

Optimization of the Power/Ground Network Wire-Sizing and Spacing Based on Sequential Network Simplex Algorithm

Ting-Yuan Wang Charlie Chung-Ping Chen
Department of Electrical and Computer Engineering
University of Wisconsin at Madison
Madison, WI 53706
wangt@cae.wisc.edu chen@engr.wisc.edu

ABSTRACT

This paper presents a fast algorithm to optimize both the widths and lengths of power/ground networks under reliability and power dip/ground bounce constraints. The space-sizing which allows the length to change gives more flexibility in solving practical problems. There are two major contributions of this paper. First, we prove that for general topology, a relaxed version of this problem is also convex. Second, we present the sequential network simplex algorithm which can solve those problems with extreme efficiency. Experimental results on several large scale problems, using a PC with a 500-MHZ Pentium III processor, show that our algorithm can solve problems with hundreds of thousands of variables within a few minutes and has a speed improvement of 25+ over sequential linear programming [1]. Experimental results also show that about 50% of the power delivery area can be reduced using our algorithm.

1. INTRODUCTION

More than ever, accurate, high-frequency operation of integrated circuits, with smaller devices buried under an expanding superstructure of interconnect layers, depends upon advanced design in power delivery technology. To ensure the quality of power delivery, advanced technologies of packaging material and methods such as organic material, flip-chip, and C4 bump technologies have been successfully deployed. However, the relentless push for low power has driven supply voltage requirements below the 1 volt region while at the same time increasing supply current requirements because of increasing transistor counts. As a result, on-chip power delivery needs to be significantly improved in order to fulfill these stringent requirements.

There are several physical-design means to improve the quality of power delivery. Wire-sizing has been shown to be an effective way to reduce the power dip/ground bounce as well as improve the electrical migration [2], [3], [4], [5], [1]. Topology optimization is another effective way to adjust the power delivery network topology to fit the current supply pattern [6], [7], [8].

There are several existing approaches to solve the wire-sizing and topology optimization problem. The wire-sizing problem was formulated as a nonlinear programming problem and solved by Augmented Lagrangian with Newton's method or the steepest descent method [2].

There are several difficulties involved in the wire-sizing and topology optimization of power delivery networks. First, the problem sizes are very large. For a 50×50 network, the number of variables and constraints can easily exceed millions. Second, this problem is non-convex in nature and hence no existing mathematical programming algorithm can guarantee optimality.

To tackle the above mentioned difficulties, a feasible direction method was developed to iteratively minimize the total wire area [5]. The other approach used both the nodal voltages and branch currents as variables, then divided the problem into two phases. The first phase reduces to a nonlinear convex problem and can be solved by a conjugate gradient solver. The second phase is a linear programming problem and is easy to solve. The convex and linear characteristics make this method much more attractive and easy to solve more efficiently [4].

However, the constrained nonlinear programming problem solved by the conjugate gradient method is actually too slow for today's VLSI design. The idea is to translate the constrained nonlinear programming problem into a sequence of linear programming problems. This method was demonstrated to be orders of magnitude faster than the best-known method based on the conjugate gradient method[1].

This paper presents a fast algorithm to optimize both the widths and lengths of power/ground network under reliability and power dip/ground bounce constraints. The space-sizing which allows the length to change gives more flexibility in solving the practical problem. There are two major contributions of this paper. First, we prove that for general topology, a relaxed version of this problem is also convex. Second, we present a sequential network simplex algorithm which can solve those problems with extreme efficiency.

We organize the remainder of this paper as follows. Section 2 gives an overview of the problem and formulates the problem. A new approach based on the network simplex method is discussed in Section 3. Experimental results are presented in Section 4, followed by the conclusion in Section 5.

2. PROBLEM FORMULATION

Consider a power/ground network $G = \{N, B\}$ with n nodes $N = \{1, \dots, n\}$ and b branches $B = \{1, \dots, b\}$ as shown in Figure 1. It is assumed that the average current drawn in

each block is given. A leaf node connects an independent current source which models the current drawn from the block. Let I_i be the current in branch i with direction from node i_1 to node i_2 . The voltage level at node i is expressed as V_i . Let r_i , L_i , and w_i represent the resistance, length, and width of branch i , respectively. In this paper, we suppose that the width L_i can be adjusted within a certain range. The resistance can be expressed as follows:

$$r_i = \rho \frac{L_i}{w_i} = \frac{V_{i1} - V_{i2}}{I_i}, \quad (1)$$

where ρ is the sheet resistivity.

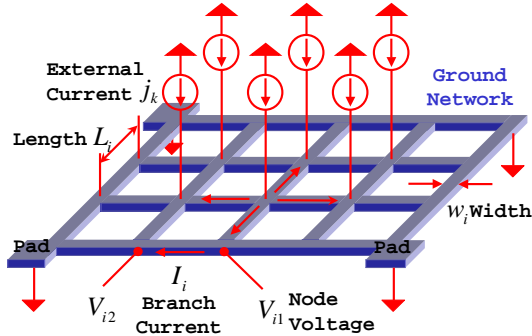


Figure 1: Power Grid

Then the area optimization problem is to minimize the total power/ground routing area in terms of voltages, currents, and lengths of branches. The objective function can be expressed as follows:

$$F(V, I, L) = \sum_{i \in B} L_i w_i = \sum_{i \in B} \frac{\rho I_i L_i^2}{V_{i1} - V_{i2}}, \quad (2)$$

and is subject to the following constraints in order to satisfy the feasibility and reliability requirements of the power/ground networks.

- **The voltage drop constraints**

On the power net, the voltage fluctuation from a power pad to a leaf node must be restricted, and thus give a lower bound on the voltage level. On the ground net, similar restrictions will give an upper bound on the voltage level. Thus the voltage constraints are

$$\begin{aligned} V_{i \in L} &\geq V_{min} \quad \text{for power grid,} \\ V_{i \in L} &\leq V_{max} \quad \text{for ground grid,} \end{aligned} \quad (3)$$

where L is the set of leaf nodes and V_{min} and V_{max} are the given lower and upper bounds.

- **The minimum width constraints**

The minimum width of a branch is technologically restricted by the metal layer on which the power/ground lines. These constraints can be expressed as follows:

$$w_{i \in B} = \frac{\rho L_i I_i}{V_{i1} - V_{i2}} \geq w_{min}, \quad (4)$$

where w_{min} is the given minimum width.

- **The electro-migration constraints**

Electromigration is the transport of mass in metals under the stress of high current density. This metallization failure is a reliability concern in IC design. The lifetime of power/ground interconnects is modeled by Black's equation as shown below [9]:

$$MTF = A J_{avg}^{-2} \exp(E_a / \kappa T) \quad (5)$$

where J_{avg} , E_a , κ , and T are average current density, activation energy, Boltzman's constant, and temperature, respectively. Therefore an upper bound on the current density of each branch is required and can be expressed as follows:

$$\begin{aligned} \frac{|I_i|}{w_i} &\leq \sigma \\ \text{or } |V_{i1} - V_{i2}| &\leq \rho \sigma L_i, \end{aligned} \quad (6)$$

where σ is the current density for fixed thickness.

- **Kirchoff's current law (KCL)**

The currents in branches connected to a node are restricted by Kirchoff's current law.

$$\sum_{j \in N(i)} I_j = 0, \quad (7)$$

where $N(i)$ is the set of branches that are connected to the node i .

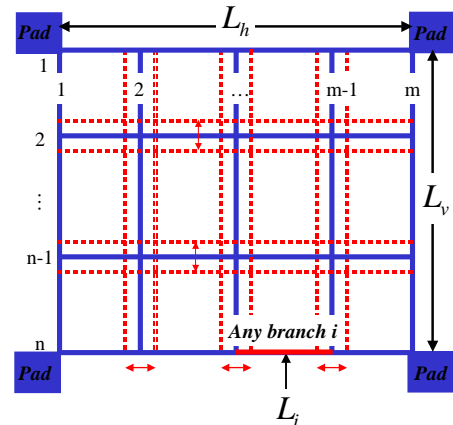


Figure 2: Length Constraints

- **The length constraints**

We allow the power/ground grid lines to move up and down and left and right under certain ranges. However, the distances between pads are fixed as shown in Figure 2. The length constraints can be expressed as follows

$$L_{i,lower} \leq L_i \leq L_{i,upper}, \quad (8)$$

where $L_{i,lower}$ and $L_{i,upper}$ are given lower and upper bounds of branch length L_i . Since the distances between pads are fixed, the following constraints need to be satisfied :

$$\sum_{i=1}^m L_i = L_h, \quad \sum_{j=1}^n L_j = L_v, \quad (9)$$

where m and n are the number of vertical and horizontal power supply lines.

The formulated optimization problem is a nonlinear problem with nonlinear constraints with respect to variables I_i , V_i , and L_i . To solve this problem is complicated and computationally inefficient. A relaxed optimization procedure by dividing the problem into two phases was suggested [4].

2.1 Two-phase optimization procedure

First, get an initial feasible solution satisfying the constraints. Then, divide the problem into phase one (\mathbf{P}_{VL}) and phase two (\mathbf{P}_I).

• Phase one (\mathbf{P}_{VL}):

Assume that branch currents are fixed values, $\{I_i^o\}$, but nodal voltages as well as branch lengths are variables. Then the objective function (2) can be expressed as follows:

$$f_{VL}(V, L) = \sum_{i \in B} \frac{\rho I_i^o L_i^2}{V_{i1} - V_{i2}}. \quad (10)$$

The constraints are subject to: voltage drop constraints (3), electro-migration constraints (6), minimum width constraints (4), and length constraints (8) and (9). Note that the voltage drop across a branch, $V_{i1} - V_{i2}$, must have the same sign as the current direction I_i . The constraints of \mathbf{P}_{VL} can be arranged as follows:

$$\begin{aligned} V_i &\geq V_{min} \quad \text{or} \quad V_i \leq V_{max} \\ |V_{i1} - V_{i2}| &\leq \rho \sigma L_i \\ \frac{V_{i1} - V_{i2}}{I_i^o} &\leq \frac{\rho L_i}{w_{min}} \\ L_{i,lower} &\leq L_i \leq L_{i,upper} \\ \frac{V_{i1} - V_{i2}}{I_i^o} &\geq 0. \end{aligned} \quad (11)$$

• Phase two (\mathbf{P}_I):

In this phase, let the nodal voltages and branch lengths be fixed values, $\{V_i^o, L_i^o\}$, but the branch currents be variables. The fixed values of nodal voltages and branch lengths are from the solutions of \mathbf{P}_{VL} . The objective function can be rewritten as follows:

$$f_I(I) = \sum_{i \in B} \frac{\rho L_i^{o2} I_i}{V_{i1}^o - V_{i2}^o}. \quad (12)$$

The constraints are subject to: Kirchoff's current law (7) and minimum width constraints (4). Similarly, the voltage drop across the branch must have the same direction as the current flow.

$$\begin{aligned} \sum_{j \in B(i)} I_j &= 0 \\ \frac{I_i}{V_{i1}^o - V_{i2}^o} &\geq \frac{w_{min}}{\rho L_i^o}. \end{aligned} \quad (13)$$

2.2 A Linear Programming based Algorithm

\mathbf{P}_{VL} is a linear constrained nonlinear programming problem. An algorithm based on a sequence of linear programs was suggested by [1] to replace the nonlinear programming

problem. Suppose that the initial feasible solution exists, then take the Taylor's expansion of the objective function (10) around the point of initial feasible solution and keep the constant and linear terms. The resulting linear objective function can be written as:

$$f(V, L) = \sum_{i \in B} \left[\frac{2\rho I_i^o L_i^o}{V_{i1}^o - V_{i2}^o} L_i - \frac{\rho I_i^o L_i^{o2}}{(V_{i1}^o - V_{i2}^o)^2} (V_{i1} - V_{i2}) \right]. \quad (14)$$

Now instead of minimizing the nonlinear function f_{VL} , minimize the linear function f iteratively. Theoretically, the sequence of linear programs always converges to an optimal solution of the relaxed convex problem \mathbf{P}_{VL} . It has been proved that there always exists a restriction factor ξ such that the sequence of linear programming converges to a global minimum [1]. The extra constraints required can be expressed as:

$$\begin{aligned} L_i &\geq \xi L_i^o \\ \text{sign}(I_i)(V_{i1} - V_{i2}) &\geq \xi \text{sign}(I_i)(V_{i1}^o - V_{i2}^o). \end{aligned} \quad (15)$$

In the next section, we will prove that the objective function (10) is a convex function.

2.3 The Objective Function of \mathbf{P}_{VL} is Convex

The two-phase optimization procedure shown in Section 2.1 is the relaxed version of the original problem. The objective function (10) in \mathbf{P}_{VL} is a function of V_i and L_i , and it is a convex function. The proof is shown as follows that a relaxed version of the problem for general topology is also convex.

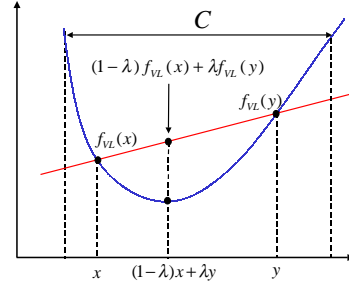


Figure 3: Convex function.

Proof: Suppose that C is a convex set of \mathfrak{R}^n . A function $f_{LV} : C \rightarrow \mathfrak{R}$ is called convex if

$$\begin{aligned} f_{LV}((1-\lambda)x + \lambda y) &\leq (1-\lambda)f_{VL}(x) + \lambda f_{VL}(y), \\ \forall x, y \in C, \forall \lambda \in [0, 1]. \end{aligned}$$

Suppose that the objective function is $f_{VL} = \frac{L^2}{V}$. At points x, y , the functions are:

$$\Rightarrow f_{VL}(x) = \frac{L_1^2}{V_1}, \quad f_{VL}(y) = \frac{L_2^2}{V_2}.$$

$$\begin{aligned}
& (1-\lambda)f_{VL}(x) + \lambda f_{VL}(y) - f_{VL}((1-\lambda)x + \lambda y) \\
&= (1-\lambda)\frac{L_1^2}{V_1} + \lambda\frac{L_2^2}{V_2} - \frac{[(1-\lambda)L_1 + \lambda L_2]^2}{(1-\lambda)V_1 + \lambda V_2} \\
&= \frac{[(1-\lambda)L_1^2]^2}{(1-\lambda)V_1} + \frac{[\lambda L_2^2]^2}{\lambda V_2} - \frac{[(1-\lambda)L_1 + \lambda L_2]^2}{(1-\lambda)V_1 + \lambda V_2} \\
&= \frac{\lambda(1-\lambda)[L_1V_2 - L_2V_1]^2}{V_1V_2[(1-\lambda)V_1 + \lambda V_2]} \geq 0
\end{aligned}$$

Hence the objective function (10) with variables of lengths and nodal voltages is convex. \square

A new approach based on the network simplex method will be presented in the following section because of the convex property of \mathbf{P}_{VL} .

3. A NEW APPROACH BASED ON THE NETWORK SIMPLEX METHOD

Motivated by the algorithm of the sequential linear programming, a new approach based on the network simplex method was used. The reason is that the simplex method is perhaps the most efficient algorithm to solve a linearly constrained linear programming problem. However, if the corresponding minimum cost flow problem has network structure, it can be solved by the network simplex method even faster.

The basic procedure of the sequential network simplex method approach is described as follows. Repeat \mathbf{P}_{VL} and \mathbf{P}_I k times until there is no improvement in the original objective function (2). The solution of each iteration is denoted by \mathbf{x}^k . Since we know that the objective function (10) is a convex function, then we can repeat the following steps l times in \mathbf{P}_{VL} and get a minimum solution. Note that the solution in each subproblem, \mathbf{P}_{VL} , is denoted by \mathbf{x}_l^k . First linearize the nonlinear objective function from (10) into (14), then translate the linear programming problem into the format of a network problem which can be solved by a network simplex algorithm. In order to guarantee that the optimal solution of a network simplex subproblem, (14) (11), is also a minimum solution of the original problem, \mathbf{P}_{VL} (10) (11), we do a line search at each iteration. Repeat the procedures several iterations until there is no improvement in the objective function (10). Then continue the \mathbf{P}_I problem.

3.1 Translate to Network Simplex Method

A network problem finds the minimum cost flow through a network $G = \{N, B\}$ which is defined by a set of n nodes, N , and a set of b branches, B . A branch i in the set B is an ordered pair $(i1, i2)$ with the flow direction from tail $i1$ to head $i2$. Then the network problem can be written in the form:

$$\begin{aligned}
\text{Minimize :} & \quad \sum_{i \in B} c_i x_i \\
\text{Subject to :} & \quad \sum_{i \in T_n} x_i - \sum_{i \in H_n} x_i = s_n \quad \forall (n \in N) \\
\text{Bounds :} & \quad l_i \leq x_i \leq u_i \quad \forall (i \in B).
\end{aligned}$$

- c_i is the cost per unit flow on the branch i .
- x_i is the flow value passing through the branch i .

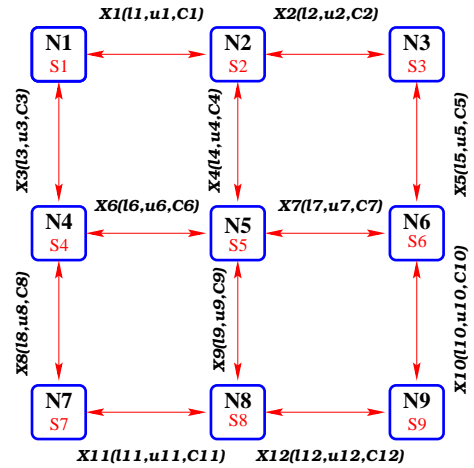


Figure 4: Example of a Network problem

- T_n and H_n are the sets of branches whose tails and heads are node n respectively.
- s_n is the supply/demand value at node n .
- l_i , the lower bound, is the capacity that determines the minimum flow passing through the branch i .
- u_i , the upper bound, is the capacity that determines the maximum flow passing through the branch i .

An example is shown in Figure 4.

The basic idea for the network simplex method [10] is the spanning tree solutions. These solutions are found by fixing the flow of each branch not in a spanning tree having either zero value or flow capacity. For the network simplex algorithm, the minimum cost flow problem always has at least one optimal spanning tree solution. This solution can be found by moving from one solution to another. A new non-tree branch is introduced at each step into the spanning tree to replace one tree branch.

The following procedures will describe how to translate the original problem into a network problem. First we rewrite the objective function (14) and constraints (11) in standard form:

$$\begin{aligned}
\text{Minimize :} & \quad z = \mathbf{p}^T \mathbf{x} \\
\text{Subject to :} & \quad \mathbf{A} \mathbf{x} \geq \mathbf{b} \\
\text{Bounds :} & \quad \mathbf{x} \geq 0.
\end{aligned} \tag{16}$$

The vector \mathbf{x} is $\mathbf{x}^T = [L^T, V^T]$, where $L = \{L_i\}$ and $V = \{V_i\}$ are variables and $\mathbf{x} \in \mathcal{R}^n$. The corresponding coefficient \mathbf{p}^T is $\frac{2\rho I_i^\circ L_i^\circ}{V_{i1}^\circ - V_{i2}^\circ}$ or $\pm \frac{\rho I_i^\circ L_i^\circ}{(V_{i1}^\circ - V_{i2}^\circ)} \mathbf{x}$. Each entry of matrix \mathbf{A} is the coefficient of V_i or L_i in the constraints (11).

Then add non-negative slack variables x_{n+1}, \dots, x_{n+m} to the system in order to have equal signs in constraints. The new vector \mathbf{x}^T is $[x, x_{n+1}, \dots, x_{n+m}]^T$ and $\mathbf{x} \in \mathcal{R}^l$, where $l = n + m$. The new matrix \mathbf{A} is $[A, -I] \in \mathcal{R}^{n \times l}$, where matrix I is the identity matrix. The vector \mathbf{p}^T is $[p^T, 0, \dots, 0]$ and

$\mathbf{p} \in \mathcal{R}^l$. Then the canonical form can be expressed as:

$$\begin{aligned} \text{Minimize :} \quad & z = \mathbf{p}^T \mathbf{x} \\ \text{Subject to :} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ \text{Bounds :} \quad & \mathbf{x} \geq \mathbf{0}. \end{aligned} \quad (17)$$

Note that the set $\{1, 2, \dots, l\}$ is separated into $N = \{1, 2, \dots, n\}$ which corresponds to non-basic variables and $B = \{n+1, \dots, l\}$ which corresponds to basic variables. The matrix $\mathbf{A}_{.B} = -\mathbf{I}$ is called a basis matrix having the property of invertibility. Note that some of the constraints in (11) can be moved to bounds to simplify the constraints. For example, the constraints $V_i \geq V_{min}$ (or $V_i \leq V_{max}$) and $L_{i,lower} \leq L_i \leq L_{i,upper}$ can be moved to the bounds. The linear programming problem now turns out to be a network problem, and can be solved by a network simplex method.

3.2 Line Search

Line search plays an important role in many algorithms for solving a nonlinear programming problem. In our approach, the original nonlinear problem is solved by a sequence of network simplex subproblems. In order to guarantee that the optimal solution of a simplex network subproblem is also a local minimum of the original problem, a linear search is needed at each iteration.

For a given point \mathbf{x}_1^k , find a direction vector d_i and a suitable step size λ_i , which yields a new point $\mathbf{x}_{i+1}^k = \mathbf{x}_1^k + \lambda_i d_i$. Finding the step size λ_i involves solving the subproblem to minimize $f_{VL}(\mathbf{x}_1^k + \lambda d_i)$, which is a one-dimensional search problem in the variable λ . In our problem, \mathbf{x}_{i+1}^k is the new optimal solution of the network simplex method with iteration $l+1$.

Some of the sequential search procedures are available such as dichotomous search, golden section method, and Fibonacci method. We use the dichotomous search [11]. Consider the objective function $\theta(\lambda) = f_{VL}(\mathbf{x}_1^k + \lambda d_i)$ to be minimized over the interval $[a_1, b_1] = [0, 1]$. Choose the distinguishability constant, $2\epsilon > 0$, and the allowable final length of uncertainty, $l_u > 0$. Let $[a_1, b_1]$ be the initial interval of uncertainty, then go to main steps as follows:

1. If $b_k - a_k < l_u$, stop; the minimum point $\lambda_k = \lambda_{min}$ lies in the interval $[a_k, b_k]$. Otherwise, consider λ_k and μ_k defined below, and go to step 2.

$$\lambda_k = \frac{a_k + b_k}{2} - \epsilon \quad \mu_k = \frac{a_k + b_k}{2} + \epsilon$$

2. If $f_{VL}(\lambda_k) < f_{VL}(\mu_k)$, let $a_{k+1} = a_k$ and $b_{k+1} = \mu_k$. Otherwise, let $a_{k+1} = \lambda_k$ and $b_{k+1} = b_k$. Replace k by $k+1$, and go to step 1.

After obtaining the minimum value of $\theta(\lambda)$, the point \mathbf{x}_{i+1}^k will be replaced by $\mathbf{x}_1^k + \lambda_{min} d_i$ for the next iteration of the network simplex method.

3.3 Algorithm

The sequential network simplex algorithm is described in Table 1. Some remarks and descriptions are given below.

1. First analyze the network and get the initial values of L^o , V^o and I^o for L^k , V^k and I^k , respectively. Then repeat \mathbf{P}_{VL} and \mathbf{P}_I k times.
2. Phase one (\mathbf{P}_{VL}):
Translate the problem into network format. For each iteration k , do l iterations of the network simplex method and line search until there is no improvement in f_{VL} . For each l which begins from 1, minimize the objective function (14) which is subject to constraints (11) by the simplex network method. The optimal solution (V_{l+1}^k, L_{l+1}^k) is \mathbf{x}_{l+1}^k . Then do the line search and find the corresponding solution as new (V_{l+1}^k, L_{l+1}^k) for the minimal solution.
3. Phase two (\mathbf{P}_I):
Substitute the values obtained from \mathbf{P}_{VL} for V_i and L_i . Minimize objective function (12) which is subject to constraints (13) by simplex network programming.
4. Iteration k stops if $|F(V^{k+1}, L^{k+1}, I^{k+1}) - F(V^k, L^k, I^k)| \leq \epsilon$.

Sequential Network Simplex Algorithm	
Begin	
/* initial feasible solution */	
$V^k \leftarrow V^o, I^k \leftarrow I^o, L^k \leftarrow L^o;$	
Repeat k	
\mathbf{P}_{VL}:	
Construct $\left\{ \begin{array}{l} \min : f(V^k, L^k, I_{const}^k) \text{ (14)} \\ \text{s.t. : (11)} \end{array} \right. ;$	
$l = 1;$	
$x_l^k = (V_l^k, L_l^k);$	
Repeat l	
Do optimization by simplex network method;	
Optimal Solution:	
$x_{l+1}^k = (V_{l+1}^k, L_{l+1}^k);$	
Do line search;	
New Solution:	
$x_{l+1}^k = \text{new } (V_{l+1}^k, L_{l+1}^k);$	
$l \leftarrow l + 1;$	
Minimal Solution:	
$x_l^k \leftarrow x_{l+1}^k;$	
until no improvement on f_{VL}	
\mathbf{P}_I:	
Construct $\left\{ \begin{array}{l} \min : f_I(V_{const}^k, L_{const}^k, I^k) \text{ (12)} \\ \text{s.t. : (13)} \end{array} \right. ;$	
Do optimization by simplex network method;	
$I^k \leftarrow I^{k+1};$	
until $ F(V^{k+1}, L^{k+1}, I^{k+1}) - F(V^k, L^k, I^k) \leq \epsilon$	
End	

Table 1: Algorithm for Power/Ground Optimization based on Sequential Network Simplex Method

4. EXPERIMENTAL RESULTS

The power/ground optimization algorithm based on the sequential network simplex method was implemented. A set of power/ground network circuits was tested on a PC with

Ckt	Nodes #	Bchs #	Network Simplex Method				Sequential Linear Programming				Speedup
			variable #	constraint #	time (sec)	area reduced	variable #	constraint #	time (sec)	area reduced	
c3x3	16	24	46	84	1	54.96 %	40	52	1	50.00 %	-
c5x5	36	60	106	204	1	53.01 %	96	124	1	50.00 %	-
c10x10	121	220	361	704	1	51.46 %	341	444	2	50.00 %	-
c20x20	441	840	2321	2604	3	50.75 %	1281	1684	18	49.98 %	6.00
c30x30	961	1860	2881	5704	28	50.43 %	2821	3724	245	49.97 %	8.75
c40x40	1681	3280	5041	10004	122	50.10 %	4961	6564	1186	49.96 %	9.72
c50x50	2601	5100	7801	15504	351	49.85 %	7701	10204	3609	49.56 %	10.28
c60x60	3721	7320	11161	22204	479	49.71 %	11041	14644	12374	49.53 %	25.83

Table 2: Comparison of the Network Simplex Method and Sequential Linear Programming

a 500-MHz Pentium III processor. The simulation program was written with C++. The ILOG CPLEX callable library was used to implement the linear programming.

Table 2 lists the names of the circuits, and the number of nodes and branches in columns 1, 2, and 3, respectively. The circuit name, e.g., $c50 \times 50$ means the power grid is 50×50 . The number of variables and constraints is shown for each method. The runtime(time) is in seconds. The reduced chip area is in comparison to the original area.

From the experimental results, the runtime of the network simplex method for $c60 \times 60$ circuit which has 11161 variables and 22204 constraints is 25.83 times faster than the sequential linear programming algorithm. When the power grid is bigger, the runtime of the network simplex method is much faster than the sequential linear programming. On a pure network problem, the performance can be 100 times faster than the primal simplex method. However, it depends on the problems that are designed.

The area reduced for all circuits comparing the original area is about 50%, and it is strongly dependent on the initial solution. For the network simplex method, it is capable of sizing widths and lengths. The total lengths between pads are constants, and the moving range for each node is 20 ~ 30% around the original location. The algorithm we proposed here can reduce the area slightly better than the sequential linear programming algorithm. The limitation comes from the power grid topology. However, it is much flexible for practice problem.

5. CONCLUSIONS

A fast algorithm based on a sequential network simplex method was proposed for simultaneously optimizing the power/ground widths and lengths of wire segments under reliability and power dip/ground bounce constraints. Experimental results have shown that the proposed method can solve problems with hundreds of thousands of variables within a few minutes. Our method also results in solutions 25 times faster over the sequential linear programming approaches. Experimental results also show that the original area can be reduced by about over 50%.

6. REFERENCES

[1] X. Tan, C. J. Richard Shi, D. Lungeanu, and L. Yuan J. Lee. Reliability-constrained area optimization of vlsi power/ground networks via sequence of linear programmings. In *36th ACM/IEEE Design*

Automation Conference, pages 78–83, 1999.

- [2] Salim U. Chowdhury and Melvin A. Breuer. Minimal area design of power/ground nets having graph topologies. *IEEE Trans. on Circuits and Systems*, CAS-34(12):1441–1450, December 1987.
- [3] Salim U. Chowdhury and Melvin A. Breuer. Optimum design of ic power/ground nets subject to reliability constraints. *IEEE Trans. Computer-Aided Design*, 7(7):787–796, July 1988.
- [4] S. Chowdhury. Optimum design of reliable ic power networks having general graph topologies. In *26th ACM/IEEE Design Automation Conference*, pages 787–790, 1989.
- [5] Robi Dutta and Malgorzata Marek-Sadowska. Automatic sizing of power/ground (p/g) networks in vlsi. In *26th ACM/IEEE Design Automation Conference*, pages 783–786, 1989.
- [6] Jaewon Oh and Massoud Pedram. Multi-pad power/ground network design for uniform distribution of ground bounce. In *35th ACM/IEEE Design Automation Conference*, pages 287–290, 1998.
- [7] H. Cai. Multi-pads single layer power net routing in vlsi circuit. In *25th ACM/IEEE Design Automation Conference*, pages 183–188, 1988.
- [8] Zahir A. Syed and Abbas El Gamal. Single layer routing of power and ground networks in integrated circuits. *Journal of Digital Systems*, 6(1):1441–1450, 1982.
- [9] James R. Black. Electromigration – a brief survey and some recent results. *IEEE Transactions on Electron Devices*, ED-16(4):338–347, April 1969.
- [10] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows*. Prentice Hall Inc., 1993. Chapter 11.
- [11] Mokhtar S. Bazaraa, Hanif D. Sherali, and C. M. Shetty. *Nonlinear Programming Theory and Algorithms*. John Wiley & Sons, Inc., 2nd edition edition, 1993. Chapter 8.