

Power Grid Fixing for Electromigration-induced Voltage Failures

Zahi Moudallal
ECE Department
University of Toronto
Toronto, Canada
zahi.moudallal@mail.utoronto.ca

Valeriy Sukharev
Mentor Graphics Corporation
Fremont, USA
valeriy_sukharev@mentor.com

Farid N. Najm
ECE Department
University of Toronto
Toronto, Canada
f.najm@utoronto.ca

Abstract—Electromigration (EM) is a major reliability concern in chip power grids in the wake of smaller feature sizes. EM degradation of grid metal lines can cause large voltage drops on the grid, leading to timing failures and logic errors. During the design process, modifications to the grid design may be required in order to protect from the risk of such EM-induced voltage drop failures. We consider this problem in light of recent efficient full-chip EM assessment techniques. We present a systematic approach that resizes the grid metal lines to meet a design target lifetime while requiring minimal increase in metal area of the grid.

Index Terms—Power grid, VLSI, Electromigration, Design specifications

I. INTRODUCTION

Electromigration (EM) is a growing reliability concern in the design of large integrated circuits (ICs) in the wake of continuing technology scaling. The EM phenomenon is the directional migration of metal atoms in a metal line due to the flow of high current density through the line. While signal and clock lines also suffer from EM degradation, these lines carry bidirectional current and so have longer lifetimes due to so-called *healing*. In contrast, power grid lines carry mostly unidirectional current with no benefit of healing and thus are more susceptible to EM failure. Hence, our focus on EM in power grids.

The power grid of an integrated circuit must deliver the right voltage levels to the underlying logic circuitry in order to guarantee correct logic functionality. The forced redistribution of atoms in the grid metal lines due to EM can lead to tensile stress (pressure) at the grid junctions. Over time, voids may be formed at locations with high tensile stress, which results in a resistance increase and voltage drop degradation. The power grid is deemed to have *failed* if the voltage drop at a any grid node exceeds a user-specified threshold; we refer to this as a *voltage failure*. This voltage drop threshold would correspond to a critical reduction in the supply voltage for some gate, somewhere on the layout, which affects the cell functionality and violates timing constraints. The *EM-lifetime* of the grid refers to the time at which an EM-induced voltage failure is expected to happen.

This work was supported in part by the Semiconductor Research Corporation (SRC), and by the Natural Sciences and Engineering Research Council of Canada (NSERC)

Many authors have focused on developing tools to accurately assess the EM-lifetime of the power grid. Traditional practice for EM assessment is to break up a grid into isolated metal branches, assess the reliability of each branch separately using Black's model [1] and then use the series model (earliest branch failure time) to determine the EM-lifetime of the whole grid. These methods have two main limitations: 1) Black's model ignores the material flow between branches where in today's mesh structured power grids, many branches within the same metal layer may be connected, leading to a so-called interconnect tree structure, and atomic flux can flow freely between the branches of an interconnect tree, and 2) the failure criterion is based on the earliest branch failure instead of the voltage levels on the grid. Recently, an efficient physics-based full-chip EM assessment approach was proposed in [2] which accounts for the material flow and the coupled stresses within an interconnect tree and employs a voltage-based failure criterion. Furthermore, the results of this approach were validated against experimental results as well as *finite element analysis* (FEA) simulations in [3], which makes the method more credible to use.

If the grid's EM-lifetime is determined to be less than the target lifetime, the grid design is said to have an *EM-lifetime violation* and needs to be fixed. Normally, there are several ways to fix an EM-lifetime violation. One way is to lower the current densities through the grid metal lines which is done by modifying the grid design. Another way is to lower the currents drawn by the logic circuitry which is done by changing the circuit layout. In our work, we focus on fixing an EM-lifetime violation that is discovered at late stages of the design cycle. It is very difficult and costly to change the circuit layout as the design gets closer to sign-off and, so, we are only interested to perform minimal modifications to the power grid design in order to fix the violation. Common practice in industry is to iteratively widen metal lines that are close to a void nucleation or voltage drop violation. However, this trial-and-error approach might iterate forever as it "blindly" tries to fix the grid and can lead to the over-design of the grid. Many authors have studied this particular problem [4], [5], [6], [7]. However, the main limitation of these works is that they are based on Black's model and series power grid failure model. In this paper, we propose a power grid fixing scheme based

on the recent work in [2].

II. BACKGROUND

A. Electromigration

Electromigration is the mass transport of metal atoms in a metal line due to momentum transfer between electrons and atoms, which eventually leads to void formation in the metal line. The process of EM degradation can be divided into two phases: *void nucleation* and *void growth*.

Under conditions of high current density, the force exerted by the flow of electrons can cause the metal atoms to move in the direction of the electron flow. If the in-flow of atoms is not equal to the out-flow, certain points within a metal segment may experience high tensile or compressive stresses. In modern chip manufacturing techniques, failure due to compressive stress is not usually observed. However, the build-up of tensile stress eventually leads to formation of a *void* when the stress reaches a critical threshold. This phase of EM degradation, when stress is increasing over time but no voids have yet nucleated, is called the *void nucleation* phase. In this phase, the resistance of a line remains roughly the same as that of a fresh (undamaged) line. Once a void nucleates, the *void growth* phase begins. The void starts to grow in the direction of the electron flow and the line resistance increases towards some high *finite* steady-state value. The conductance of the line decreases but never quite goes to zero.

B. Power Grid Model

An on-die power grid is a multi-layered metallic mesh that is used to deliver power from the external supply pins of a chip to the underlying logic circuitry. On every layer, modern grids consist of long interleaved lines that carry supply and ground, with vias connecting them to the layers above and below. These long structures on every layer have been called *interconnect trees* in the reliability literature, but this name is unfortunate because it leads to confusion with the word “interconnect” as typically used for signal lines, and because even though these lines are often “trees” (acyclic graphs), they do not have to be so. In our work, we allow for both trees and for general (cyclic) graphs, but we will continue to use the term *interconnect trees* to refer to these structures.

There are three types of parasitic effects in power grids: resistive, capacitive and inductive. Because EM is a long-term failure mechanism, short-term transients in chip workload (circuit switching activity) do not play a significant role in EM degradation. Thus, a standard practice in the field is to use a (constant) effective-current model [8] to estimate EM degradation, so that the lifetime of a metal line when carrying the constant effective current and the time-varying transient current is roughly the same. When voids nucleate due to EM, branch resistances change fairly quickly. Correspondingly, the branch currents also change fairly quickly to their new effective values. Hence, between any two successive void nucleations, the power grid has constant (effective) branch currents, voltages and conductances, and so can be modelled

as a DC system consisting of only resistive parasitics. With this, the power grid model can be expressed as

$$G(t)v(t) = u \quad (1)$$

where $G(t)$ is the piecewise-constant conductance matrix (it varies over time, over large time-scales, as the lines age and deform, hence the time dependence), $v(t)$ is the corresponding time-varying but piecewise constant node voltage drop vector and u is the vector of constant effective source current values that model the underlying logic blocks.

C. The Korhonen Model

Korhonen et al. [9] proposed a one-dimensional (1D) model to describe the *hydrostatic stress* (average pressure) σ arising under the influence of electromigration. For a uniform metal line embedded in a rigid dielectric, Korhonen’s model captures the change in σ using the following *partial differential equation* (PDE):

$$\frac{\partial \sigma}{\partial t} = \frac{B\Omega}{k_b T_m} \frac{\partial}{\partial x} \left\{ D_a \left(\frac{\partial \sigma}{\partial x} - \frac{q^* \rho}{\Omega} j \right) \right\}, \quad (2)$$

where j is the current density in the line, $D_a = D_0 e^{-Q/(k_b T_m)}$ is the lognormally distributed [10] effective atomic diffusivity with constant coefficient D_0 , Ω is the atomic volume, k_b is the Boltzmann’s constant, T_m is the temperature in Kelvin, q^* is the absolute value of the effective charge of the conductor, ρ is the resistivity of the conductor, and Q is the activation energy for vacancy formation and diffusion. The corresponding atomic flux J_a in the line is

$$J_a = \frac{D_a C \Omega}{k_b T_m} \left(\frac{\partial \sigma}{\partial x} - \frac{q^* \rho}{\Omega} j \right). \quad (3)$$

Note that EM degradation is highly dependent on the specific microstructure of a given line, which is affected by random manufacturing variations. This randomness is primarily accounted for by the corresponding randomness in D_a . Finally, and as is typical in the field, a void is said to nucleate once the stress exceeds a predefined threshold value $\sigma_{th} > 0$.

D. Stress in an Interconnect Tree

Chatterjee et al. [2] augmented Korhonen’s model by introducing boundary laws to track the material flow between the connected branches. This enables one to evaluate the EM degradation of an *interconnect tree* as a whole. A branch is a continuous straight metal line of uniform width and a junction is any point on the interconnect tree where a branch ends or where a via is located.

The authors show that the stress evolution within a tree can be represented as a Linear Time-Invariant (LTI) system:

$$\dot{\sigma}(t) = A\sigma(t) + B\mu \quad (4)$$

where μ is the input vector which depends on the current densities in the tree’s branches, and A and B are the system and input matrices, respectively, which can be constructed using *state stamps*. The reader is referred to [2] for a detailed description and derivation of the LTI formulation. Furthermore, the initial condition to the above system is the thermal stress at time $t = 0$, i.e. $\sigma(0) = \sigma_T(0)$.

The authors make the simplifying assumption, which is typical in the field, that the diffusivity is the same throughout a branch. As a result, voids nucleate only at junctions of a tree. Once the stress at any junction reaches σ_{th} , a void is said to nucleate at that point and affects all the connected branches. When a void nucleates at a junction, the junction is conceptually treated as a new junction for each of the connected branches such that there is no material flow between these new junctions. Thus, the tree is effectively divided into separate *subtrees*, where the stress evolution in each subtree can be captured by a new LTI system (in the form of (4) using suitable state stamps and initial condition). However, even though there is no material flow between the formed subtrees, the conductivity (electron flow) does not quite go to zero (due to conduction in the metal liner).

E. Voltage-aware EM Analysis

Generally, for a grid to function as intended, the voltage drop at each of its nodes should be smaller than a certain threshold because otherwise, timing violations and logic failures may occur. A node is said to be *safe* when its voltage drop meets the corresponding threshold condition. Let V_{th} be the vector of all the threshold values which are typically user-specified, and assume that $V_{th} > 0$ to avoid trivial cases. The *time-to-failure* of a grid is the earliest time t for which a *voltage violation* occurs, i.e. $v(t) \leq V_{th}$ is no longer true.

Notice that the time-to-failure of the grid, denoted as TTF, is a random variable, because the stress evolution is highly dependent on the effective atomic diffusivity of the branches in the grid, which are random variables. Assigning a diffusivity value to each branch in the grid defines a *grid sample* $\mathcal{G}^{(i)}$. The time-to-failure of $\mathcal{G}^{(i)}$ is a deterministic value and will be referred to as the TTF sample of $\mathcal{G}^{(i)}$ and denoted as $\text{TTF}^{(i)}$. To obtain a TTF sample of a given grid sample $\mathcal{G}^{(i)}$, the authors construct a set of LTI systems (each corresponding to an interconnect tree) as in (4), with initial thermal stress. Each of these systems are then solved numerically (simulated) to determine the earliest void nucleation among all trees. The simulation is then interrupted, a void growth model and a resistance model are applied, the impact on voltage drop is found. Based on this, the grid is either declared failed or the next round of simulation is restarted to find the next nucleation time, and this process is repeated until a grid failure is found. This flow is part of an overall Monte Carlo (MC) loop that accounts for the randomness in the lines and ultimately provides the grid *mean time-to-failure* (MTF) as the average

$$\text{MTF} \approx \frac{1}{N_{mc}} \sum_{i=1}^{N_{mc}} \text{TTF}^{(i)} \quad (5)$$

where N_{mc} is the number of TTF samples required to satisfy a user-specified error tolerance.

III. PROBLEM DEFINITION AND NOTATION

If the grid's MTF is less than the target lifetime, the grid design is said to have an *EM-lifetime violation* and needs to be fixed. There are several ways to fix an EM-lifetime violation. In this paper, we focus on fixing an EM-lifetime violation

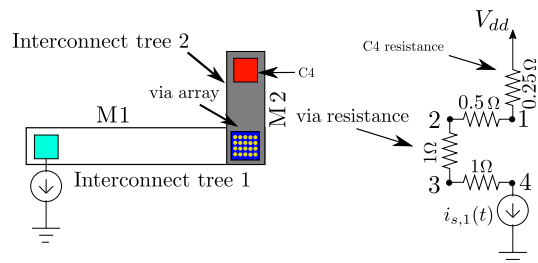


Fig. 1. A simple example of a power grid with 2 trees.

that is discovered close to design sign off by widening the metal lines of the grid. Intuitively, this would increase the conductivity of the grid which increases the TTF⁽ⁱ⁾ for every grid sample $\mathcal{G}^{(i)}$ and, in turn, increases the MTF of the grid. We aim to solve the following problem: given a power grid, and the effective currents drawn by the underlying logic circuitry, and a target lifetime, we will resize the interconnect trees on the various layers in order to satisfy the target lifetime, by making minimal changes to the metal area.

In our framework, we assume that the width of each interconnect tree k can be scaled by a factor s_k , so that the conductance of each metal branch within that tree is multiplied by s_k . Furthermore, because a via's resistance usually corresponds to the equivalent resistance of a via array of the same width as the metal line, as shown in Fig. 1, then we assume that the conductance of a via connected between interconnect trees j and k is a linear function of the overlapping area between trees j and k , i.e. the via's conductance is a linear function of the product $s_j s_k$. We refer to the grid before scaling any of its interconnect trees, i.e. the grid where the width of each interconnect tree is the width initially set by the designer, as the *original grid*. Furthermore, let $s = [s_1 \dots s_{n_t}]^T$ be an $n_t \times 1$ vector of scaling factors. Thus, the original grid corresponds to $s = \mathbf{1}$, where $\mathbf{1}$ is a vector of all 1 entries, whose size will be clear from the context.

In Fig. 1, we show an example of a simple power grid with two interconnect trees each of which is a single branch, so that $s = [s_1 \ s_2]^T$. Suppose that the resistance values shown in the figure correspond to the original grid, i.e. for $s = [1 \ 1]^T$. Notice that, if $s = [2 \ 2]^T$, then both interconnect trees would have twice their original widths, so that the resistance between nodes 3 and 4 would be 0.5Ω and its corresponding conductance would be $2\Omega^{-1}$. Furthermore, the via resistance (connected between nodes 2 and 3) would be 0.25Ω and its corresponding conductance would be $4\Omega^{-1}$. Thus, for this simple example, the conductance matrix [11] can be expressed in terms of s as follows:

$$G(s) = \begin{bmatrix} 4 + 2s_2 & -2s_2 & 0 & 0 \\ -2s_2 & 2s_2 + s_1 s_2 & -s_1 s_2 & 0 \\ 0 & -s_1 s_2 & s_1 s_2 + s_1 & -s_1 \\ 0 & 0 & -s_1 & s_1 \end{bmatrix} \quad (6)$$

Clearly, in general, the conductance matrix is a function of s . Again, because it is standard to model the effective atomic diffusivity of each branch in the grid as a random variable,

different grid samples will experience different sequences of void nucleations. Hence, for a grid sample $\mathcal{G}^{(i)}$, the conductance matrix at time t and scaling factors s will be denoted as $G^{(i)}(t, s)$. Here, and throughout the paper, the superscript (i) will be used to identify a grid sample.

To determine $G^{(i)}(t, s)$ at a specific time t and for a specific scaling factors s , we start with the conductance matrix where the line conductivities are scaled by s . Then, we construct the LTI system in (4) for each tree. As voids nucleate, the conductance matrix as well as the LTI systems are updated, as described in Section II-D, until we reach time t . The conductance matrix obtained at time t is $G^{(i)}(t, s)$.

The voltage drop at time t and under scaling factors s is expressed as $v^{(i)}(t, s)$ and can be obtained by solving the following linear system $G^{(i)}(t, s)v^{(i)}(t, s) = u$. The TTF sample corresponding to $\mathcal{G}^{(i)}$ is also a function of s , which we will denote as $\text{TTF}^{(i)}(s)$. Thus, the MTF of the grid is a function of s , in fact a *nonlinear* function of s , which we will denote as $\text{MTF}(s)$. In this paper, we aim to find an s such that:

$$\text{MTF}(s) \geq T^* \quad (7)$$

where T^* is a user-specified target lifetime.

IV. PROPOSED APPROACH

There are certain design considerations that impose constraints on s , such as the minimum spacing between metal lines and the maximum metal area usage. We will see in Section V that these design constraints can be represented as a set of linear constraints on s which define a *feasible space* for s , denoted as \mathcal{S} , but for now it is useful to note a key requirement that will be useful in this section, namely $s \geq \mathbb{1}$, which means that we will never resize a tree to below its original width.

In order to fix an EM-lifetime violation, one should search for an $s \in \mathcal{S}$ such that $\text{MTF}(s) \geq T^*$. This is difficult because for one thing, $\text{MTF}(s)$ is an implicit nonlinear function of s . One way is to iteratively increase the value of s_k for every interconnect tree k that has a junction failure, or at which a voltage violation occurred, while satisfying $s \in \mathcal{S}$, and determine the corresponding $\text{MTF}(s)$, until $\text{MTF}(s) \geq T^*$ is satisfied. This approach, however, performs localized (greedy) improvements to the grid design without factoring in the response of the whole grid and, as such, fixing the problem in a specific area may simply move the problem to another area of the design. Furthermore, this approach does not provide any guidance on how much a tree needs to be widened. It is left to the user to decide, which, in many cases, may result in over-design of the grid. In fact, this trial-and-error approach may iterate forever as it searches “blindly” for a safe point s in an intractable space of possible values. In this section, we describe our approach to fix an EM-lifetime violation using Successive Linear Programming (SLP) [12], an iterative nonlinear optimization method. As we will see, our approach provides a systematic way to fix an EM-lifetime violation by factoring in both the grid design, both locally and globally, and the randomness in EM degradation.

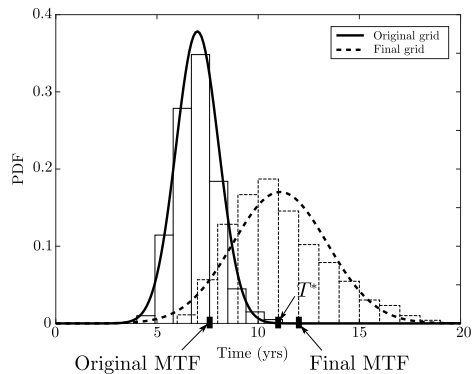


Fig. 2. TTF distributions for original and resized grids.

A. Overview

Starting from the original grid, one would like to find an s that increases the $\text{MTF}(s)$. Strictly speaking, it may be enough to only increase the TTFs of some, but not all, sample grids. However, when we find a new s to “fix” one grid sample, this will also affect the TTFs of all other samples, so they have to be checked or updated as well. For this reason, and to get full confidence that the MTF will be improved, we search for an s that increases $\text{TTF}^{(i)}(s)$ for every grid sample $\mathcal{G}^{(i)}$ which, in turn, would increase $\text{MTF}(s)$. This can be done by searching for an s that reduces the voltage drop of every grid sample, by enough to achieve $v^{(i)}(t, s) \leq V_{th}, \forall t \leq T^*, \forall i$. As an example of what is possible, Fig. 2 shows the TTF distribution of an original grid and the TTF distribution of the grid after being resized using our approach.

Ideally, one would like to find the shortest distance to a “safe point” in the space of s . In other words, one would like to find an s that ensures the voltage drop at the nodes of every grid sample remain within the threshold value until time $t = T^*$ while requiring minimal increase in the total metal area. Mathematically, this can be formulated as the following nonlinear optimization problem:

$$\begin{aligned} &\text{Minimize} && a^T s \\ &\text{s.t.} && v^{(i)}(t, s) \leq V_{th}, \forall t \leq T^*, \forall i \\ &&& s \in \mathcal{S} \end{aligned} \quad (8)$$

where $a = [w_1 l_1 \dots w_{n_t} l_{n_t}]^T$ is an $n_t \times 1$ vector which consists of the metal areas of each interconnect tree, and \mathcal{S} is the linearly bound domain given in (38) which represents the feasible space of s based on the design rules. Note that, for any $s \in \mathcal{S}$, we have $s \geq \mathbb{1}$ so that the above optimization problem only widens the interconnect trees relative to their original size. The above nonlinear optimization problem is solved by means of an iterative *stepping strategy*, which we will implement using a linearization of the voltage drop around the latest solution point - this leads to a *linear program* (LP) formulation in every iteration. The following provides a high-

level description of the proposed stepping strategy:

while an EM-lifetime violation exists, **do**

1. Find the MTF at the latest solution point, and quit if the MTF is within specification.
2. Linearize the voltage drop of all grid samples around that point.
3. Determine a descent direction that reduces the voltage drop of all grid samples (solve an LP).
4. Update the solution point based on a step in the descent direction.

end

This stepping strategy benefits from the result of the following lemma, which provides a *descent direction* for the voltage drop. The lemma applies to any grid sample $\mathcal{G}^{(i)}$ and so we will drop the superscript (i) to simplify the notation. Under the standard assumption that the original undamaged grid (i.e. before any void nucleation) is connected and has at least one voltage source, the conductance matrix of the original grid $G(0, \mathbb{1})$ is *non-singular* [11] so that $G^{-1}(0, \mathbb{1})$ exists. As voids nucleate over time, the conductance of a branch does not quite go to zero, so that the grid remains connected and its conductance matrix remains non-singular. Furthermore, widening the grid metal lines also keeps the grid connected, so $G^{-1}(t, s)$ exists for any $t \geq 0$ and $s \geq \mathbb{1}$. Here is the lemma.

Lemma 1. *For any $s \geq \mathbb{1}$, we have:*

$$\frac{\partial v(t, s)}{\partial s_k} = -G^{-1}(t, s) \frac{\partial G(t, s)}{\partial s_k} G^{-1}(t, s) u \quad (9)$$

Proof: For any $s \geq \mathbb{1}$, we can write:

$$v(t, s) = G^{-1}(t, s) u \quad (10)$$

so that:

$$\frac{\partial v(t, s)}{\partial s_k} = \frac{\partial G^{-1}(t, s)}{\partial s_k} u \quad (11)$$

where we used the fact that u is independent of s . Starting with:

$$G(t, s) G^{-1}(t, s) = I \quad (12)$$

where I is the $n \times n$ identity matrix, we can differentiate both sides with respect to s_k to get:

$$G(t, s) \frac{\partial G^{-1}(t, s)}{\partial s_k} + \frac{\partial G(t, s)}{\partial s_k} G^{-1}(t, s) = 0 \quad (13)$$

or equivalently,

$$\frac{\partial G^{-1}(t, s)}{\partial s_k} = -G^{-1}(t, s) \frac{\partial G(t, s)}{\partial s_k} G^{-1}(t, s) \quad (14)$$

Substituting (14) in (11), we get:

$$\frac{\partial v(t, s)}{\partial s_k} = -G^{-1}(t, s) \frac{\partial G(t, s)}{\partial s_k} G^{-1}(t, s) u \quad (15)$$

and the proof is complete. \blacksquare

B. Stepping Strategy

This section introduces a linearization of the voltage drop which allows us to provide a linearization of (8) into an LP around the latest solution point. Given a grid sample $\mathcal{G}^{(i)}$, the first-order Taylor's expansion of $v^{(i)}(t, s)$ in the neighborhood of $s = s^{(r)}$, denoted as $\bar{v}^{(i)}(t, s)$, is:

$$\bar{v}^{(i)}(t, s) \triangleq v^{(i)}(t, s^{(r)}) + J^{(i)}(t, s^{(r)})(s - s^{(r)}) \quad (16)$$

where $J^{(i)}(t, s^{(r)})$ is the $n \times n_t$ *Jacobian* matrix of $v^{(i)}(t, s^{(r)})$, defined as follows:

$$J^{(i)}(t, s^{(r)}) \triangleq \left[\frac{\partial v^{(i)}(t, s^{(r)})}{\partial s_1} \dots \frac{\partial v^{(i)}(t, s^{(r)})}{\partial s_{n_t}} \right] \quad (17)$$

The columns of $J^{(i)}(t, s^{(r)})$ can be computed using Lemma 1. With this, at each step r where an EM-lifetime violation exists, we can construct the linearized voltage drop $\bar{v}^{(i)}$ around the latest solution point $s^{(r)}$, as in (16), to determine a descent direction that guarantees the linearized voltage drop of all grid samples remain within specifications until time T^* , while requiring minimal increase in metal area. This can be formulated as the following LP:

$$\begin{aligned} & \text{Minimize} && a^T s \\ & \text{s.t.} && \bar{v}^{(i)}(t, s) \leq V_{th}, \forall t \leq T^*, \forall i \\ & && s \in \mathcal{S} \end{aligned} \quad (18)$$

Clearly, the number of constraints in (18) is intractable because of the continuous t domain. In the following, we will make a simplifying assumption that simplifies the constraints space of (18) and allows us to get rid of the $\forall t \leq T^*$ requirement. This assumption is not really a limitation to our work. It is used to guide our optimization but it does not invalidate the result if the assumption does not hold, because in our flow we always check the MTF before we exit. But the assumption provides significant speed-up when it holds, and we will show an empirical result that confirms the validity of this assumption in the majority of cases.

Assumption 1. (Monotonicity) *For a grid sample $\mathcal{G}^{(i)}$, the voltage drop $v^{(i)}(t, s)$ is a monotonically increasing function with respect to time, i.e. $v^{(i)}(t_1, s) \leq v^{(i)}(t_2, s)$, $\forall t_1, t_2$, such that $0 \leq t_1 \leq t_2$.*

In other words, the creation of voids always causes the voltage drop to increase. This intuitively makes sense, because void nucleation causes a resistance increase, but an increase in branch resistance does not *necessarily* lead to a voltage drop increase. Generally, however, the assumption holds most of the time. In fact, empirical results for a 37k-node grid show that the assumption holds in $\approx 90\%$ of the cases. Based on this assumption, for any $s \in \mathcal{S}$, we have:

$$v^{(i)}(t, s) \leq v^{(i)}(T^*, s) \quad (19)$$

for any $t \leq T^*$. As a result, it is enough to search for an $s \in \mathcal{S}$ that decreases the voltage drop at time T^* . With this, we can simplify (18) into:

$$\begin{aligned} & \text{Minimize} && a^T s \\ & \text{s.t.} && \bar{v}^{(i)}(T^*, s) \leq V_{th}, \forall i \\ & && s \in \mathcal{S} \end{aligned} \quad (20)$$

Note that the LP in (20) still has a large number of constraints (on the order of $N_{mc} \times n$, where N_{mc} is the number of grid samples) and, thus, solving (20) is computationally expensive. However, one does not have to solve an LP that includes *all* the constraints at once, because we have found that fixing one grid sample will often automatically fix many others. Instead, we start by solving the LP (20) using the EM constraints of a single grid sample (e.g., the grid sample with the smallest TTF). The solution of this LP, denoted as \hat{s} , is then used to check whether $\bar{v}^{(i)}(T^*, \hat{s}) \leq V_{th}$ is satisfied for other grid samples. If not, we add the constraints of the most violated grid sample (the one for which $(\bar{v}_j^{(i)} - V_{th,j})$ is largest); we solve the resulting (larger) LP, and repeat until the constraints of all grid samples are satisfied. It is possible for this *incremental approach* (as we will refer to it) to become more expensive than solving the original LP (20), however, our experience is that this approach is much faster in general.

C. Step-size Selection

Let \hat{s} denote the vector that solves the above LP (20), found using our incremental approach. Because the LP is a linearization of the original nonlinear problem (8) around the latest solution, taking a large step-size towards \hat{s} may be *overkill* for two reasons. First, the farther we go from the current solution, the less accurate the linearization becomes. Second, taking a large step may result in an MTF that is much larger than the target lifetime, leading to over-design of the grid. One typical way [13] of taking a partial step in a specific direction is to enforce a fraction of the full-step that would respect some user criterion, e.g.

$$s^{(r+1)} = s^{(r)} + \lambda^{(r)}(\hat{s} - s^{(r)}) \quad (21)$$

where the unitless $\lambda^{(r)} \in [0, 1]$ represents the fractional step-size at iteration r that is chosen based on user criteria. Note that $\lambda^{(r)} \geq 0$ because we should move in the direction found by the LP, and $\lambda^{(r)} \leq 1$ because there is no reason to take a larger step than the one returned by the LP.

The following lemma establishes that if we start with an original grid that satisfies the design constraints in Section V, i.e. $s^{(0)} \in \mathcal{S}$, then the “scaled” grid, at every step r , also satisfies the design constraints, i.e. $s^{(r)} \in \mathcal{S}$. Thus, the final grid satisfies the design constraints.

Lemma 2. *Given $s^{(0)} \in \mathcal{S}$, and with reference to (21), with $\lambda^{(r)} \in [0, 1], \forall r \geq 0$, it follows that $s^{(r)} \in \mathcal{S}, \forall r \geq 0$.*

Proof: The proof is by induction. Notice that $s^{(0)} \in \mathcal{S}$, due to the statement of the lemma. In the following, we will show that, given an $s^{(r)} \in \mathcal{S}$, we have $s^{(r+1)} \in \mathcal{S}$, which would complete the proof. Because $s^{(r)} \in \mathcal{S}$, and $\hat{s} \in \mathcal{S}$ due to (20), then $s^{(r)} \geq \mathbf{1}$ and $\hat{s} \geq \mathbf{1}$. And due to $\lambda^{(r)} \in [0, 1]$, we have $\lambda^{(r)} \geq 0$ and $(1 - \lambda^{(r)}) \geq 0$, so that $\lambda^{(r)}\hat{s} \geq \lambda^{(r)}\mathbf{1}$ and $(1 - \lambda^{(r)})s^{(r)} \geq (1 - \lambda^{(r)})\mathbf{1}$. This leads to

$$s^{(r+1)} = s^{(r)} + \lambda^{(r)}(\hat{s} - s^{(r)}) = \lambda^{(r)}\hat{s} + (1 - \lambda^{(r)})s^{(r)} \quad (22)$$

$$\geq \lambda^{(r)}\mathbf{1} + (1 - \lambda^{(r)})\mathbf{1} = \mathbf{1} \quad (23)$$

so that $s^{(r+1)} \geq \mathbf{1}$. Likewise, because $s^{(r)} \in \mathcal{S}$ and $\hat{s} \in \mathcal{S}$, due to (20), then

$$d_{lb} \leq Ds^{(r)} \leq d_{ub} \quad (24)$$

and

$$d_{lb} \leq D\hat{s} \leq d_{ub} \quad (25)$$

Due to $\lambda^{(r)} \in [0, 1]$, we have $\lambda^{(r)} \geq 0$ and $(1 - \lambda^{(r)}) \geq 0$, so that multiplying (24) with $(1 - \lambda^{(r)})$ gives

$$(1 - \lambda^{(r)})d_{lb} \leq (1 - \lambda^{(r)})Ds^{(r)} \leq (1 - \lambda^{(r)})d_{ub} \quad (26)$$

and multiplying (25) with $\lambda^{(r)}$ gives

$$\lambda^{(r)}d_{lb} \leq \lambda^{(r)}D\hat{s} \leq \lambda^{(r)}d_{ub} \quad (27)$$

Adding (26) and (27) gives:

$$d_{lb} \leq D \left((1 - \lambda^{(r)})s^{(r)} + \lambda^{(r)}\hat{s} \right) \leq d_{ub} \quad (28)$$

or equivalently,

$$d_{lb} \leq Ds^{(r+1)} \leq d_{ub} \quad (29)$$

so that $s^{(r+1)} \in \mathcal{S}$, and the proof is complete. \blacksquare

In our work, we choose $\lambda^{(r)}$ such that the incremental increase in the total metal area is within a user-specified value $\delta > 0$, i.e.

$$\frac{a^T s^{(r+1)} - a^T s^{(r)}}{a^T \hat{s}^{(r)}} \leq \delta \quad (30)$$

Note that the left-hand side of the above inequality can be negative, in which case, step $r + 1$ tries to reduce an unnecessary additional metal area that was introduced at step r . In the following, Lemma 3 provides a *necessary and sufficient* condition on $\lambda^{(r)}$ so that (30) is satisfied. In fact, Lemma 3, combined with the fact that $\lambda^{(r)} \in [0, 1]$, provides a range of feasible values of $\lambda^{(r)}$ as follows:

$$0 \leq \lambda^{(r)} \leq \min(\gamma^{(r)}, 1) \quad (31)$$

where the scalar $\gamma^{(r)}$ is defined below:

$$\gamma^{(r)} \triangleq \begin{cases} \frac{\delta a^T \hat{s}^{(r)}}{a^T \hat{s} - a^T s^{(r)}} & \text{if } a^T \hat{s} > a^T s^{(r)}, \\ 1 & \text{otherwise.} \end{cases} \quad (32)$$

Lemma 3. *For any $\lambda^{(r)} \in [0, 1]$, then $\gamma^{(r)} > 0$ and, with reference to (21)-(30), we have:*

$$\frac{a^T s^{(r+1)} - a^T s^{(r)}}{a^T s^{(r)}} \leq \delta \iff \lambda^{(r)} \leq \gamma^{(r)} \quad (33)$$

Proof: First, we will show that $\gamma^{(r)} > 0$. Notice that if $a^T \hat{s} \leq a^T s^{(r)}$, then $\gamma^{(r)} = 1$, due to (32), so that $\gamma^{(r)} > 0$. Otherwise, if $a^T \hat{s} > a^T s^{(r)}$, then $a^T \hat{s} - a^T s^{(r)} > 0$ which, combined with $\delta > 0$ and $a^T s^{(r)} > 0$, because $a > 0$ and $s^{(r)} > 0$, gives $\gamma^{(r)} > 0$. Next, we will prove (33). Notice that, because $a^T s^{(r)} > 0$, then

$$\frac{a^T s^{(r+1)} - a^T s^{(r)}}{a^T s^{(r)}} \leq \delta \quad (34)$$

$$\iff a^T s^{(r+1)} - a^T s^{(r)} \leq \delta a^T s^{(r)} \quad (35)$$

$$\iff a^T \left(s^{(r)} + \lambda^{(r)} (\hat{s} - s^{(r)}) \right) - a^T s^{(r)} \leq \delta a^T s^{(r)} \quad (36)$$

$$\iff \lambda^{(r)} \left(a^T \hat{s} - a^T s^{(r)} \right) \leq \delta a^T s^{(r)} \quad (37)$$

We will now show that (37) $\iff \lambda^{(r)} \leq \gamma^{(r)}$ by separately considering the two cases: $a^T \hat{s} > a^T s^{(r)}$ and $a^T \hat{s} \leq a^T s^{(r)}$. Considering first the case $a^T \hat{s} > a^T s^{(r)}$, we have by definition $\gamma^{(r)} = \delta a^T s^{(r)} / (a^T \hat{s} - a^T s^{(r)})$ and, with $a^T \hat{s} - a^T s^{(r)} > 0$, then (37) $\iff \lambda^{(r)} \leq \delta a^T s^{(r)} / (a^T \hat{s} - a^T s^{(r)}) = \gamma^{(r)}$. Considering now the case $a^T \hat{s} \leq a^T s^{(r)}$, recall that, if p and q are two statements then $p \iff q$ is true if and only if the logical statement $(pq + \bar{p}\bar{q})$ is always true (where $+$ denotes the boolean OR operator), so that p and q are always either both true or both false. In this case, with $a^T \hat{s} \leq a^T s^{(r)}$, then $\gamma^{(r)} = 1$ so that $\lambda^{(r)} \leq \gamma^{(r)}$ and, with $a^T \hat{s} - a^T s^{(r)} \leq 0$, then $\lambda^{(r)}(a^T \hat{s} - a^T s^{(r)}) \leq 0 \leq \delta a^T s^{(r)}$. With both statements always true, it follows that (37) $\iff \lambda^{(r)} \leq \gamma^{(r)}$ in this case as well, and the proof is complete. ■

Due to Lemma 3, it will always be possible to choose a $\lambda^{(r)}$ in the feasible range (31). So, in every step, we will start with $\lambda^{(r)}$ as the largest value in this range because it already satisfies all the requirements and there is no reason to take a smaller step, i.e. $\lambda^{(r)} = \min(\gamma^{(r)}, 1)$. However, it is possible that taking such a step from the latest solution point $s^{(r)}$ would *overshoot* the target lifetime beyond a certain acceptable margin Δ , i.e. $\text{MTF}(s^{(r)} + \lambda^{(r)}(\hat{s} - s^{(r)})) > T^* + \Delta$. In this case, we have “bracketed” a solution, because $\text{MTF}(s^{(r)}) < T^*$ and $\text{MTF}(s^{(r)} + \lambda^{(r)}(\hat{s} - s^{(r)})) > T^*$, and so, we will perform a line search to find a $0 \leq \lambda^{(r)} \leq \min(\gamma^{(r)}, 1)$ (referred to as a bracketed region) such that $\text{MTF}(s^{(r)} + \lambda^{(r)}(\hat{s} - s^{(r)})) \in [T^*, T^* + \Delta]$. The process of finding such a $\lambda^{(r)}$ can be posed as a *root finding* problem because basically we are searching for a root to the nonlinear function $f(\lambda^{(r)}) \triangleq \text{MTF}(s^{(r)} + \lambda^{(r)}(\hat{s} - s^{(r)})) - T^*$. There are many common methods in the literature to find a root in a bracketed region, such as bisection method, false position method, secant method, etc. The reader is referred to [14] for details on such methods. Some of these methods preserve the “bracketing of a root” property while tracking down a root, such as the bisection and the false position methods, while other methods do not preserve this property, such as the secant method. In our work, we use a method that preserves this important property, namely the false position method, as it guarantees convergence to a root.

V. DESIGN RULES

Several design rules must be considered when resizing the metal lines of the grid. These rules are constraints on how the grid may be modified, e.g., which interconnect trees can be scaled, and by how much. In this section, we discuss some of these design rules and show how they boil down to linear constraints on s which can be compactly represented as $d_{lb} \leq Ds \leq d_{ub}$. Every interconnect tree in the grid has a minimum width limit that it should satisfy, usually derived based on the technology node and other design considerations. We assume that the width of each tree in the original grid is already set to its minimum, so that trees can only be scaled up relative to their original width, which defines a linear constraint on s , namely $s \geq \mathbb{1}$. As mentioned earlier, these linear constraints will define a *feasible space* for s , denoted as \mathcal{S} , i.e.

$$\mathcal{S} \triangleq \{s \in \mathbb{R}^{n_t} : s \geq \mathbb{1}, d_{lb} \leq Ds \leq d_{ub}\} \quad (38)$$

All the constraints discussed below, along with any additional similar constraint specified by the user, can be handled by our approach as long as they can be represented as linear inequalities on s .

A. Maximum Metal Area Usage

The design team might have specifications on the maximum allowable increase (relative to the original grid) in the total metal area. This specification can be expressed as

$$\frac{(a^T s)}{(a^T \mathbb{1})} \leq \beta \quad (39)$$

where $\beta > 0$ is the user-specified maximum allowable increase in the total metal area, or equivalently, $a^T s \leq \beta a^T \mathbb{1}$, because $a^T \mathbb{1} > 0$.

B. Minimum Spacing

Typically, each metal layer consists of a set of alternating supply and ground interconnect trees. Of course, supply and ground interconnect trees within the same layer must not overlap. In fact, there is a minimum allowable spacing between supply and ground interconnect trees. Suppose that a supply interconnect tree j is adjacent to a ground interconnect tree within metal layer p with original spacing of κ between the two trees. Notice that after the interconnect tree j of original width w_j is scaled by s_j , it will have a width of $s_j w_j$, so that the tree width will increase by $(s_j w_j - w_j)/2$ at each of its two sides. Thus, under scaling factors s , the space between the two trees is $\kappa - \frac{s_j w_j - w_j}{2}$. Let $\hat{\kappa}_p$ be the minimum allowable spacing between a supply and ground interconnect tree in metal layer p , then s should satisfy $\kappa - \frac{s_j w_j - w_j}{2} \geq \hat{\kappa}_p$, or equivalently, $s_j w_j \leq 2(\kappa - \hat{\kappa}_p) + w_j$.

VI. EXPERIMENTAL RESULTS

The approach discussed in Section IV has been implemented in C++. We verified our approach using two types of test grids: IBM power grids [15] and our own (internal) grids. The internal grids were generated based on user specifications, including grid dimensions, metal layers, number of blocks, number of metal layers in the global grid, pitch and width per layer, and C4 and current source distributions. The technology specifications were consistent with 1V 45 nm CMOS technology. The grids named PG1-PG5 are internal grids. All results were obtained using a hyperthreaded 12-core 3GHz Linux machine with 128GB of RAM. The optimizations were performed using the MOSEK optimization package [16]. All the linear systems are solved using Cholmod [17], except for the voltage drop updates required to compute the MTF which were done using the Preconditioned Conjugate Gradient (PCG) method described in [18]. In our implementation, we use *Pthread* to parallelize the MC simulations and to take advantage of the 12-core machine. All power grids are assumed to have a target MTF of 12 years, an acceptable overshoot margin of 1 year (i.e. $\Delta = 1$ year), and the incremental increase in the total metal area between two consecutive steps is required to be within 0.2% (i.e. the δ parameter in (30) is 0.2%).

TABLE I
SUMMARY OF RESULTS

Power Grid			Proposed Approach						
Name	Nodes	Trees	Original MTF	Final MTF	Metal area increase	Num of scaled trees	Num of steps	Num of MTF evaluations	Total time
ibmpg1	6K	709	7.1 yrs	12.6 yrs	0.688%	3	4	6	11.6 min
ibmpg2	62K	462	9.6 yrs	12.3 yrs	0.600%	15	3	4	13.0 min
ibmpg4	475K	9.6K	9.8 yrs	12.9 yrs	0.061%	8	1	4	44.3 min
ibmpg6	404K	10.2K	10.2 yrs	12.8 yrs	0.010%	2	1	4	3.9 hrs
ibmpgnew1	316K	19.5K	9.8 yrs	12.7 yrs	0.006%	2	1	3	34.6 min
PG1	37K	0.7K	9.3 yrs	12.3 yrs	0.134%	27	1	3	1.0 min
PG2	560K	2.6K	10.0 yrs	12.9 yrs	0.010%	10	1	8	25.1 min
PG3	1.2M	5.6K	10.5 yrs	12.2 yrs	0.021%	14	1	4	1.2 hrs
PG4	2.6M	12.2K	6.4 yrs	12.2 yrs	0.007%	6	1	8	2.9 hrs
PG5	4.1M	12.6K	8.8 yrs	13.0 yrs	0.018%	4	1	4	1.9 hrs

Table I summarizes the results. In columns 4–5, we show the MTF of the original grid and the MTF of the grid after widening the interconnect trees, respectively. Furthermore, columns 6–7 show the percentage increase in the total metal area of the grid as well as the number of trees that were scaled to fix the EM-lifetime violation. For example, on a 1.2 million nodes grid, we were able to increase the MTF of the grid from 10.5 years to 12.2 years using 0.02% metal area increase and by scaling 14 interconnect trees. It is important to note that 7 out of the 14 scaled trees did not have either a void nucleation or a voltage drop violation. This shows that fixing an EM-lifetime violation might not be intuitively easy or obvious, and demonstrates the value of our optimization based approach that factors in the behavior of the whole grid.

The total runtime of our approach, i.e. the total wall clock time of the whole parallel Pthread implementation, including both the MTF evaluations and the incremental LP solving, is shown in column 10. Furthermore, columns 8–9 show the number of steps taken and the number of MTF evaluations required to fix an EM-lifetime violation, respectively. For example, on a 1.2 million nodes grid, our approach fixed the violation using a single step and 4 MTF evaluations, which took 1.2 hrs. It is important to note that about 70% of the total runtime to fix any of the grids in Table I was spent on the MTF evaluations.

VII. CONCLUSION

We proposed a power grid fixing scheme that fixes an EM-lifetime violation. Subject to design rules and other design considerations, the proposed approach iteratively improves the EM-lifetime of the grid, by factoring in both the grid design and the randomness in EM degradation, to meet a target EM-lifetime.

REFERENCES

- [1] J. R. Black, "Electromigration failure modes in aluminum metallization for semiconductor devices," *Proceedings of the IEEE*, vol. 57, no. 9, pp. 1587–1594, 1969.
- [2] S. Chatterjee, V. Sukharev, and F. N. Najm, "Power grid electromigration checking using physics-based models," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 7, pp. 1317–1330, Feb 2017.
- [3] J. Choy, V. Sukharev, S. Chatterjee, F. N. Najm, A. Kteyan, and S. Moreau, "Finite-difference methodology for full-chip electromigration analysis applied to 3D IC test structure: Simulation vs. experiment," in *Int. Conf. on Simulation of Semiconductor Processes and Devices*, Sept 2017, pp. 41–44.
- [4] S. Chowdhury and M. A. Breuer, "Optimum design of IC power/ground nets subject to reliability constraints," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 787–796, July 1988.
- [5] R. Dutta and M. Marek-Sadowska, "Automatic sizing of power/ground (p/g) networks in VLSI," in *Design Automation Conf.*, 1989, pp. 783–786.
- [6] X. Tan, C. Shi, D. Lungeanu, J. Lee, and L. Yuan, "Reliability-constrained area optimization of VLSI power/ground networks via sequence of linear programmings," in *Design Automation Conference*, 1999, pp. 78–83.
- [7] S. X. Tan and C. R. Shi, "Efficient very large scale integration power/ground network sizing based on equivalent circuit modeling," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 277–284, March 2003.
- [8] L. Ting, J. S. May, W. R. Hunter, and J. W. McPherson, "AC electromigration characterization and modeling of multilayered interconnections," in *International Reliability Physics Symposium (IRPS)*, 1993, pp. 311–316.
- [9] M. A. Korhonen, P. Borgesen, K. N. Tu, and C. Li, "Stress evolution due to electromigration in confined metal lines," *J. of App. Phys.*, vol. 73, no. 8, pp. 3790–3799, 1993.
- [10] J. Lloyd and J. Kitchin, "The electromigration failure distribution: The fine-line case," *Journal of Applied Physics*, vol. 69, no. 4, pp. 2117–2127, Feb 1991.
- [11] F. N. Najm, *Circuit Simulation*. Hoboken, NJ: John Wiley & Sons, Inc., 2010.
- [12] F. Palacios-Gomez, L. Lasdon, and M. Engquist, "Nonlinear optimization by successive linear programming," *Management Science*, vol. 28, no. 10, pp. 1106–1120, 1982.
- [13] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Philadelphia, PA: SIAM, 1996.
- [14] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2007.
- [15] S. R. Nassif, "Power grid analysis benchmarks," in *ASPDAC*, Jan. 21–24 2008, pp. 376–381.
- [16] (2015) MOSEK optimization software. [Online]. Available: www.mosek.com
- [17] Y. Chen and et al., "Algorithm 887: CHOLMOD, supernodal sparse cholesky factorization and update/downdate," *Trans. on Math. Soft.*, vol. 35, no. 3, pp. 22:1–22:14, 2008.
- [18] S. Chatterjee, V. Sukharev, and F. N. Najm, "Fast physics-based electromigration assessment by efficient solution of linear time-invariant (LTI) systems," in *Int. Conf. on Computer-Aided Design*, 2017, pp. 659–666.