since $H_C(\mathbf{x})$ is positive definite for any $\mathbf{x} \in \Omega$, we have $\mathbf{z}^\top H_C(\mathbf{z}^0) \mathbf{z} > 0$. Therefore,

$$\mathbf{z}^\top H_{P_{\mathbf{f}}}(\mathbf{z}^0) \mathbf{z} = \mathbf{z}^\top H_C(\mathbf{z}^0) \mathbf{z} + \sum_{(i,j) \in E} f_{i,j}(\mathbf{z}^\top H_{d_{i,j}}(\mathbf{z}^0) \mathbf{z}) > 0.$$

This contradicts Eq. (7). Thus the lemma holds. \blacksquare

Base on Lemma 1 and Theorem 6.3.3 [8] we have,

Theorem 8: For a proper SD-GCS problem, the function $Q(\mathbf{f})$ is differentiable for $\mathbf{f} \in \mathcal{N}$ and the gradient is $\mathbf{d}(\mathbf{x}_f)$ where \mathbf{x}_f is the only vector in Ω that minimizes $P_{\mathbf{f}}(\mathbf{x})$ for $\mathbf{x} \in \Omega$.

D. Detecting Infeasible GCS Problems

The GCS problem can be infeasible, e.g. the timing constraints of a circuit cannot be met no matter how the gates are sized. Since the objective function $C(\mathbf{x})$ is continuous and the set Ω is compact, the image $C(\Omega)$ is compact. Therefore, we can assume there is a $U \in R$ such that $C(\mathbf{x}) \leq U$, $\forall \mathbf{x} \in \Omega$. Suppose $(\mathbf{x}', \mathbf{t}')$ is a feasible solution, then for any $\mathbf{f} \in \mathcal{F} \cap \mathcal{N}$, we have

$$P_{\mathbf{f}}(\mathbf{x}') = C(\mathbf{x}') + \sum_{(i,j) \in E} f_{i,j}(t'_i + d_{i,j}(\mathbf{x}') - t'_j) \leq U.$$

On the other hand, if $\forall \mathbf{f} \in \mathcal{F} \cap \mathcal{N}$, there is a $\mathbf{x}' \in \Omega$ such that $P_{\mathbf{f}}(\mathbf{x}') \leq U$, then $Q(\mathbf{f}) = \inf\{P_{\mathbf{f}}(\mathbf{x}) : \mathbf{x} \in \Omega\} \leq U$. Thus $Q(\mathbf{f})$ is bounded for $\mathbf{f} \in \mathcal{F} \cap \mathcal{N}$. We have,

Theorem 9: Suppose $C(\mathbf{x})$ is upper-bounded by $U \in R$ for $\mathbf{x} \in \Omega$. If the duality gap is zero, then the GCS problem is feasible iff $Q(\mathbf{f}) \leq U$, $\forall \mathbf{f} \in \mathcal{F} \cap \mathcal{N}$.

Theorem 9 provides a method to check whether the GCS problem is feasible when optimizing $Q(\mathbf{f})$. For a sizing problem where $Q(\mathbf{f}) > U$ for some dual feasible \mathbf{f} , by investigating the non-zero multipliers, one can identify the troublesome part of the circuit that makes it infeasible. Other optimization techniques could be performed to that part to make the circuit feasible and we leave this as a future research topic.

E. Solving Proper SD-GCS Problems via Method of Feasible Directions

Recall that the objective function $Q(\mathbf{f})$ of a proper SD-GCS problem has the gradient $\mathbf{d}(\mathbf{x}_f)$ according to Theorem 8. We apply the method of feasible directions (see Chapter 10 [8]) to solve the proper SD-GCS problem.

For any dual feasible \mathbf{f} , the vector $\Delta \mathbf{f}$ is an improving feasible direction iff there exists $\lambda > 0$ such that $Q(\mathbf{f}) < Q(\mathbf{f} + \lambda \Delta \mathbf{f})$ and $\mathbf{f} + \lambda \Delta \mathbf{f} \in \mathcal{N} \cap \mathcal{F}$. An improving feasible direction can be found by solving the following direction finding (DF) problem.

Problem 5 (DF):

$$\begin{aligned} &\text{Maximize} && \Delta \mathbf{f}^\top \mathbf{d}(\mathbf{x}_f) \\ &\text{s.t.} && \mathbf{f} + \Delta \mathbf{f} \in \mathcal{F} \cap \mathcal{N}, \\ &&& -u \leq \Delta f_{i,j} \leq u, \forall (i, j) \in E. \end{aligned}$$

Here the improving feasible direction $\Delta \mathbf{f}$ are the decision variables, \mathbf{f} should be dual feasible, and u is a positive constant.

The intuition is to maximize the improvement in a dual feasible neighborhood assuming the function Q is linear. The DF problem is a min-cost network flow problem. Since \mathbf{f} is dual feasible, $\Delta \mathbf{f} = \mathbf{0}$ is always a feasible solution of the DF problem. The following theorem relates the optimal solution to an improving feasible direction.

Theorem 10: Suppose $\Delta \mathbf{f}$ is the optimal solution of the DF problem. If $\Delta \mathbf{f}^\top \mathbf{d}(\mathbf{x}_f) = 0$, then \mathbf{f} is the optimal solution of the SD-GCS problem and there exists a vector \mathbf{t} such that $(\mathbf{x}_f, \mathbf{t})$ is the optimal solution of the GCS problem; otherwise $\Delta \mathbf{f}$ is an improving feasible direction.

ALGORITHM DualFD	
Inputs	The GCS problem and N .
Outputs	Optimal \mathbf{f} and \mathbf{x}_f .
1	$\mathbf{f} \leftarrow \mathbf{0}$. Compute \mathbf{x}_f .
2	For $i = 1$ to N :
3	Solve the DF problem for $\Delta \mathbf{f}$.
4	Claim optimality if $\mathbf{d}(\mathbf{x}_f)^\top \Delta \mathbf{f}$ is small enough.
5	Compute λ_{\max} .
6	Perform a line search on $Q(\mathbf{f} + \lambda \Delta \mathbf{f})$ with $0 \leq \lambda \leq \lambda_{\max}$ for an increase in Q .
7	Update \mathbf{f} . Compute \mathbf{x}_f .
8	Claim optimality if the changes in \mathbf{f} , \mathbf{x}_f , and $Q(\mathbf{f})$ are marginal.

Fig. 4. The DualFD Algorithm.

We design the DualFD algorithm as shown in Figure 4 to solve the proper SD-GCS problem. It starts with the dual feasible $\mathbf{f} = \mathbf{0}$ and iteratively improves $Q(\mathbf{f})$ by finding an improving feasible direction and performing a line search. On line 1 and 7, the vector \mathbf{x}_f is computed by solving the Lagrangian subproblem. On line 5, the λ_{\max} is computed as the maximum λ such that $\mathbf{f} + \lambda \Delta \mathbf{f}$ remains dual feasible. It is straightforward that $\mathbf{f} + \lambda \Delta \mathbf{f} \in \mathcal{F}$. Therefore, $\lambda_{\max} = \min_{\Delta f_{i,j} < 0} -f_{i,j} / \Delta f_{i,j}$, or $\lambda_{\max} = +\infty$ if $\Delta f_{i,j} \geq 0$, $\forall (i, j) \in E$. On line 6, a number of the Lagrangian subproblems should be solved to compute $\mathbf{x}_{f+\lambda \Delta \mathbf{f}}$ and then $Q(\mathbf{f} + \lambda \Delta \mathbf{f})$. Since determining the exact λ that maximizes $Q(\mathbf{f} + \lambda \Delta \mathbf{f})$ would be time consuming, we perform an inexact line search by Armijo's Rule (see Chapter 8.3 [8]) with two parameters ϵ and α , whose typical values are $\epsilon = 0.2$ and $\alpha = 2$. The feasibility of the GCS problem is checked every time when the objective function Q is computed according to Theorem 9. The algorithm terminates when a pre-defined rounds N of iterations are performed, or the optimal solution is found according to Theorem 10 on line 4, or the changes in \mathbf{f} , \mathbf{x}_f , and $Q(\mathbf{f})$ are marginal on line 8.

Note that it is guaranteed in the DualFD algorithm that $Q(\mathbf{f})$ will increase strictly each iteration, which is not guaranteed by the SubGrad algorithm.

V. EXPERIMENTS

A. Experimental Setup

We use the sequential circuits from the ISCAS89 benchmarks as our test cases. The Elmore delay model is used for gate delays. An utility program reads the circuits and generates the GCS problems to be solved. The gate sizing problems that minimize the total gate sizes under a given clock period constraint without clock skew optimization are experimented. The arrival times at the primary input ports and the clock skews of the FFs are 0. The required arrival times at the primary output ports are equal to the clock period. Transistor sizing, wire sizing, and wire capacitances are not currently integrated into the utility program and they can be added straightforwardly if necessary.

TABLE I
STATISTICS OF THE BENCHMARKS.

name	# vertices	# edges	# vars
s27	40	51	13
s208	291	374	104
s298	396	540	133
s344	470	607	175
s349	475	615	176
s382	511	685	179
s386	527	756	165
s420	585	748	212
s444	580	777	202
s510	668	902	217
s526	685	963	214
s641	993	1195	398
s713	1059	1298	412
s820	1076	1655	294
s832	1086	1680	292
s838	1161	1484	422
s953	1214	1670	424
s1196	1590	2259	547
s1238	1601	2343	526
s1423	1988	2573	731
s1488	2062	2868	659
s1494	2062	2880	653
s5378	7386	9046	2958
s9234	14028	16681	5808
s13207	20456	24473	8589
s15850	24564	29158	10306
s35932	47827	63569	17793
s38417	57509	69099	23815
s38584	54901	70133	20679

As suggested in Section II-A, we associate different delay functions $d_{i,j}$ with each timing arc from an input port i to an output port j of a given gate. For each timing arc (i, j) , a weight $w_{i,j}$ is introduced. The driving resistance associated with each timing arc (i, j) in each gate is $R^{nom} w_{i,j} / x$ and the load capacitance is $C^{nom} w_{i,j} x$, where R^{nom} and C^{nom} are the characteristic parameters of the gate and x is the size of the gate. We emphasize our paper focuses on algorithmic developments for the sizing problem. Such experimental setup would suffice our purpose of algorithm evaluation.

The statistics of the GCS problems generated by the utility program from the ISCAS89 benchmarks are shown in Table I. We report the size of the graph G in the columns “# vertices” and “# edges”. The numbers of the sizable gates are shown in the column “# vars”. For each circuit, the clock period constraint T_0 is computed as half of the minimum clock period that the circuit can operate assuming that each gate takes its average size. Note that we do not know whether there is a feasible solution for each GCS problem at this point.

B. Experimental Results

We implement the DualFD and the SubGrad algorithms in C++ where the DF problem is solved by the min-cost network flow solver CS2 version 4.3 [15]. The code is compiled by GCC version 3.4 and runs on a Linux workstation with two 927MHz Pentium III processors and 512M memory. We prefer to implement the SubGrad algorithm without the pre-processing heuristics in the work [6] because we want to compare the algorithms that solve the simplified dual problems while the pre-processing heuristics apply to both the DualFD and the SubGrad algorithms.

TABLE II
RESULTS COMPARISON BETWEEN THE DUALFD AND THE SUBGRAD ALGORITHMS.

name	DualFD						SubGrad				
	area	dual	T/T ₀	N	L	t(s)	area	dual	T/T ₀	L	t(s)
s27	90.8	90.9	1.000	15	47	0.01	68.3	-86.2	1.226	5001	0.89
s208 ⁺	854.2	1073.7	1.012	200	1067	1.97	2210.9	-34062.0	1.461	5001	9.03
s298 ⁺	242.5	242.5	1.000	35	185	0.22	316.2	-1512.9	1.159	5001	4.93
s344 ⁺	445.2	445.8	1.001	69	382	0.93	517.5	-3657.0	1.447	5001	7.64
s349 ⁺	446.8	447.5	1.001	53	289	0.71	518.5	-3642.2	1.447	5001	7.78
s382 ⁺	315.7	315.7	1.000	29	178	0.24	317.6	-396.9	1.107	5001	4.79
s386 ⁺	1222.1	1357.5	1.021	200	679	7.35	6940.1	-152K	1.445	5001	23.28
s420*	1471.0	24461.9	2.589	1	9	0.03	4904.4	-97197.0	2.500	5001	22.22
s444 ⁺	360.7	360.6	1.000	167	926	1.69	364.2	-593.5	1.190	5001	5.88
s510 ⁺	479.9	479.8	1.000	134	515	2.13	2374.9	-24810.9	1.303	5001	27.71
s526	383.5	383.6	1.000	43	208	0.51	375.2	51.4	1.490	5001	7.09
s641 ⁺	887.7	888.9	1.004	29	122	0.79	1069.5	-17839.8	1.313	5001	18.59
s713 ⁺	1400.1	1409.3	1.002	48	259	2.39	1458.3	-38666.9	1.421	5001	26.73
s820	1088.4	1093.5	1.002	200	737	8.18	496.9	495.9	2.000	318	0.81
s832	1060.5	1073.9	1.005	200	713	8.28	493.5	493.4	2.000	43	0.11
s838*	2916.0	80458.3	4.769	1	9	0.07	10188.8	-257K	4.769	5001	51.51
s953 ⁺	775.7	775.5	1.001	200	848	5.92	3449.3	-79502.6	1.238	5001	55.15
s1196 ⁺	1088.8	1088.2	1.001	200	841	10.90	1642.5	-14107.7	1.242	5001	54.88
s1238	1079.7	1079.9	1.001	131	540	7.80	973.8	-1823.4	1.498	5001	27.19
s1423 ⁺	1668.2	1670.0	1.002	18	81	1.53	3345.8	-251K	1.299	5001	120.76
s1488	2055.1	2106.7	1.011	200	710	28.22	1149.7	557.3	1.991	5001	31.28
s1494	2160.9	2317.8	1.020	200	717	28.33	1139.9	529.3	1.992	5001	32.02
s5378 ⁺	5855.7	6083.3	1.051	200	652	91.49	9396.1	-52345.1	1.441	5001	308.39
s9234	12935.3	15507.7	1.080	200	835	236.49	11516.7	-46086.7	1.491	5001	384.86
s13207 ⁺	14608.3	14608.3	1.002	101	456	111.91	15641.7	-121K	1.242	5001	432.81
s15850 ⁺	17765.6	17765.9	1.003	157	602	229.25	20627.7	-287K	1.499	3946	600.10
s35932	33522.1	44343.6	1.814	65	193	304.61	80649.6	-1038K	1.394	668	600.34
s38417 ⁺	42175.9	44551.4	1.194	67	274	301.48	49125.6	-363K	1.483	1260	600.35
s38584 ⁺	34973.0	34973.0	1.002	45	225	149.87	35015.7	-23232.2	1.380	3144	600.19

* The DualFD algorithm determines that the benchmark is infeasible.

⁺ The result of the DualFD algorithm dominates that of the SubGrad algorithm.

The experimental results are compared in Table II. For the DualFD algorithm, we run each benchmark with at most 200 rounds of iterations and at most 300 seconds. The results are reported under the columns “DualFD”. The total gate sizes, i.e. the values of the objective function, are shown in the column “area”. The values of the function $Q(\mathbf{f})$ are shown in the column “dual”. We compute the minimum clock period T that each circuit can operate and the ratios to the clock period constraint T_0 are shown in the column “ T/T_0 ”. The total number of iterations, the number of the Lagrangian subproblems being solved, and the total running time in seconds are shown in the columns “N”, “L”, and “t” respectively. Note that for the benchmarks “s420” and “s838” marked by * in the table, the DualFD algorithm determines that there is no feasible solution in the first round since $Q(\mathbf{f})$ is larger than U , which is computed as the total gate size assuming that each gate takes its maximum size. The convergence sequence of the benchmark “s38584” is plotted in Figure 5.

For the SubGrad algorithm, we use a step size of the form c/k where c is a constant and k is the current iteration number. For each benchmark, we set $c = 10^{-i}$ for $i = 0, 1, \dots, 6$ and run the algorithm 7 times for each c with at most 5000 rounds of iterations and at most 600 seconds. Two set of the initial dual feasible \mathbf{f} are tried: one is $\mathbf{f} = \mathbf{0}$ and the other is the one obtained by running the DualFD algorithm by 1 round. The results of the latter setting are inferior to those of the former and thus are not reported here. For

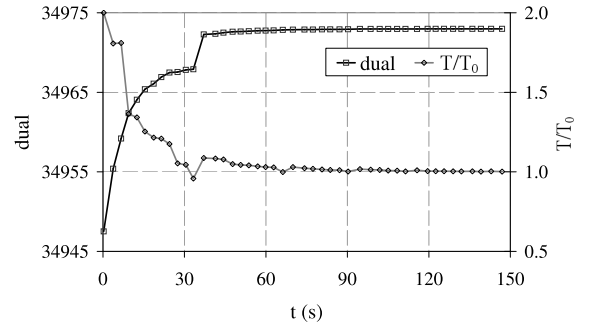


Fig. 5. Convergence sequence of s38584 for DualFD.

each of the 7 results, we compute the minimum clock period T that the circuit can operate. If there is a result with $T/T_0 < 1.5$, then we report the result with $T/T_0 < 1.5$ and the maximal value of the function $L(\mathbf{f})$ under the columns “SubGrad”; otherwise, we report the result with the minimal T . The values of the function $L(\mathbf{f})$ are shown in the column “dual”. All the other columns have the similar meanings as those of the DualFD algorithm. We do not report the total number of iterations since it is equal to the number of the Lagrangian subproblems being solved minus 1 for the SubGrad algorithm.

It can be seen from the table that the DualFD algorithm consistently generates results with better quality, requires a smaller number of the Lagrangian subproblems to be solved, and spends less running time compared to the SubGrad algorithm. More specifically, the results can be compared based on the “area” and the “t” columns. In addition,

although the “dual” columns represent different functions, they can be compared directly since both functions take the same value for the dual feasible \mathbf{f} and they are the objective functions to be maximized in both algorithms. The “ T/T_0 ” columns can be compared since they measure the infeasibility of the results, i.e. the distance of the \mathbf{x} in the result to a feasible one. Define that a result dominates another if it has a smaller value in the columns “area”, “ T/T_0 ”, and “ t ” and a larger value in the column “dual”. There are 18 benchmarks, as marked by $+$ in the table, out of the 27 benchmarks (the other two have no feasible solution as shown by the DualFD algorithm) where the results of the DualFD algorithm dominate those of the SubGrad algorithm and the results of the SubGrad algorithm never dominate those of the DualFD algorithm.

We collect feasible solutions when running the DualFD and the SubGrad algorithms and report the best ones, i.e. the ones with the minimal total gate size, in Table III. Only the best feasible solution of the benchmark “s27”, as marked by $*$ in the table, is found by the SubGrad algorithm. All the other best feasible solutions are found by the DualFD algorithm. The total gate sizes are shown in the column “ area_{fea} ”. We compute the minimum clock period T_{fea} that a circuit can operate and the ratios to the clock period constraint are shown in the column “ T_{fea}/T_0 ”. The benchmarks with - in the columns are the ones without a feasible solution being found. When there is a feasible solution, we can obtain an upper-bound on the gap between the results generated by the DualFD algorithm and the optimal solutions. A zero gap proves that the result is an optimal solution. We compute the gaps relatively as $(\text{area}_{\text{fea}} - \text{dual})/\text{area}_{\text{fea}}$ where “dual” is the value in the column “dual” in Table II for the DualFD algorithm and report the values in the column “gap”. There are 14 benchmarks where the results generated by the DualFD algorithm are close (less than 3%) to the optimal solutions.

The detailed running time information of the DualFD algorithm are also reported in Table III. The time spent by solving the DF problems and the Lagrangian subproblems in seconds are shown in the columns “ t_{DF} ” and “ t_{LRS} ” respectively. Other parts of the DualFD algorithm spend only a small portion of the total running time and are thus excluded from being reported. The time spent in solving the DF problems gradually dominates the time spent in solving the Lagrangian subproblems for the Elmore delay model as the circuit sizes increase. If a more sophisticated but accurate delay model is used in the future, the time spent in solving the Lagrangian subproblems would increase but the time spent in solving the Direction Finding problems would remain most likely unchanged since the complexity of the DF problem only depends on the size of the graph G . Thus, considering the difference in the number of the Lagrangian subproblems that should be solved in both the DualFD and the SubGrad algorithms, we expect that the running time ratio of the DualFD algorithm to the SubGrad algorithm will become more significant if a more sophisticated delay model is used.

VI. CONCLUSIONS

In this work, we revisited the approach to solving the gate sizing problem via Lagrangian relaxation. We formulated a generalized sizing problem GCS, identified a class of proper GCS problems, and presented a method to handle simultaneous sizing and clock skew optimization as a proper GCS problem. We established conditions for the objective function of the simplified dual problem to be differentiable and for the GCS problem to be feasible. We designed an algorithm based on the method of feasible directions to solve the proper GCS problem. The efficiency and effectiveness of our new approach was confirmed by the experimental results.

TABLE III
ADDITIONAL RESULTS FOR DUALFD.

name	area_{fea}	T_{fea}/T_0	gap	$t_{\text{DF}}(\text{s})$	$t_{\text{LRS}}(\text{s})$
s27	137.6*	0.997*	0.340	0.0	0.0
s208	-	-	-	0.6	1.4
s298	242.5	1.000	0.000	0.1	0.1
s344	448.4	1.000	0.006	0.4	0.6
s349	454.0	0.999	0.014	0.3	0.4
s382	315.7	1.000	0.000	0.1	0.1
s386	-	-	-	1.6	5.7
s420	-	-	-	0.0	0.0
s444	360.7	1.000	0.000	0.9	0.7
s510	480.9	1.000	0.002	1.1	1.0
s526	383.7	1.000	0.000	0.3	0.2
s641	959.1	0.983	0.073	0.4	0.4
s713	1449.2	0.995	0.027	0.7	1.6
s820	-	-	-	3.1	4.9
s832	-	-	-	3.1	5.1
s838	-	-	-	0.0	0.1
s953	776.5	1.000	0.001	3.2	2.5
s1196	1105.0	1.000	0.015	5.4	5.3
s1238	1094.8	0.999	0.014	3.4	4.3
s1423	1685.5	0.998	0.009	0.5	1.0
s1488	-	-	-	8.6	19.4
s1494	-	-	-	7.0	21.2
s5378	-	-	-	69.7	21.1
s9234	-	-	-	160.6	74.5
s13207	14614.9	1.000	0.000	89.4	21.5
s15850	-	-	-	190.4	36.9
s35932	-	-	-	250.1	52.8
s38417	-	-	-	237.8	61.7
s38584	34975.1	0.999	0.000	119.7	28.7

* Found when applying the SubGrad algorithm.

REFERENCES

- [1] J. Fishburn and A. Dunlop, “TILOS: A posynomial programming approach to transistor sizing,” in *ICCAD*, 1985, pp. 326–328.
- [2] S. S. Sapatnekar, V. B. Rao, P. M. Vaidya, and S. M. Kang, “An exact solution to the transistor sizing problem for CMOS circuits using convex optimization,” *IEEE TCAD*, vol. 12, pp. 1621–1634, Nov. 1993.
- [3] V. Chuang, S. S. Sapatnekar, and I. N. Hajj, “Timing and area optimization for standard-cell VLSI circuit design,” *IEEE TCAD*, vol. 14, no. 3, pp. 308–320, Mar. 1995.
- [4] C.-P. Chen, C. C. N. Chu, and D. F. Wong, “Fast and exact simultaneous gate and wire sizing by Lagrangian relaxation,” *IEEE TCAD*, vol. 18, no. 7, pp. 1014–1025, July 1999.
- [5] V. Sundararajan, S. S. Sapatnekar, and K. K. Parhi, “Fast and exact transistor sizing based on iterative relaxation,” *IEEE TCAD*, vol. 21, no. 5, pp. 568–581, May 2002.
- [6] H. Tennakoon and C. Sechen, “Gate sizing using Lagrangian relaxation combined with a fast gradient-based pre-processing step,” in *ICCAD*, 2002, pp. 395–402.
- [7] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [8] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming*, 3rd ed. Wiley-Interscience, 2006.
- [9] W. C. Elmore, “The transient response of damped linear networks with particular regard to wide-band amplifiers,” *Journal of Applied Physics*, vol. 19, no. 1, pp. 55–63, Jan. 1948.
- [10] K. Kasamsetty, M. Ketkar, and S. S. Sapatnekar, “A new class of convex functions for delay modeling and their application to the transistor sizing problem,” *IEEE TCAD*, vol. 19, no. 7, pp. 779–788, July 2000.
- [11] Faraday Technology Corporation, “UMC 90-nano Libraries,” <http://freelibrary.faraday-tech.com/>, Nov. 2005.
- [12] H. Tennakoon and C. Sechen, “Efficient and accurate gate sizing with piecewise convex delay models,” in *DAC*, 2005, pp. 807–812.
- [13] C. Lin and H. Zhou, “Clock skew scheduling with delay padding for prescribed skew domains,” in *ASP-DAC*, 2007.
- [14] R. T. Rockafellar, “Ordinary convex programs without a duality gap,” *Journal of Optimization Theory and Applications*, vol. 7, no. 3, pp. 143–148, Mar. 1971.
- [15] Andrew Goldberg’s network optimization library, <http://www.avglab.com/andrew/soft.html>.