Freeze: Engineering a Fast Repeater Insertion Solver for Power Minimization Using the Ellipsoid Method

Yuantao Peng Xun Liu
Department of Electrical and Computer Engineering
North Carolina State University, Raleigh, NC 27695
{ypeng, xunliu}@ncsu.edu

ABSTRACT

This paper presents a novel repeater insertion algorithm for the power minimization of realistic interconnect trees under given timing budgets. Our algorithm judiciously combines a local optimizer based on the dynamic programming technique and a global search engine using the *ellipsoid method*. As a result, our approach is capable of producing high-quality solutions at a very fast speed. Furthermore, our scheme is robust and does not need any manual tuning of the iteration-control parameters.

We have developed a repeater insertion tool, called FREEZE, using the proposed algorithm and applied it to various interconnect trees with different timing targets. Experimental results demonstrate the high effectiveness of our approach. In comparison with the state-of-the-art low-power repeater insertion schemes, FREEZE requires 5.8 times fewer iterations on the average, achieving up to 27 times speedup with even better power savings. When compared with a dynamic programming based scheme, which guarantees the optimal solution, our tool delivers up to 50 times speedup with 0.9% power increase on the average.

Categories and Subject Descriptors

J.6 [**Computer-aided design (CAD)**]: Generic CADD

General Terms

Algorithms

Keywords

Interconnect, Repeater Insertion, Low Power

1. INTRODUCTION

Repeater insertion is a widely used technique to reduce the delay of long interconnects. Future VLSI designs are expected to consist of millions of repeaters that could affect the system speed and power significantly [13, 23, 24]. Consequently, repeater insertion methodologies are in urgent need for the implementation of high-performance and low-power systems. Although fast repeater insertion algorithms for interconnect delay minimization have been proposed, efficient and effective algorithms for repeater power minimization are still elusive.

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

This paper presents a fast repeater insertion algorithm for power minimization of global interconnects using the ellipsoid method. Our algorithm, called FREEZE, is based on a mathematical framework of Lagrangian relaxation [15]. Specifically, FREEZE performs a sequence of search operations in a hyper space defined by a set of Lagrangian multipliers. For each hyper location traversed, a local optimization is performed using a dynamic programming (DP) based scheme to derive a repeater insertion solution. This solution is then used to determine the next search location. The iterative procedure continues until it converges. After the convergence, the solution computed at the last hyper location is returned as the final result. The major contribution of our paper is two-fold. First, we adopted the ellipsoid method to control the parameter update in the outer loop of the iteration, improving the convergence rate significantly. Second, we developed a new DP algorithm to derive the repeater insertion solution within the iteration loops.

Our scheme has several advantages. First, it is highly practical and applicable to realistic interconnects routed in actual design scenarios. In particular, our algorithm models interconnects as tree structures comprising wire segments with fixed lengths and distinct RC characteristics, as derived from a routing procedure. Furthermore, it can handle *forbidden zones*, i.e., parts of interconnects through macrocells in which no repeater can be placed. Second, our scheme achieves a superior trade-off between the runtime and solution quality. With the adoption of the ellipsoid method, the iteration in FREEZE converges very fast with little power degradation. Third, our scheme is highly robust and stable. Its convergence rate is insensitive to the initial solution and, therefore, the time consuming initialization procedure often needed in other repeater insertion algorithms is eliminated. Moreover, unlike several previously proposed schemes, our scheme does not require manual tuning of the iteration-control coefficients.

We have applied FREEZE to a suite of interconnect designs to demonstrate its effectiveness. Our interconnect trees are routed on multiple metal-layers, in different topologies, and with various timing budgets. Experimental results show that FREEZE has achieved up to 50 times speedup in comparison with the optimum DP-based approach with only 0.9% power increase on the average. Compared with the state-of-the-art low-power repeater insertion schemes that target the balance of runtime and power savings, our scheme runs 9.2 times faster on the average with better solution quality.

The rest of our paper contains 7 sections. Section 2 reviews previous repeater insertion research. Section 3 describes our circuit model. The low-power repeater insertion problem is formulated in Section 4. Section 5 presents a general framework based on Lagrangian relaxation to derive the low-power repeater insertion solution. In Section 6, our algorithm is proposed. Section 7 presents our experimental procedure. Section 8 summarizes our paper.

2. PREVIOUS RESEARCH

Repeater insertion has been investigated extensively in the literature [5, 21]. Several circuit models have been proposed to compute the delay or power dissipation of repeaters such as the switch-level RC model [9], the generalized model considering slew rate [14], and the moment matching model [3]. Various design objectives are used such as delay minimization [4, 12, 19], power minimization [6, 10, 13, 16, 18], and cross-coupling noise reduction [1, 7].

Repeater insertion algorithms can be classified as analytical approaches [8] and DP-based approaches [11]. In analytical approaches, the optimization objectives are described using functions of repeater width and location. The optimal repeater insertion solutions can be derived by setting the derivatives of these functions to zero and solving the ensuing equations. When designing interconnect trees routed in multiple metal layers with forbidden zones, analytical schemes often generate very complex and intractable non-linear equations. Moreover, analytical approaches cannot handle the discreteness of repeater counts and widths. Consequently, they are usually applied to interconnects of simple topologies and with uniform RC characteristics. DP-based techniques can handle realistic interconnects and therefore do not suffer from the limitation of analytical schemes. Specifically, in DP-based schemes, the possible widths and locations of the repeaters are discrete and finite, and the algorithms choose the best solution out of all the possibilities. The drawback of DP based approaches is the long runtimes, especially when power minimization is the design objective. The joint application of the DP-based approach and non-linear analytical solver has been proposed recently for both fast and high-quality repeater insertion solutions [15, 17].

3. CIRCUIT MODEL

Figure 1 illustrates our circuit models. Repeaters are represented using the switch-level RC model, where w is the repeater width, and R_s, C_o and C_p are output resistance, input capacitance, and output capacitance per unit repeater width, respectively. Each uniform interconnect segment is described using the lumped-RC π model, where l is the interconnect length, and c and r are the capacitance and resistance per unit length, respectively.

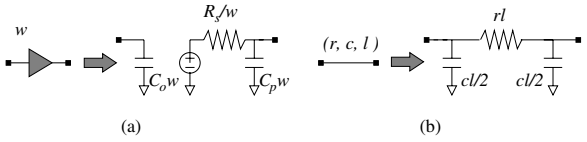


Figure 1: Circuit model for (a) repeaters and (b) interconnects.

To calculate the repeater delay, our scheme uses the widely adopted Elmore delay model as most other repeater insertion solvers so that a fair runtime comparison can be made. The total power of repeaters can be approximated as the sum of the dynamic power and leakage power. Since all nodes on a single interconnect have the same signal activity and repeater transistors often have the same channel length, the dynamic power is a linear function of the total repeater width. Furthermore, the leakage power is also linear with respect to repeater width. Consequently, for the rest of this paper, we replace the power minimization by the minimization of the total repeater width, as proposed in [10].

4. PROBLEM FORMULATION

Figure 2 shows the structure of a multi-layer interconnect tree. The driver s is a buffer *within* the circuit block that sends data onto the global interconnect. The sinks t_i, $i = 1,2,\ldots,n$, are buffers *within* the receiver blocks and can be modeled as capacitors. The

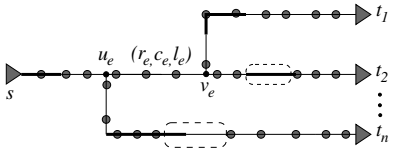


Figure 2: Non-uniform interconnect tree.

sizes of the driver and sinks are given and cannot be changed, therefore maintaining the delays of the circuit paths before the driver and after the sinks. The interconnect is made of several segments with distinct RC characteristics connected in a tree topology. The connectivity among the segments is derived by the corresponding routing procedure. Each segment e is assumed to be routed on a single layer with a uniform width. The beginning and end points of e are denoted as u_e and v_e. The length, resistance and capacitance per unit length for segment e are denoted as l_e, r_e and c_e, respectively. In a realistic routing scenario, an interconnect tree may go through some macro-blocks, in which no repeater can be placed. The portions of the interconnect within the macro-blocks, called forbidden zones, are marked by the dotted boxes. Due to the discreteness of the layout grid, feasible repeater locations are limited to certain discrete places, which are shown as the gray dots in Figure 2.

The *primal problem* of low-power repeater insertion for realistic interconnects is described as follows:

Problem LPRI: *Let $T = (E,V)$ be an interconnect tree structure with n sinks, where E is the set of interconnect segments and V is the set of vertices at which two or more segments join. The source of the tree is denoted as s and sinks of the tree are denoted as t_i, $i = 1,2,\ldots,n$. The buffer sizes of both the source driver and sinks are given and fixed. Furthermore, let* $\mathbf{W}$ *be the set of possible widths defined by a repeater library, and let B_e be the set of candidate repeater locations along interconnect segment e. Given the required arrival time (RAT) q_i for each sink i, find the repeater width $w_{e,j} \in \{\mathbf{W} \cup 0\}$, $j = 1,\ldots,|B_e|$ for each location $b_{e,j}$ in any B_e to*

$$\begin{aligned} &\textit{Minimize:} && \sum_{e \in E} W_e && (1)\\ &\textit{Subject to:} && \forall k \in V, \exists a_k \geq 0\ , \\ & && a_{u_e} + d_e \leq a_{v_e}\ , \\ & && \forall v_e \in \{t_i | i = 1,2,\ldots,n\},\ a_{v_e} = q_i\ , \end{aligned}$$

where $W_e = \sum_{j=1,2,\ldots,|B_e|} w_{e,j}$ denotes total widths of repeaters on e and $w_{e,j} = 0$ indicates that no repeater is placed at $b_{e,j}$. The parameter d_e represents the signal delay from u_e to v_e, which is a function of upstream resistance before u_e, downstream capacitance after v_e, and repeater insertion solution $w_{e,j}$ on e.

Intuitively, a_k represents the RAT at vertex k. The inequality constraints ensure the validity of the RATs. The equality constraints ensure that all timing targets are satisfied.

5. LOW-POWER REPEATER INSERTION USING LAGRANGIAN RELAXATION

Solving the primal problem in Section 4 is very challenging due to the existence of a large number of inequality constraints. Furthermore, the segment delay d_e cannot be described using simple analytical expressions. Moreover, additional unknowns a_k need to be calculated besides repeater insertion variables $w_{e,j}$. Therefore, the primal problem is often converted to an equivalent problem called the dual problem that has less unknowns and constraints [22]. Specifically, using Lagrangian relaxation, a set of non-negative values called Lagrangian multipliers λ_e are introduced for each

edge e. The Lagrangian relaxation function is then written as:

$$L(\vec{W},\vec{a},\vec{\lambda}) = \sum_{e\in E} W_e + \sum_{e\in E} \lambda_e(a_{u_e} + d_e - a_{v_e}) \quad (2)$$

where $\vec{\lambda} = (\lambda_1,\lambda_2,\ldots,\lambda_{|E|})$ is the Lagrangian multiplier vector, $\vec{W} = (W_1,W_2,\ldots,W_{|E|})$ is the set of total repeater widths for all edges, and $\vec{a} = (a_1,a_2,\ldots,a_{|V|})$ are the RATs of all vertices.

In the optimal solution, $\forall k \in \{1,\ldots,|V|\}$, $\partial L(\vec{W},\vec{a},\vec{\lambda})/\partial a_k = 0$. Therefore, for all edges that do not connect to any sink $t_i, i = 1,\ldots,n$,

$$\lambda_e = \sum_{e'\in des(e)} \lambda_{e'} , \quad (3)$$

where $des(e)$ denotes the set of descendant edges of e.

From Equation (3), λ_e is an independent variable only for e that satisfies $v_e \in \{t_i, i = 1,\ldots,n\}$. Consequently, the set of Lagrange multipliers λ_e is completely determined by those assigned to the edges connected to the sinks. Furthermore, by combining Equations (2) and (3), the parameters a_k can be eliminated as follows:

$$\begin{aligned} L(\vec{W},\vec{a},\vec{\lambda}) &= \sum_{e\in E} W_e + \sum_{e\in E} \lambda_e(a_{u_e} + d_e - a_{v_e}) \\ &= \sum_{e\in E} W_e + \sum_{e\in E} \lambda_e d_e + \sum_{e\in E} \lambda_e(a_{u_e} - a_{v_e}) \\ &= \sum_{e\in E} W_e + \sum_{e\in E} \lambda_e d_e + \sum_{e\in E} (\lambda_e a_{u_e} - \sum_{e'\in des(e)} \lambda_{e'} a_{v_e}) \\ &= \sum_{e\in E} (W_e + \lambda_e d_e) - \sum_{\forall v_e = t_i} \lambda_e q_i . \end{aligned} \quad (4)$$

From Equation (4), the Lagrange function $L(\vec{W},\vec{a},\vec{\lambda})$ does not depend on the RATs. Therefore, given $\vec{\lambda}$, $L(\vec{W},\vec{a},\vec{\lambda})$ is just a function of $W_e, e \in E$, and can be denoted as $L_{\vec{\lambda}}(\vec{W})$.

The equivalent dual of the primal problem of repeater insertion for low power is described as follows:

Dual problem: *Given a design specification, including an interconnect tree $T = (E,V)$ of n sinks, possible repeater locations $B = \{B_e | e \in E\}$, a repeater library* $\mathbf{W}$*, and timing constraints at all sinks* $\mathbf{q}$*, for every location $b_{e,j}$ in each B_e, find a repeater width $w_{e,j} \in \{\mathbf{W} \cup 0\}$ to*

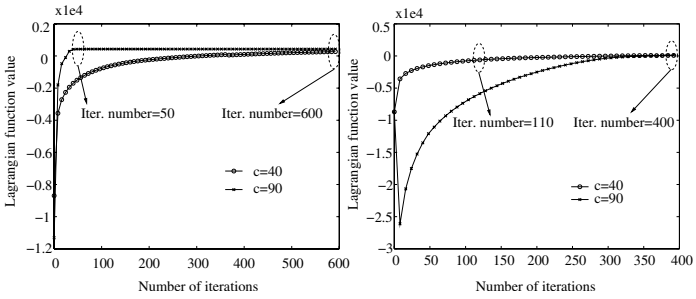
$$\begin{aligned} &\textit{Maximize:} && Q_{\vec{\lambda}} \\ &\textit{Subject to:} && \lambda_e = \sum_{e'\in des(e)} \lambda_{e'} \\ & && \forall e \in E , \lambda_e \geq 0 , \end{aligned} \quad (5)$$

where $Q_{\vec{\lambda}}$ is defined as the minimal value of $L_{\vec{\lambda}}(\vec{W})$ for any $\vec{W}$.

LRI($T,B,\mathbf{W},\mathbf{q}$)
1 initialize $\vec{\lambda} = \vec{\lambda}_0$
2 **repeat**
3 $\forall e \in E, \forall j \in \{1,\ldots,|B_e|\}$, compute $w_{e,j}$ to minimize $L_{\vec{\lambda}}(\vec{W})$
4 update $\vec{\lambda}$
5 **if** convergence **return** all $w_{e,j}$

Figure 3: Framework of solving the Lagrange dual problem.

Figure 3 shows the pseudocode of a general framework that can be used to solve the Lagrange dual problem. Specifically, $\vec{\lambda}$ is initialized in Line 1. During each iteration in Lines 2–5, $L_{\vec{\lambda}}(\vec{W})$ is first minimized using current $\vec{\lambda}$. Based on the results, a new $\vec{\lambda}$ is derived. If the solution has not converged under a fixed threshold, the iteration continues. Otherwise, the current solution $w_{e,j}$ is returned.



(a) (b)

Figure 4: Convergence rate of subgradient method with different c. Each element of $\vec{\lambda}_0$ in (a) is 0.7 times that in (b).

It is worth mentioning that $L(\vec{W},\vec{a},\vec{\lambda}) = \sum_{e\in E} W_e$ holds in the optimal solution [22]. Consequently, the convergence can be defined as $(L(\vec{W},\vec{a},\vec{\lambda}) - \sum_{e\in E} W_e)/\sum_{e\in E} W_e) < \varepsilon$, where ε is a preselected threshold.

6. HEURISTIC SOLVER

In this section, our contributions are presented that lead to a fast low-power repeater insertion solver.

6.1 Update of Lagrangian Multipliers

The computation time of solving the Lagrangian dual problem is the product of runtime per iteration and iteration times. Consequently, the convergence speed of the Lagrangian multiplier update affects the algorithm efficiency significantly. Since objective function $Q(\vec{\lambda})$ is proved to be concave [22], previous repeater insertion algorithms based on the Lagrangian relaxation framework use the subgradient method to perform the update of Lagrangian multipliers $\vec{\lambda}$. Specifically, after the kth iteration during which the repeater insertion solution $w_{e,j}$ is derived that minimizes $L_{\vec{\lambda}}(\vec{W})$ for $\vec{\lambda} = \vec{\lambda}_k$, the timing slack at each sink t_i is calculated by examining the delay d_e of each interconnect segment e. The Lagrangian multipliers are then updated as $\vec{\lambda}_{k+1} = \vec{\lambda}_k - \rho_k \vec{g}_k$, where $\vec{g}_k$ is the vector comprising the timing slacks at all sinks and $\rho_k > 0$ is the step size of iteration k that must satisfy $\sum_{k=1}^{\infty} \rho_k \to \infty$ and $\lim_{k\to\infty} \rho_k = 0$.

Although the subgradient method guarantees to find the optimal solution in a concave solution space, its convergence rate is often slow and can change significantly depending on the choice of the parameter ρ_k. Figure 4(a) shows the convergence results of a typical interconnect design using the subgradient method. The step size function of kth iteration ρ_k is set to c/k as in [15], where c is a tuning constant. As can be seen, when c changes, the number of iterations before convergence can change by more than 10 times, from 50 to 600. Furthermore, even the same c can result in significantly different convergence speed when the initial $\vec{\lambda}_0$ changes. Figure 4(b) shows the convergence results using the same c values but different $\vec{\lambda}_0$ than that in Figure 4(a). The c that results in faster(slower) convergence rate in Figure 4(a) leads to slower(faster) rate in Figure 4(b). Consequently, manual tuning of c and $\vec{\lambda}_0$ is usually required for fast convergence.

We use a completely different mathematical scheme, called the ellipsoid method [20], to compute $\vec{\lambda}_{k+1}$ at the end of iteration k. The basic principle of the ellipsoid method is described as follows. First, an n-dimensional ellipsoid $Z(A,\vec{x}) = \{\vec{z} | (\vec{z}-\vec{x})^T A^{-1} (\vec{z}-\vec{x}) \leq 1\}$ is created that contains the optimal solution where A is a $n \times n$ matrix. The volume and center of the ellipsoid are $|DetA|$ and $\vec{x}$, respectively. In each iteration, a new ellipsoid is created with a reduced volume while still keeping the optimum inside. The volume reduction factor is constant for the given solution space dimension

n. The optimal solution will be derived as the ellipsoid center, when the ellipsoid volume is smaller than a preset threshold.

In our repeater insertion algorithm, the n independent Lagrangian multipliers form an n-dimensional space. The first ellipsoid is centered at the initial $\vec{\lambda}_0$ with the matrix A being a diagonal matrix with the elements $a_{ii} = \lambda^2_{max_i}$, $i \in \{1,2,\ldots,n\}$, where λ_{max_i} is the maximal possible Lagrangian multiplier of the interconnect segment connected to the sink t_i. The function ELP that computes the new ellipsoid, defined by a new $(\vec{\lambda}, A)$ pair, during each iteration is shown in Figure 5. In particular, the timing slacks $\vec{S}$ at n sinks are first computed. Based on the values of $\vec{S}$ and A, a vector $\vec{g}$ is derived, which assists the creation of the new ellipsoid in Lines 3–4. It is worth mentioning that ELP in Figure 5 is only valid for the case $n \geq 2$. When $n = 1$, the ellipsoid becomes an interval and the ellipsoid method becomes a simple bisection algorithm [20].

ELP($\vec{\lambda}, A$)
1 compute the slacks at all the sinks $\vec{S}$
2 $\vec{g} = \vec{S}/\sqrt{\vec{S}^T A \vec{S}}$
3 $\vec{\lambda} = \vec{\lambda} - \frac{1}{n+1} A\vec{g}$
4 $A = \frac{n^2}{n^2-1}(A - \frac{2}{n+1} A\vec{g}\vec{g}^T A)$
5 **return** $(\vec{\lambda}, A)$

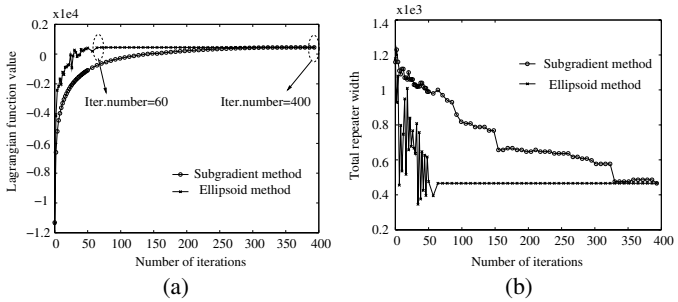
Figure 5: Algorithm ELP.

In our experiment, we observe a faster convergence speed of the ellipsoid method compared with the subgradient method. Figure 6 illustrates the convergence speeds of the subgradient and ellipsoid methods for a typical design. As can be seen, the ellipsoid method converges more than 6 times faster the subgradient method, although it oscillates a little initially.

In contrast to the subgradient method, the ellipsoid method does not have a moving step function that needs to be carefully selected and therefore is much more stable and easy to apply. Furthermore, the convergence rate of the ellipsoid method is hardly affected by the initial solutions. Figure 7 shows the convergence results of both the ellipsoid method and the subgradient method under the same pair of initial $\vec{\lambda}_0$. It clearly shows that change of convergence speed due to the initial solution is negligible for the ellipsoid method, whereas the subgradient method is very sensitive to the initial solution. Consequently, no time-consuming initialization procedure is needed for the ellipsoid-method-based repeater insertion schemes, resulting in further runtime reduction.

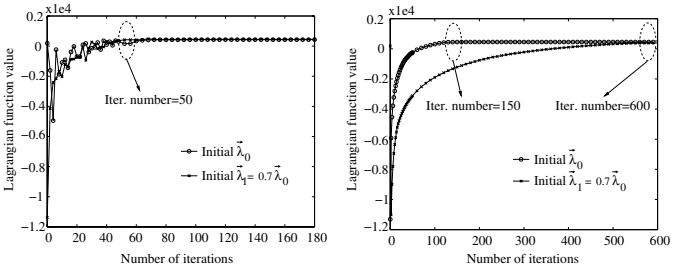
6.2 Local Optimization

Another key component in solving the Lagrangian dual problem is the derivation of the repeater insertion solution within each iteration to minimize $L_{\vec{\lambda}}(\vec{W})$ for a given $\vec{\lambda}$. In [15], Liu *et al.* proposed an iterative heuristic to solve this problem. Specifically, each



(a) (b)

Figure 6: Convergence rate of the subgradient and ellipsoid methods. (a) Lagrangian function (b) Total repeater width.



(a) (b)

Figure 7: Effect of initial $\vec{\lambda}_0$ on convergence rate (a) ellipsoid method (b) subgradient method.

branch in the interconnect tree was optimized individually using the DP algorithm. Since the branches were solved in an inverse topological sort order, the upstream driving resistance of each branch was not known. An estimated value had to be used, which could be inaccurate and lead to suboptimal solutions. To improve the solution quality, once the repeater insertion results of all branches were computed, they were used to derive the estimates of the upstream driving resistance for each branch and the problem was solved again. The iteration continued until convergence. In our paper, we denote this branch-based DP scheme BDP.

Algorithm BDP relies on iterations and therefore cannot guarantee the optimality of the solution. Furthermore, since the local optimization step is already inside the loop of the Lagrangian multiplier update, the entire repeater insertion algorithm becomes a double-nested iteration loop, potentially leading to long runtimes. We next present a DP-based algorithm that guarantees to get the minimal $L_{\vec{\lambda}}(\vec{W})$ without any iteration.

Our algorithm computes a set of possible repeater insertion *scenarios* $S_{e,i}$ for each repeater candidate location $b_{e,i} \in B_e, i = 1,2,\ldots,|B_e|$ along every segment e as well as the starting point $u(e)$ and ending point $v(e)$. Each scenario $S_{e,i}$ is a tuple (c,l) where c is the downstream capacitance at $b_{e,i}$ and l represents the total contribution to the value of $L_{\vec{\lambda}}(\vec{W})$ from all the downstream candidate locations including $b_{e,i}$.

The pseudocode of our tree-based DP scheme, called TDP, is shown in Figure 8. Without loss of generality, the interconnect tree $T = (E,V)$ is assumed to be a binary tree, since a fork point of a fanout $n > 2$ can be converted into $n-1$ fork points with a fanout of 2. As in the problem formulation, a repeater width 0 is added into the repeater library $\mathbf{W}$, indicating no repeater is inserted. The interconnect segments are processed in an inverse topological sort order. For each e, the solution set at its endpoint $v(e)$ is derived first in Lines 3–11. Specifically, if $v(e) = t_i, i \in \{1,2,\ldots,n\}$, i.e., e connects to sink i, $S_{v(e)} = \{(c_{t_i}, 0)\}$, where c_{t_i} is the gate capacitance of t_i. Otherwise, if e has only one descendant e', $S_{v(e)}$ is set to that of the starting point of e', $S_{u(e')}$, since $v(e)$ and $u(e')$ are the same vertex. If e has two descendants e_L and e_R, $S_{v(e)}$ is derived by combining any pair of scenarios formed by one from $S_{u(e_L)}$ and one from $S_{u(e_R)}$. It is worth mentioning that a scenario (c,l) is inferior to (c^*,l^*) if $c > c^*$ and $l > l^*$. Therefore, we prune the set to delete the inferior ones each time a scenario is inserted. After the computation of $S_{v(e)}$, TDP derives the scenario sets for all the candidate repeater locations along e in Lines 12–20, in an order opposite to the signal propagation direction. In particular, for each location, all possible repeater widths including 0 are analyzed and the corresponding results are stored. Note that $v(e)$ and $u(e)$ are denoted as the candidate locations $|B_e|+1$ and 0, respectively, to simplify the code. The parameter $\Delta_{e,i}$ represents the interconnect length between candidate locations $b_{e,i}$ and $b_{e,i+1}$. The value c_e is the unit-length capacitance of edge e. The symbol $d_{e,i,w}$ represents

the signal delay from $b_{e,i}$ to $b_{e,i+1}$ when a repeater with a width w is inserted at $b_{e,i}$. In Lines 21–24, $S_{u(e)}$ is derived in a similar fashion. After all edges have been processed, the scenario at the root of the interconnect tree T with the minimum l is selected to derive the repeater insertion solution at Line 25.

Since it is based on the DP technique, our approach guarantees to derive the repeater insertion solution that minimizes $L_{\vec{\lambda}}(\vec{W})$ for a given $\vec{\lambda}$. Moreover, it performs no iteration. The weakness of our approach is that, in the worst case, the sizes of solution sets will increase exponentially from the sinks to the source and therefore require a large amount of memory, leading to long CPU time.

TDP(T,λ,B)
1 initialize all scenarios to empty set ϕ
2 **for** each edge $e \in E$ in an inverse topological sort order
3 **if** $v(e) \in t_i, i = 1,2,\ldots,n$
4 $S_{v(e)} = \{(c_{t_i}, 0)\}$
5 **else if** e has a single descendant e'
6 $S_{v(e)} = S_{u(e')}$
7 **else**
8 **for** any $(c_1,l_1) \in S_{u(L_e)}$ and $(c_2,l_2) \in S_{u(R_e)}$
9 $c = c_1 + c_2$, $l = l_1 + l_2$
10 insert (c,l) into $S_{v(e)}$
11 prune $S_{v(e)}$
12 **for** $i = |B_e|, 1$
13 **for** each scenario $(c,l) \in S_{e,i+1}$
14 **for** each $w \in \mathbf{W}$
15 **if** $w = 0$
16 $c' = c + c_e * \Delta_{e,i}$, $l' = l + \lambda_e * d_{e,i,w}$
17 **else**
18 $c' = C_o * w$, $l' = l + w + \lambda_e * d_{e,i,w}$
19 insert (c',l') into $S_{e,i}$
20 prune $S_{e,i}$
21 **for** each scenario $(c,l) \in S_{e,1}$
22 $c' = c + c_e * \Delta_{e,0}$, $l' = l + \lambda_e * d_{e,0,w}$
23 insert (c,l) into $S_{u(e)}$
24 prune $S_{u(e)}$
25 derive the repeater insertion solution based on the scenario at the interconnect root with the minimal l

Figure 8: Algorithm TDP.

6.3 Algorithm Summary

Figure 9 shows the pseudocode of our fast low-power repeater insertion solver called FREEZE. Given an interconnect tree T with n sinks, repeater location candidates B, a repeater library $\mathbf{W}$, and timing constraints $\mathbf{q}$ for all the sinks, FREEZE returns the repeater insertion solution that minimizes total repeater width and satisfies the timing targets. Specifically, our scheme first initializes the Lagrangian multipliers $\vec{\lambda}$ and the ellipsoid diameter A in Line 1. FREEZE then iteratively maximizes the objective function $Q(\vec{\lambda})$ in Lines 3–7. In particular, if the number of sinks is less than 4, algorithm TDP will be used. Otherwise, the algorithm BDP from [15] will be applied. Such a choice is based on the observation that, TDP derives solutions for each given $\vec{\lambda}$ very fast for trees with a small number of sinks, a property shared by most global interconnects, since the numbers of the candidate solutions do not increase significantly. Whereas algorithm BDP still needs many iterations to reach convergence in this case and, therefore, is slow. However, for the trees with a large number of sinks and candidate locations, the size of the solution set may increase significantly from sinks to the source when TDP is used. In this case, FREEZE chooses algorithm BDP. After the local optimization derives a repeater insertion solution, the diameter A and center $\vec{\lambda}$ of the ellipsoid are updated by ELP in Line 7. Our algorithm returns the final solution in Line 9 when convergence is reached.

FREEZE($T, B, \mathbf{W}, \mathbf{q}$)
1 initialize A and $\vec{\lambda}$
2 **do**
3 **if** $n < 4$
4 Call TDP to compute all $w_{e,j}$
5 **else**
6 Call BDP to compute all $w_{e,j}$
7 (A, $\vec{\lambda}$)=ELP(A, $\vec{\lambda}$)
8 **while** $(L(\vec{W},\vec{a},\vec{\lambda}) - \sum_{e\in E} W_e)/\sum_{e\in E} W_e) < \varepsilon$
9 **return** all $w_{e,j}$

Figure 9: Algorithm FREEZE.

7. EXPERIMENT SETUP AND RESULTS

We applied our scheme to various interconnect trees to demonstrate its effectiveness. Specifically, our interconnect trees were generated using TSMC 0.18 μm technology. The total number of tree branches ranged from 3 to 30. The length of branches ranged from 1000 to 5000μm. Each tree branches contained 1 to 5 segments which might be routed on different metal layers. The circuit parameters of repeaters and interconnects, e.g., unit-length wire capacitance and unit-width gate capacitance, were extracted from TSMC technology files and calibrated using SPICE simulations. We assigned each tree segment with several candidate repeater locations using the segmenting scheme in [2]. The buffer library $\mathbf{W}$ contained repeaters from $1u$ to $400u$ with a granularity of $10u$, where u is the minimal repeater width. The timing targets ranged from $1.05\tau_{min}$ to $1.65\tau_{min}$, where τ_{min} is the minimal delay of the interconnect tree that can be achieved by repeater insertion.

We used our tool FREEZE to optimize the interconnect designs. Specifically, each element of the initial $\vec{\lambda}_0$ was set to 0. The value λ_{max_i} was chosen as $10W_{ub}/|s_i|$, where W_{ub} was an upper bound of the total repeater width calculated using the entire tree length and s_i was the worst-case timing slack estimated under the assumption that no repeater was inserted. The value of ε was chosen to be 1%.

For comparison purposes, we implemented two low-power repeater insertion schemes. The first one is from [15] which uses subgradient method to update the Lagrangian multipliers and algorithm BDP to perform the local optimization. It is worth mentioning that the original scheme in [15] chooses $\rho_k = c/k$ for the subgradient moving step size. We chose $c/\sqrt{k}$, however. The reason is that during our experiment, we found that for 70% of the nets, convergence speed was extremely slow with no convergence reached after 2000 iterations, if $\rho_k = c/k$, unless manual tuning of the parameter c was performed for each individual interconnect. On the other hand, when the step size $c/\sqrt{k}$ was used, most nets could reach convergence in less than 600 iterations. In our experiment, c was set to 10. We also implemented a DP-based algorithm [14] and used it to evaluate the design quality of FREEZE, since the DP-based method guarantees to derive the optimal results. We use LRS and PDP as the names of the schemes in [15] and [14], respectively.

Table 1 shows the experimental results of 25 interconnect trees. Columns 2–3 show the number of sinks and the total candidate repeater locations in the interconnects. Column 4 shows the average runtimes of our scheme FREEZE under different timing target. Columns 5–6 give the corresponding runtimes of schemes LRS and PDP, respectively. Columns 7–8 list the speedup of FREEZE over LRS and PDP. As can be seen, our scheme runs significantly faster. The average speedups are 9.2 and 17.5 in comparison with LRS and

Table 1: Experimental results.

			Run time(s)			Speedup		δP
Net	Sink	Loc	FREEZE	LRS	PDP	S_{LRS}	S_{PDP}	
1	2	54	0.7	16.0	23.2	27.4	37.2	0.2
2	2	48	0.6	13.2	22.3	26.1	40.4	0.3
3	2	54	0.8	15.5	22.9	23.6	32.9	0
4	2	60	1.0	18.7	38.0	26.5	50.4	0
5	2	48	0.8	13.9	14.2	21.7	21.0	0
6	3	72	2.0	10.0	18.6	5.6	10.1	1.3
7	4	84	3.1	12.7	21.8	6.0	9.8	2.4
8	5	108	4.2	18.3	21.7	6.0	6.0	0.5
9	5	96	3.9	15.6	38.2	5.2	11.9	0.0
10	6	96	4.8	14.8	67.2	4.4	18.3	0.1
11	6	108	5.4	17.0	48.8	4.6	12.0	0.5
12	6	114	4.8	18.2	31.5	5.5	8.8	0.4
13	7	114	6.4	21.3	54.8	5.3	12.2	0.2
14	7	126	6.2	24.5	80.8	5.4	16.7	0.9
15	8	132	8.4	25.7	69.0	4.4	11.1	0.2
16	11	192	12.7	61.5	134.3	9.5	21.5	0.1
17	4	78	2.6	9.3	17.9	4.5	8.4	0.4
18	3	78	2.3	10.2	27.3	5.4	13.9	0.0
19	4	90	3.0	11.6	26.4	4.7	9.5	1.3
20	4	84	2.6	11.5	18.0	5.6	7.9	3.1
21	5	114	5.7	18.4	49.9	4.9	11.8	0.2
22	5	96	3.8	14.5	66.6	5.5	22.4	1.0
23	5	108	4.9	15.7	32.9	4.3	8.5	0.4
24	6	120	6.0	19.8	76.2	4.0	17.2	6.2
25	5	114	5.6	21.9	75.8	4.9	17.1	2.0
Ave		-	-	-	-	9.2	17.5	0.9

PDP. The corresponding maximum speedups are 27.4 and 50.4, respectively. The average iteration number of FREEZE is about 60, whereas average number of iterations for algorithm LRS is 350. Column 9 compares the power dissipation results from FREEZE and PDP. The power degradation of our scheme is only 0.9% on the average across all designs. This degradation can be further reduced by decreasing the iteration-ending threshold at the cost of longer runtimes. The power results of FREEZE is even about 1% better than that of LRS on the average. Our experiments were performed on a Pentium-IV 2.8GHz machine with 1GB memory running Redhat Linux9.0.

8. CONCLUSION

This paper presents a repeater insertion tool called FREEZE for the power minimization of realistic interconnect trees under given timing budgets. The novel contribution is the adoption of the ellipsoid method to achieve the efficient and effective solution space exploration. In contrast to most of the previously proposed schemes, our scheme is robust and does not need any manual tuning of the iteration-control parameters or the initial solution.

Our approach is capable of producing high-quality results at a very fast speed. Experimental results demonstrate that, in comparison with the state-of-the-art low-power repeater insertion schemes, FREEZE requires 5.8 times fewer iterations on the average, achieving up to 27 times speedup with better power savings. When compared with the dynamic programming based scheme, which guarantees the optimal solution, our tool delivers 17x speedup with only a 0.9% power increase on the average.

In our current implementation, we have chosen simple RC models to compute the interconnect delay and power so that a fair comparison can be made between our scheme and previous proposed techniques. We are currently applying our scheme in conjunction with more accurate circuit models that consider the signal slew rate. It is an interesting future research topic to combine our scheme and global interconnect routing for further power savings. In addition, the convergence speed of the ellipsoid method becomes slow when the dimension of the solution space increases significantly. The derivation of efficient low-power repeater insertion algorithms for interconnects of hundreds of sinks remains an open research problem.

9. REFERENCES

[1] C. Alpert, A. Devgan, and S. T. Quay. Buffer insertion for noise and delay optimization. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 18(11):1633–1645, Nov. 1999.

[2] C. J. Alpert and A. Devgan. Wire segmenting for improved buffer insertion. In *Design Automation Conference*, June 1997.

[3] C. J. Alpert, A. Devgan, and S. T. Quay. Buffer insertion with accurate gate and interconnect delay computation. In *Design Automation Conference*, June 1999.

[4] C. J. Alpert, J. Hu, S. S. Sapatnekar, and P. G. Villarrubia. A practical methodology for early buffer and wire resource allocation. *IEEE Trans. CAD*, 22(5), May 2003.

[5] H. B. Bakoglu. *Circuits, Interconnects, and Packaging for VLSI*. Reading, MA: Addison-Wesley, 1990.

[6] K. Banerjee and A. Mehrotra. A power-optimal repeater insertion methodology for global interconnects in nanometer designs. *IEEE Trans. VLSI Systems*, 49(11):2001–2007, Nov. 2002.

[7] C. P. Chen and N. Menezes. Noise-aware repeater insertion and wire sizing for on-chip interconnect using hierarchical moment-matching. In *Design Automation Conference*, June 1999.

[8] C.-C. N. Chu and D. F. Wong. Closed form solution to simultaneous buffer insertion/sizing and wire sizing. In *Inter. Symp. on Physical Design*, Apr. 1997.

[9] J. Cong, L. He, C. K. Koh, and P. H. Madden. Performance optimization of VLSI interconnect layout. *Integration, the VLSI Journal*, 21(1):1–94, Jan. 1996.

[10] G. S. Garcea, N. P. van der Meijs, and R. H. Otten. Simultaneous analytical area and power optimization for repeater insertion. In *Inter. Conf. on CAD*, Nov. 2003.

[11] L. P. P. P. van Ginneken. Buffer placement in distributed RC-tree networks for minimal Elmore delay. In *Proc. Intl. Symposium on Circuits and Systems*, 1990.

[12] N. Hedenstierna and K. O. Jeppson. CMOS circuit speed and buffer optimization. *IEEE Trans. CAD*, 6(2):270–280, Feb. 1987.

[13] P. Kapur, G. Chandra, and K. C. Saraswat. Power estimation in global interconnect and its reduction using a novel repeater optimization methodology. In *Design Automation Conference*, June 2002.

[14] J. Lillis, C. K. Cheng, and T.-T. Y. Lin. Optimal wire sizing and buffer insertion for low power and a generalized delay model. *J. of Solid-State Circuits*, 31(3):437–447, Mar. 1996.

[15] I.-M. Liu, A. Aziz, and D. F. Wong. Meeting delay constraints in DSM by minimal repeater insertion. In *Proc. IEEE Int. Conf. Design, Automation and Test Eur.*, Mar. 2000.

[16] X. Liu, Y. Peng, and M. C. Papaefthymiou. Practical repeater insertion for low power: What repeater library do we need? In *Design Automation Conference*, June 2004.

[17] X. Liu, Y. Peng, and M. C. Papaefthymiou. RIP: An efficient hybrid repeater insertion scheme for low power. In *Design, Automation, and Test in Europe*, Mar. 2005.

[18] A. Nalamalpu and W. P. Burleson. A practical approach to DSM repeater insertion: Satisfying delay constraints while minimizing area and power. In *IEEE International ASIC/SOC Conference*, Sept. 2001.

[19] M. Nekili and Y. Savaria. Optimal methods of driving interconnections in VLSI circuits. In *International Symposium on Circuits and Systems*, May 1993.

[20] A. Nemirovsky and D. Yudin. *Informational complexity and efficient methods for solution of convex extremal problems*. J. Wiley & Sons, New York, 1983.

[21] R. Otten. Global wires harmful? In *Inter. Symp. on Physical Design*, Apr. 1998.

[22] J. F. Shapiro. *Mathematical Programming: Structures and Algorithms*. Wiley-Interscience Publication, 1979.

[23] D. Sylvester and K. Keutzer. Getting to the bottom of deep submicron. In *Inter. Conf. on CAD*, Nov. 1998.

[24] D. Sylvester and K. Keutzer. A global wiring paradigm for deep submicron design. *IEEE Trans. CAD*, 19(2):242–252, Feb. 2000.