

A Realistic Early-Stage Power Grid Verification Algorithm Based on Hierarchical Constraints

Yuanzhe Wang, Xiang Hu, Chung-Kuan Cheng, *Fellow, IEEE*, Grantham K. H. Pang, *Senior Member, IEEE*, and Ngai Wong, *Member, IEEE*

Abstract—Power grid verification has become an indispensable step to guarantee a functional and robust chip design. Vectorless power grid verification methods, by solving linear programming (LP) problems under current constraints, enable worst-case voltage drop predictions at an early stage of design when the specific waveforms of current drains are unknown. In this paper, a novel power grid verification algorithm based on hierarchical constraints is proposed. By introducing novel power constraints, the proposed algorithm generates more realistic current patterns and provides less pessimistic voltage drop predictions. The model order reduction-based coefficient computation algorithm reduces the complexity of formulating the LP problems from being proportional to steps to being independent of steps. Utilizing the special hierarchical constraint structure, the submodular polyhedron greedy algorithm dramatically reduces the complexity of solving the LP problems from over $O(k_m^3)$ to roughly $O(k_m \log k_m)$, where k_m is the number of variables. Numerical results have shown that the proposed algorithm provides less pessimistic voltage drop prediction while at the same time achieves dramatic speedup.

Index Terms—Hierarchical constraints, model order reduction, submodular polyhedron, vectorless power grid verification, worst-case voltage drop.

I. INTRODUCTION

POWER grid verification is the procedure of verifying that the voltage drops on power grids are within an acceptable range. Due to the ever increasing voltage drops on power grids and decreasing noise margins of logic gates, power grid verification has become an indispensable step in chip design to avoid logic errors and to reduce gate delays. Although solving the IR or $L \frac{di}{dt}$ voltage drops is essentially equivalent to a DC or transient analysis, the extremely large size of power grid models renders traditional software like SPICE inefficient.

Manuscript received March 2, 2011; revised May 30, 2011; accepted August 29, 2011. Date of current version December 21, 2011. The work of C.-K. Cheng was supported by NSF CCF-1017864. This work was supported in part by the Hong Kong Research Grants Council, under Projects HKU 718509E and 718711E, and by the University Research Committee of the University of Hong Kong. This paper was recommended by Associate Editor S. Vrudhula. Y. Wang is with Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: yzwang@eee.hku.hk).

X. Hu and C.-K. Cheng are with the Department of Computer Science and Engineering, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: x2hu@ucsd.edu; ckcheng@ucsd.edu).

G. K. H. Pang and N. Wong are with the Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam, Hong Kong (e-mail: gpang@eee.hku.hk; nwong@eee.hku.hk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCAD.2011.2167328

Substantial work has been done in the past decades on efficient and application-specific algorithms for power grid verification [1]–[29].

Power grid verification methods can be roughly divided into two groups: simulation-based methods and vectorless methods. Simulation-based methods model the power grids as RC(L) networks while transistors, logic gates, and others as current sources. These methods accept as input the waveforms of current sources and generate as output the voltage drops on grid nodes [1]–[5], [7]–[9], [11], [20], [24]. Backward Euler or trapezoidal rules are employed to perform transient analysis. Differently, vectorless methods accept as input the constraints on current sources (instead of specific current waveforms), and generate maximum possible voltage drops on grid nodes [12]–[19], [21], [22]. Linear programming (LP), integer programming, or convex optimization problems are solved to obtain these worst-case predictions. Vectorless methods are preferred over simulation-based methods on the following occasions: 1) we want to predict voltage drops at an early stage of design when the specific current waveforms are unknown, and 2) there exist too many work modes or possible current waveforms thus it is complicated to determine which one results in the worst-case voltage drops.

Vectorless method is first proposed in [12]. Worst-case voltage drops are solved from LP problems under local and global current constraints. The current constraints are determined by experience or design requirements. Current waveforms in [12] are assumed to be static (essentially a DC analysis). In [15], the algorithm is extended to transient analysis. A geometrical algorithm is developed to solve the LP problems more efficiently in sacrifice of some accuracy. However, the current patterns in [15] are still assumed to be static. An efficient matrix inversion algorithm is proposed in [16], where small entries of the inverse (system) matrix are set to zero. This approximation reduces the CPU time for formulating the LP problems, yet introduces additional errors in voltage predictions. A dual algorithm is developed in [17] which achieves better efficiency by converting the LP problems to convex problems with fewer variables. However, the efficiency of the algorithm may decrease if the number of constraints is large. The deductions in [15]–[17] are based on the assumption that the system matrix is an M -matrix [30], thus the power grid models cannot contain inductors. The algorithm in [14] handles power grid models with inductors, yet is less efficient due to the increased number of variables. Besides, the constraints in [14] at different time

steps are entirely independent of each other, which is generally not true in practice. An impulse response-based algorithm is proposed in [18] which takes the transition time of current sources into consideration. However, the models investigated in [18] are basically lumped models (instead of resistor-capacitor-inductor (RCL) mesh models) of power grids.

In this paper, a more realistic early-stage power grid verification algorithm based on hierarchical current and power constraints is proposed. The main contributions are as follows.

- 1) Novel power constraints are developed, leading to more realistic current waveforms and less pessimistic voltage drop predictions. Besides, the proposed algorithm does not rely on M -matrix assumption and naturally handles general RCL models.
- 2) The constraints in the LP problems are designed to be hierarchical. Such special constraint structure enables a fast submodular polyhedron greedy algorithm (SPGA), which reduces the complexity of solving LP problems *dramatically* from over $O(k_m^3)$ to roughly $O(k_m \log k_m)$. Here k_m is the number of variables.
- 3) An efficient coefficient computation algorithm based on model order reduction (MOR) is proposed. Coefficient computation is a major step in the LP problem formulation. Compared with direct coefficient computation, the MOR-based algorithm is more efficient and its complexity is independent of the number of time steps. The MOR-based algorithm can further be modified to handle adaptive step length without increase of computation complexity.

This paper is an extension of our previous work [19]. Compared with [19], the new contributions of this paper are as follows.

- 1) MOR-based coefficient computation method is proposed, whose complexity is lower than that of direct method and is independent of total number of steps. Thus, it enables the usage of large number of steps and adaptive step lengths. The error introduced by MOR is negligible.
- 2) The feasible set of the LP problem with hierarchical constraints is proved to be a submodular polyhedron, which enables the usage of SPGA. The fact that the LP problems in this paper are special cases of submodular optimization problems not only raises the proposed algorithm to a higher theoretical level but also provides the possibility of further generalizing the constraint structure.
- 3) The complexity of the proposed algorithm is detailed. A wide spectrum of experimental results are presented to verify the proposed algorithm.

The remainder of this paper is organized as follows. Background information is introduced in Section II. Hierarchical constraints and LP problem formulation are proposed in Section III. The MOR-based coefficient computation algorithm and SPGA are proposed in Section IV. The complexity of the proposed algorithm is analyzed in Section IV. Experiments on industrial power grid examples are presented in Section V. Finally, conclusions are drawn in Section VI.

II. BACKGROUND

A. Power Grid Model

A typical 3-D power grid consists of several layers, with each layer containing either horizontal or vertical metal wires. Wires in different layers are connected to each other at the intersection points by vias. The power grid can be modeled as an RCL network (which may or may not be regular) for simulation. External power sources (connected to the top layer) can be modeled as ideal voltage sources. Transistors, logic gates, and memory units (connected to the bottom layer) can be modeled as ideal current drains.

Let N be the number of nodes in the RCL network that are not connected to ideal voltage sources. If only the voltage drops at these N nodes are of interest, a *revised* circuit model can be generated by short-circuiting all ideal voltage sources and reversing the directions of all ideal current sources [12]. Nodal voltages of the revised circuit are equal to voltage drops of the original circuit. The state-space equation obtained by modified nodal analysis [31] of the revised circuit can be written as follows:

$$C\dot{x}(t) + Gx(t) = Bu(t). \quad (1)$$

Here, $x(t) \in \mathbb{R}^n$ is the vector of nodal voltages and inductor currents, n is the sum of node number and inductor number, $u(t) \in \mathbb{R}^m$ is the vector of current drains, $C \in \mathbb{R}^{n \times n}$ is a diagonal matrix with its diagonal elements being capacitances and inductances, $G \in \mathbb{R}^{n \times n}$ is a matrix of conductances and “ ± 1 ,” $B \in \mathbb{R}^{n \times m}$ is the 0–1 current distribution matrix. Note that each row of B may contain more than one “1,” which means that more than one current source can be attached to a single node. On the other hand, each column of B contains exactly one “1,” which corresponds to the position at which a current source is attached.

B. Submodular Function and Polyhedron

Definition 1: Submodular Function [32]: Let M be a finite set (the ground set 2^M denotes the set of all the subsets of M), a set function $f : 2^M \mapsto \mathbb{R}$ is *submodular* if and only if for $\forall A_1, A_2 \subseteq M$, $f(A_1 \cup A_2) + f(A_1 \cap A_2) \leq f(A_2) + f(A_1)$.

One can prove the submodularity of a set function based on the following lemma [32]. According to the lemma, submodularity intuitively means that adding an element to a smaller set helps more than adding it to a larger set.

Lemma 1: A set function $f : 2^M \mapsto \mathbb{R}$ is *submodular* if and only if for $\forall A \subseteq M$ and $a_i, a_j \in M \setminus A$, $f(A \cup a_j) - f(A) \geq f(A \cup a_i \cup a_j) - f(A \cup a_i)$.

Definition 2: Submodular Polyhedron [32]: Let $f : 2^M \mapsto \mathbb{R}$ be a set function, the set $\{u \in \mathbb{R}^M : \sum_{a \in A} u(a) \leq f(A), \forall A \subseteq M\}$ is called a submodular polyhedron if f is submodular.

Here, the set M is simply an index set $\{1, 2, \dots, M_1\}$, $x \in \mathbb{R}^{M_1}$ is a vector of length M_1 with its subscripts being the members of M .

In LP problems, if the variables are restricted to a submodular polyhedron, SPGA can be utilized to find the optimal solution [32]–[34].

C. Krylov Subspace-Based MOR

Consider the state-space model in (1). If a linear combination of the state variables is of interest, (1) can be rewritten as

$$C\dot{x}(t) + Gx(t) = Bu(t) \quad (2a)$$

$$y(t) = e^T x(t) \quad (2b)$$

where (2a) is the same as (1), $y(t) \in \mathbb{R}$, $e \in \mathbb{R}^{n \times 1}$ is an arbitrary output distribution matrix.

In industrial power grid models, the number of state variables is large (in many cases $n > 10^6$), which renders transient analysis expensive. Therefore, MOR can be applied to reduce the computation complexity [35]–[37]. The transfer function of (2) and its Taylor expansion (at $s = 0$) is

$$\begin{aligned} H(s) &= e^T (sC + G)^{-1} B \\ &= e^T G^{-1} (I - sCG^{-1} + (sCG^{-1})^2 + \dots) B. \end{aligned} \quad (3)$$

As the number of outputs is much smaller than the number of inputs ($1 \ll m$), we choose to build output Krylov subspace. An order- r Krylov subspace based on output distribution matrix can be expressed as

$$\begin{aligned} &\mathcal{K}_r (G^{-T} C^T, G^{-T} e) \\ &= \text{span} \left\{ G^{-T} e, (G^{-T} C^T) G^{-T} e, \dots, (G^{-T} C^T)^{r-1} G^{-T} e \right\}. \end{aligned} \quad (4)$$

Using the Arnoldi orthogonalization approach, a set of orthonormal basis of the Krylov subspace can be found. Let $\mathcal{V} \in \mathbb{R}^{n \times r}$ be the matrix whose columns are the orthonormal basis (i.e., $\mathcal{V}^T \mathcal{V}$ is an identity matrix of dimension r and $\text{colspan}\{\mathcal{V}\} = \mathcal{K} (G^{-T} C^T, G^{-T} e, r)$). The reduced-order model can be expressed as follows:

$$C_r \dot{x}_r(t) + G_r x_r(t) = B_r u(t) \quad (5a)$$

$$y(t) = e_r^T x_r(t) \quad (5b)$$

where $x_r(t) = \mathcal{V}^T x(t)$, $G_r = \mathcal{V}^T G \mathcal{V}$, $C_r = \mathcal{V}^T C \mathcal{V}$, $B_r = \mathcal{V}^T B$, and $e_r = \mathcal{V}^T e$. It can be proven that the first r moments of the transfer function of the reduce-order model equal to those of the transfer function of the original model. The reduced-order model can be used for transient analysis which achieves faster speed. Besides, the reduced-order model (5) is guaranteed to be passive [35], [38].

III. HIERARCHICAL CONSTRAINTS AND LINEAR PROGRAMS

A. Transient Analysis

In this subsection, we derive the relationship between voltage drops and currents based on the procedure of transient analysis. Similar derivations appeared in [16]. Using the backward Euler method, (1) can be discretized as follows:

$$\left(G + \frac{C}{\Delta t} \right) x(t + \Delta t) = \frac{C}{\Delta t} x(t) + Bu(t + \Delta t). \quad (6)$$

Under the stability assumption [all the poles of (1) are in the open left half of complex plane], the system matrix $G + \frac{C}{\Delta t}$ is invertible. Define

$$\mathcal{M} = \left(G + \frac{C}{\Delta t} \right)^{-1} \frac{C}{\Delta t} \quad \mathcal{N} = \left(G + \frac{C}{\Delta t} \right)^{-1} B. \quad (7)$$

We have

$$x(t + \Delta t) = \mathcal{M}x(t) + \mathcal{N}u(t + \Delta t). \quad (8)$$

Dividing the simulation time span into k_t time steps, and with a zero initial state assumption (i.e., $x(0) = 0$), the voltage drops at the final time step can be represented as

$$x(k_t \Delta t) = \sum_{k=1}^{k_t} \mathcal{M}^{k_t-k} \mathcal{N}u(k \Delta t). \quad (9)$$

B. Hierarchical Constraints

In practice the peak value of a current source is bounded, i.e., $u_i(t) \leq I_{L,i}$. Local current constraints can be formulated by combining all these inequalities as

$$0 \leq u(k \Delta t) \leq I_L \quad (k = 1, \dots, k_t) \quad (10)$$

where $I_L \in \mathbb{R}^m$ is a vector with its i th element being the upper bound of the i th current source. Here, and in what follows, “ \leq ” is taken to be element-wise. On the other hand, block-level current constraints are formulated as follows:

$$Uu(k \Delta t) \leq I_G \quad (k = 1, \dots, k_t) \quad (11)$$

where $U \in \mathbb{R}^{p \times m}$ is a 0–1 matrix. Each inequality in (11) corresponds to a functional block, i.e., the total current of a functional block is bounded. The block-level current constraints here are different from [12] and [16] in the sense that each column of U contains at most one nonzero entry. This determines that one current source only belongs to one specific functional block (appears in only one inequality). If two different functional blocks draw currents from one node, the current drawn from this node should be modeled as two independent current sources. This agrees with the fact that each row in B may contain more than one “1,” as shown in Fig. 1.

Novel power constraints are introduced which restrict the average power consumptions of functional blocks

$$U \left(\sum_{k=1}^{k_t} u(k \Delta t) \right) \leq \frac{k_t}{V_{dd}} P_B. \quad (12)$$

Here $V_{dd} \in \mathbb{R}$ is the voltage of external power supply. $P_B \in \mathbb{R}^p$ is a vector with its i th element being the power limit of the i th functional block. Power constraints are usually specified as design requirements.

As functional blocks have interactions with each other, high-level power constraints are introduced to restrict the total

According to the definition of $c_{i,k}$ we have

$$c_{i,k} = e_i^T \left[\left(G + \frac{C}{\Delta t} \right)^{-1} \frac{C}{\Delta t} \right]^{k_t-k} \left(G + \frac{C}{\Delta t} \right)^{-1} B \quad (15)$$

where $e_i \in \mathbb{R}^{n \times 1}$ is the i th elementary unit vector. Directly computing the matrix inversion and multiplication is prohibitively expensive. Thus, we propose a method which only requires one sparse-LU decomposition and k_t forward and backward substitutions. Transpose (15), and we have

$$c_{i,k}^T = B^T \left(G^T + \frac{C^T}{\Delta t} \right)^{-1} \left[\frac{C^T}{\Delta t} \left(G^T + \frac{C^T}{\Delta t} \right)^{-1} \right]^{k_t-k} e_i. \quad (16)$$

Let $G^T + \frac{C^T}{\Delta t} = L_d U_d$ be an LU decomposition, we have

$$c_{i,k}^T = B^T U_d^{-1} L_d^{-1} \underbrace{\left(\frac{C^T}{\Delta t} \right) U_d^{-1} L_d^{-1} \dots \left(\frac{C^T}{\Delta t} \right) U_d^{-1} L_d^{-1}}_{k_t-k \text{ times}} e_i. \quad (17)$$

$L_d^{-1} (U_d^{-1})$ multiplying to a vector is fundamentally a forward (backward) substitution. Thus, computing $c_{i,k}$ from $k = 1$ to $k = k_t$ following (17) involves one sparse-LU factorization and k_t forward and backward substitutions. The computation complexity is lower than computing matrix inversion and matrix multiplication. However, it is proportional to the total number of steps (i.e., k_t). When k_t is large, the computation complexity is still large. Therefore, an efficient coefficient computation method based on MOR is proposed.

Computing $c_{i,k}$ for a particular i is (numerically) equivalent to transient analysis of the following state-space model:

$$C\dot{x}(t) + Gx(t) = Bu(t) \quad (18a)$$

$$y(t) = e_i^T x(t). \quad (18b)$$

A reduced model can be obtained using the projection matrix \mathcal{V} where $\text{colspan}(\mathcal{V}) = \mathcal{K} (G^{-T} C^T, G^{-T} e_i, r)$. We can use the reduced model to compute the coefficients $c_{i,k}$. The detailed flow is shown in Algorithm 1. Note that G_L^{-1} , G_U^{-1} , M_L^{-1} , M_U^{-1} multiplying a vector is essentially a forward or backward substitution.

Coefficient computation following Algorithm 1 has two advantages. First, the complexity of Algorithm 1 is lower than direct computation following (17) and is independent of the total number of steps k_t . Experiments have shown that a reduced model of order $r = 20$ – 50 can provide accurate results. Hence the complexity of steps 13–19 of Algorithm 1 is almost negligible compared with the complexity of (17). Steps 1–12 dominate the complexity of Algorithm 1, which is proportional to the reduced order r . As r is independent of step number k_t and increases very slowly with the increase of original order n , the overall complexity of Algorithm 1 is independent of k_t . Second, Algorithm 1 can be modified (as Algorithm 2) to handle adaptive step length. Algorithm 2 has roughly the same

Algorithm 1 Coefficient Computation Based on MOR

Input: $C, G, B, e_i, r, k_t, \Delta t$

Output: $c_{i,k}$ (for $k = 1, \dots, k_t$)

- 1: $G_L G_U = G^T$; (LU decomposition)
 - 2: $\mathcal{V}_0 = G_U^{-1} G_L^{-1} e_i$;
 - 3: $\mathcal{V}_0 = \mathcal{V}_0 / \|\mathcal{V}_0\|_2$;
 - 4: **for** $j_1 = 1, \dots, r - 1$ **do**
 - 5: $\mathcal{V}_{j_1} = -G_U^{-1} G_L^{-1} (C^T \mathcal{V}_{j_1-1})$;
 - 6: **for** $j_2 = 0, \dots, j_1 - 1$ **do**
 - 7: $\mathcal{V}_{j_1} = \mathcal{V}_{j_1} - (\mathcal{V}_{j_1}^T \mathcal{V}_{j_2}) \mathcal{V}_{j_2}$;
 - 8: **end for**
 - 9: $\mathcal{V}_{j_1} = \mathcal{V}_{j_1} / \|\mathcal{V}_{j_1}\|_2$;
 - 10: **end for**
 - 11: $\mathcal{V} = [\mathcal{V}_0 \dots \mathcal{V}_{r-1}]$;
 - 12: $G_r = \mathcal{V}^T G \mathcal{V}$; $C_r = \mathcal{V}^T C \mathcal{V}$; $B_r = \mathcal{V}^T B$; $e_{ir} = \mathcal{V}^T e_i$;
 - 13: $M_L M_U = G_r^T + \frac{C_r^T}{\Delta t}$; (LU decomposition)
 - 14: $T_{k_t} = M_U^{-1} M_L^{-1} e_{ir}$;
 - 15: $c_{i,k_t} = B_r^T T_{k_t}$;
 - 16: **for** $j_3 = k_t - 1, \dots, 1$ **do**
 - 17: $T_{j_3} = M_U^{-1} M_L^{-1} (\frac{C_r^T}{\Delta t} T_{j_3+1})$;
 - 18: $c_{i,j_3} = B_r^T T_{j_3}$;
 - 19: **end for**
-

Algorithm 2 Coefficient Computation w/ Adaptive Time Step

Input: $C, G, B, e_i, r, k_t, \Delta t_1, \dots, \Delta t_{k_t}$

Output: $c_{i,k}$ (for $k = 1, \dots, k_t$)

- 1: Same as steps 1~1 of Algorithm 1;
 - 2: $T_{k_t} = (G_r^T + \frac{C_r^T}{\Delta t_{k_t}})^{-1} e_{ir}$;
 - 3: $c_{i,k_t} = B_r^T T_{k_t}$;
 - 4: **for** $j_3 = k_t - 1, \dots, 1$ **do**
 - 5: $T_{j_3} = (G_r^T + \frac{C_r^T}{\Delta t_{j_3}})^{-1} (\frac{C_r^T}{\Delta t_{j_3}} T_{j_3+1})$;
 - 6: $c_{i,j_3} = B_r^T T_{j_3}$;
 - 7: **end for**
-

complexity as that of Algorithm 1. Moreover, when we employ Algorithm 3 to solve the LP problems, only the order of \bar{c}_k has influence on the result. In other words, although model reduction may introduce inaccuracy in coefficients, the voltage drop prediction will be still accurate as long as the order of \bar{c}_k is not changed significantly. Therefore, the accuracy of voltage-drop prediction is even better than the accuracy of coefficient computation. Thus, a low-order reduced-model can be used.

B. Problem Reformulation

With the coefficients $c_{i,k}$ computed by Algorithm 1 or Algorithm 2, we can set up the LP problem as in (14). Different components of a vector $u(k\Delta t)$ in (14) are in fact independent variables. Thus, there exist $k_t m$ variables and $k_t m$ coefficients in total. For simplicity, we rename the coefficients and the variables as follows:

$$\begin{aligned} c_1 &= e_1^T c_{i,1} & \cdots & & c_m &= e_m^T c_{i,1}, \\ & \vdots & & & & \vdots \\ c_{(k_t-1)m+1} &= e_1^T c_{i,k_t}, & \cdots & & c_{k_t m} &= e_m^T c_{i,k_t}, \end{aligned} \quad (19a)$$

$$\begin{aligned}
u_1 &= u_1(\Delta t), & \dots & & u_m &= u_m(\Delta t), \\
\vdots & & \ddots & & \vdots & \\
u_{(k_t-1)m+1} &= u_1(k_t \Delta t), & \dots & & u_{k_t m} &= u_m(k_t \Delta t).
\end{aligned} \tag{19b}$$

The constraints in (14) are essentially a set of inequalities. Specifically, there exist a total of $k_t m + k_t p + p + p_1 + \dots + p_q$ inequalities ($k_t m$ local current constraints, $k_t p$ block-level current constraints, p block-level power constraints, $p_1 + \dots + p_q$ inter-block power constraints). Each of these inequalities decides that the sum of a subset of the variables in (19b) is bounded. We summarize the notations involved in the constraints as follows:

- 1) κ_t : the total number of constraints. Here $\kappa_t = k_t m + k_t p + p + p_1 + \dots + p_q$;
- 2) κ : the index of constraint, $\kappa = 1, 2, \dots, \kappa_t$;
- 3) \mathcal{L}_κ : the set of indices of variables involved in the κ th constraint;
- 4) ℓ_κ : the bound of the κ th constraint;
- 5) i_{nd} : the index of grid node, with the subscript nd representing ‘‘node.’’ Here $i_{nd} \in \Omega$;
- 6) $x_{i_{nd}}$: the voltage drop of node i_{nd} at time step k_t .

The LP problem (14) can be rewritten as follows:

$$\begin{aligned}
\max x_{i_{nd}} &= \sum_{k=1}^{m k_t} c_k u_k \\
\text{s.t. } 0 &\leq \sum_{i \in \mathcal{L}_\kappa} u_k \leq \ell_\kappa \quad (\kappa = 1, \dots, \kappa_t).
\end{aligned} \tag{20}$$

Note that some of the coefficients (c_k) may be negative, which indicates that the currents of some nodes at some time steps may have reverse effects on the voltage drops. In other words, the smaller some currents (u_k) are, the larger the voltage drop ($x_{i_{nd}}$) is. This phenomenon is caused by the existence of inductors. From a mathematical perspective, the existence of inductors results in a non- M matrix (see [30] for the definition of M matrix). Thus, some entries of \mathcal{M} and/or \mathcal{N} may be negative. From a physical perspective, voltage drops (on inductors) caused by the changes of currents may be greater than voltage drops (on resistors) caused by the absolute values of currents. Hence small currents at previous time steps may result in large voltage drops at the current time step.

Lemma 3: $x_{i_{nd}}$ reaches its maximum value when all the variables u_k s associated with negative coefficients c_k s are set to zero.

Proof: See Appendix B. \blacksquare

Given the LP problem (20), the first step should be setting all the u_k s associated with negative c_k s to zero. Hence the number of variables to be determined is smaller than (or equal to) $k_t m$. Accordingly, some constraints can be deleted if all the variables it involves have been set to zero. Besides, we sort the positive c_k s in descending order and rearrange the variables following the order of c_k s. The following notations are used to further simplify the LP problem:

- 1) k_m : total number of positive coefficients ($k_m \leq m k_t$);
- 2) \bar{c}_k : positive coefficients in descending order, i.e., $\bar{c}_1 \geq \dots \geq \bar{c}_{k_m}$ ($k = 1, \dots, k_m$);

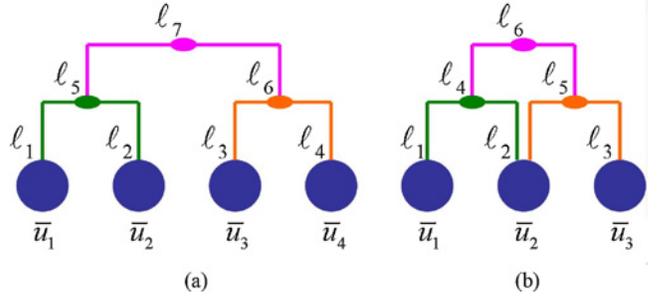


Fig. 3. Two illustrative examples. (a) Hierarchical. (b) Nonhierarchical.

- 3) \bar{u}_k : the variables associated with \bar{c}_k s ($k = 1, \dots, k_m$);
- 4) κ_m : total number of constraints left ($\kappa_m \leq \kappa_t$);
- 5) $\ell_\kappa, \mathcal{L}_\kappa$: a rearrange of constraints that still exist ($\kappa = 1, \dots, \kappa_m$).

The LP problem in (20) is converted to

$$\begin{aligned}
\max x_{i_{nd}} &= \sum_{k=1}^{k_m} \bar{c}_k \bar{u}_k \\
\text{s.t. } 0 &\leq \sum_{i \in \mathcal{L}_\kappa} \bar{u}_k \leq \ell_\kappa \quad (\kappa = 1, \dots, \kappa_m).
\end{aligned} \tag{21}$$

Note that $\bar{c}_1 \geq \dots \geq \bar{c}_{k_m} > 0$ in (21). To predict the voltage overshoot (instead of the voltage drop), the sign of the objective function should be reversed.

C. Submodular Polyhedron Greedy Algorithm

The constraints in the reformulated LP problem (21) remain hierarchical. The hierarchical constraint structure has the following property.

Property 1: For any two constraints j_1, j_2 ($1 \leq j_1, j_2 \leq \kappa_m$), at least one of the following three equations holds: 1) $\mathcal{L}_{j_1} \subseteq \mathcal{L}_{j_2}$; 2) $\mathcal{L}_{j_2} \subseteq \mathcal{L}_{j_1}$; and 3) $\mathcal{L}_{j_1} \cap \mathcal{L}_{j_2} = \emptyset$.

We look at two illustrative examples depicted in Fig. 3. The constraints in Fig. 3(a) are hierarchical, which can be written explicitly as follows:

$$\begin{aligned}
0 &\leq \bar{u}_1 \leq \ell_1, \quad 0 \leq \bar{u}_2 \leq \ell_2, \quad 0 \leq \bar{u}_3 \leq \ell_3, \quad 0 \leq \bar{u}_4 \leq \ell_4, \\
\bar{u}_1 + \bar{u}_2 &\leq \ell_5, \quad \bar{u}_3 + \bar{u}_4 \leq \ell_6, \quad \bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4 \leq \ell_7.
\end{aligned} \tag{22}$$

In this example, $k_m = 4$, $\kappa_m = 7$, $\mathcal{L}_1 = \{1\}$, $\mathcal{L}_2 = \{2\}$, $\mathcal{L}_3 = \{3\}$, $\mathcal{L}_4 = \{4\}$, $\mathcal{L}_5 = \{1, 2\}$, $\mathcal{L}_6 = \{3, 4\}$, $\mathcal{L}_7 = \{1, 2, 3, 4\}$. The upper bound of the sum of some variables (\circ) is denoted as $f(\circ)$. For this example, we have

$$\begin{aligned}
f(\bar{u}_1) &= \min\{\ell_1, \ell_5, \ell_7\}, \quad f(\bar{u}_3) = \min\{\ell_3, \ell_6, \ell_7\}, \\
f(\bar{u}_1 \cup \bar{u}_2) &= \min\{\ell_1 + \ell_2, \ell_5, \ell_7\}, \\
f(\bar{u}_1 \cup \bar{u}_2 \cup \bar{u}_3) &= \min\{f(\bar{u}_1 \cup \bar{u}_2) + f(\bar{u}_3), \ell_7\}.
\end{aligned} \tag{23}$$

Unlike the constraints in Fig. 3(a), the constraints in Fig. 3(b) are non-hierarchical because $\mathcal{L}_4 \cap \mathcal{L}_5 = \{1, 2\} \cap \{2, 3\} = \{2\} \neq \emptyset$ and $\mathcal{L}_4 \not\subseteq \mathcal{L}_5$ and $\mathcal{L}_5 \not\subseteq \mathcal{L}_4$.

Theorem 1: The feasible space defined by the constraints in (21) is a submodular polyhedron.

Algorithm 3 Submodular Polyhedron Greedy Algorithm**Input:** \bar{c}_k s, \mathcal{L}_κ s, ℓ_κ s, k_m , κ_m **Output:** \bar{u}_k s

- 1: **for** $k = 1, \dots, k_m$ **do**
- 2: Select all the sets \mathcal{L}_κ that satisfy $k \in \mathcal{L}_\kappa$. The subscripts of these \mathcal{L}_κ s form a set Φ_k ;
- 3: Set \bar{u}_k to $\min\{\ell_\kappa | \kappa \in \Phi_k\}$;
- 4: $\ell_\kappa = \ell_\kappa - \bar{u}_k$ for all $\kappa \in \Phi_k$;
- 5: **end for**

Algorithm 4 Algorithm Flow for Voltage-Drop Prediction**Input:** $C, G, B, \Omega, r, k_t, \Delta t$ **Output:** $x_{ind}, u(k\Delta t)$ ($i_{ind} \in \Omega, k=1, \dots, k_t$)

- 1: Set up hierarchical constraints (10)-(13);
- 2: **for** $i_{ind} \in \Omega$ (*in parallel*) **do**
- 3: $c_{ind,k} = \text{ALG1}(C, G, B, e_{i_{ind}}, r, k_t, \Delta t)$;
- 4: Set up LP problem as (14);
- 5: Reformulate (14) as (21) following Subsection IV-B. \bar{c}_k , \mathcal{L}_κ , ℓ_κ , k_m and κ_m are generated in this process;
- 6: $\bar{u}_k = \text{ALG3}(\bar{c}_k, \mathcal{L}_\kappa, \ell_\kappa, k_m, \kappa_m)$;
- 7: $x_{i_{ind}} = \sum_{k=1}^{k_m} \bar{c}_k \bar{u}_k$;
- 8: **end for**
- 9: $x_{\text{worst}} = \max\{x_{i_{ind}} | i_{ind} \in \Omega\}$;

Proof: See Appendix C. ■

According to Theorem 1, the feasible space in the LP problem (21) is a submodular polyhedron, which enables a fast greedy algorithm as shown in Algorithm 3.

Theorem 2: $\bar{u} = [\bar{u}_1, \dots, \bar{u}_{k_m}]^T$ solved by Algorithm 3 is the optimal solution of LP problem (21).

Proof: See Appendix D. ■*D. Algorithm Flow and Complexity Analysis*

The complete algorithm flow for worst-case voltage drop prediction is summarized as Algorithm 4, where ALG1 and ALG3 represent Algorithm 1 and Algorithm 3, respectively.

Given different time step lengths, Algorithm 4 can be adapted by replacing ALG1 in Step 4 by ALG2. The complexity of Algorithm 4 is analyzed below.

- 1) Complexity of coefficient computation (Algorithm 1).
 - a) Steps 1–10. These steps involve one sparse LU decomposition, r forward and backward substitutions, $r(r-1)/2$ inner products, $r(r-1)/2$ vector subtractions. As G^T is a sparse matrix with $O(n)$ nonzero entries, the complexity of a sparse LU decomposition is $O(n^\alpha)$ ($1 < \alpha < 2$) and the complexity of one forward (backward) substitution is $O(n^\beta)$ ($1 < \beta < 2$). Thus, the complexity of Steps 1–10 is $O(n^\alpha + n^\beta r + nr^2)$.
 - b) Steps 11–12. The complexity of these two steps is $O(nr)$.
 - c) Steps 13–19. These steps involve one LU decomposition and k_t forward and backward substitutions. The complexity is $O(r^3 + r^2 k_t)$.

In summary, the complexity of 12–19 is negligible compared with the complexity of Steps 1–10. Therefore, the

overall complexity of Algorithm 1 is $O(n^\alpha + n^\beta r + nr^2)$.

- 2) Complexity of problem reformulation: problem reformulation process involves finding negative c_k s and sorting positive c_k s. The complexity of former is $O(mk_t)$. Using efficient sorting algorithm, the complexity of the latter is $O(k_m \log k_m)$ with $k_m \leq mk_t$.
- 3) Complexity of SPGA: Algorithm 3 involves one “min” function and $|\Phi_k|$ scalar subtractions in each cycle. Here $|\Phi_k|$ denotes cardinality of the set Φ_k . $|\Phi_k|$ roughly equals to $q+3$, the total number of constraint levels. Thus, the complexity of one cycle is $O(q+3)$. The overall complexity of Algorithm 3 is $O(k_m q)$.

Therefore, the complexity for analysis of one node in Ω is $O(n^\alpha + n^\beta r + nr^2 + k_m \log k_m + k_m q)$. The total complexity is $O(|\Omega|(n^\alpha + n^\beta r + nr^2 + k_m \log k_m + k_m q))$.

1) *Comparison with Direct Coefficient Computation:* The complexity of direct coefficient computation (as in [19]) is $O(n^\alpha + k_t n^\beta)$ (using notations in this paper). Note that r is small ($< k_t$) and independent of total number of time steps. Thus, complexity of Algorithm 1 is less than that of direct computation. Moreover, the complexity of Algorithm 1 is independent of k_t , which enables more time steps to be considered.

2) *Comparison with Standard LP Solver:* Standard LP methods include simplex algorithm, ellipsoid algorithm, and interior point algorithm [39]. Simplex algorithm is theoretically NP-hard and has a complexity larger than $O(k_m^3)$ in most practical examples. The complexity of ellipsoid algorithm is $O(k_m^4)$. The complexity of interior point algorithm is $O(k_m^{3.5})$. Here k_m is the number of variables. In contrast, the complexity of SPGA is $O(k_m \log k_m + k_m q)$ (including problem reformulation). We thereby conclude that SPGA *dramatically* reduces the computational complexity.

V. NUMERICAL EXAMPLES

A. Data Description

We generate four 3-D power grids as benchmarks. Each of the power grids contains four metal layers (namely, M1, M3, M6, and AP layers) and can be modeled as an RCL network. The effects of package are also considered by including package resistors, package inductors, and package capacitors in the RCL network. The parameters of these four power grids are shown in Table I. LP problems are set up to predict voltage drops of the power grids, with both local constraints and global (including global current, block-level power, and inter-block-level power) constraints. The experiments are run on a Linux workstation with 2.5 GHz Intel Xeon CPU and 8 GB RAM.

B. Comparison of SPGA with Standard LP Solver

In this experiment, we compare the runtime and accuracy of standard LP solver with that of SPGA. The results (single node prediction) are recorded in Table II. Table II also reports the number of variables and global constraints of the LP problems. “Setup” time is the same for both methods as they both employ Algorithm 1 to calculate coefficients. For PG-1 and PG-2, the voltage predictions of standard LP solver converge to the values obtained by SPGA with the increasing of maximum

TABLE I
PARAMETERS OF POWER GRIDS USED IN THE EXPERIMENTS

Power Grids	Nodes (N)	Drains (m)	Matrix Size (n)	No of Rs	No of Cs	No of Ls	Mean of Rs	Mean of Cs	Mean of Ls
PG-1	594	297	878	494	279	284	5.82 S	1.50 pF	4.35 pH
PG-2	18 480	1540	27 646	13 651	9121	9166	4.70 S	0.466 pF	0.109 pH
PG-3	78 542	6546	117 648	56 327	39 021	39 106	24.7 S	0.103 pF	3.82 pH
PG-4	1 127 355	93 947	1 660 823	805 253	562 679	562 858	17.7 S	0.103 pF	3.82 pH

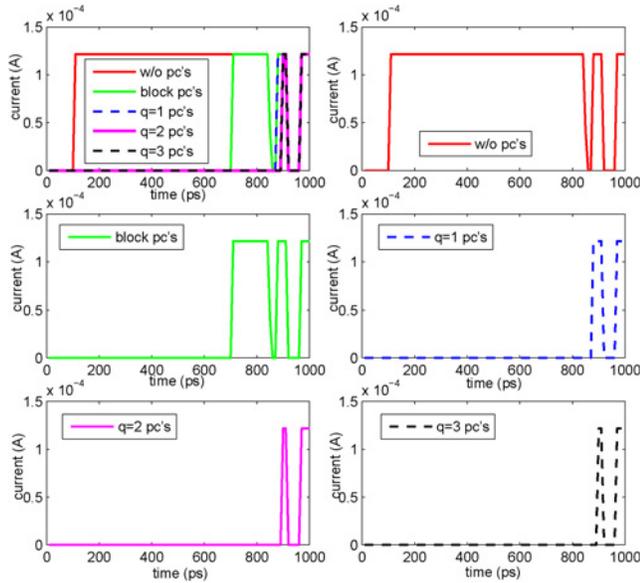


Fig. 4. Worst-case current waveforms of one current drain in PG-2 under different constraints. The second to sixth subfigures are current waveforms under constraints 1 to 5. The first subfigure is an overlap the second to six subfigures.

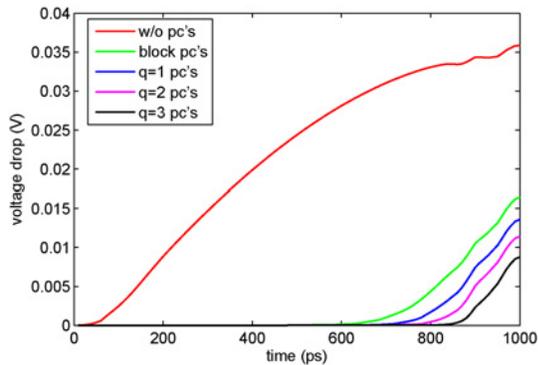


Fig. 5. Worst-case voltage drop on one node in PG-2 under different constraints.

iteration number. This fact indicates that standard LP solver and SPGA give the same voltage drop prediction provided that the maximum iteration number is large enough. For PG-3 and PG-4, the standard LP solver cannot converge in the given maximum iteration number. The speedup of SPGA is recorded in the last column.

From Table II we conclude that SPGA *greatly* ($53 \times \sim \infty$) reduces the runtime for worst-case voltage drop prediction while at the same time provides guaranteed optimal solution.

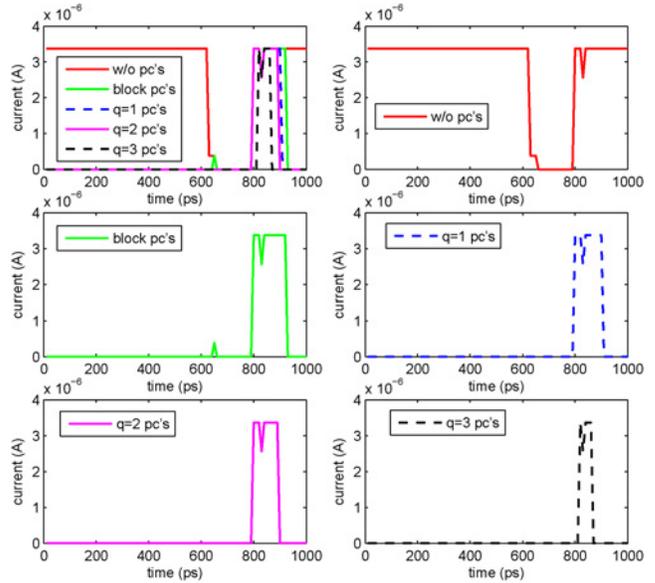


Fig. 6. Worst-case current waveforms of one current drain in PG-3 under different constraints. The second to sixth subfigures are current waveforms under constraints 1 to 5. The first subfigure is an overlap the second to six subfigures.

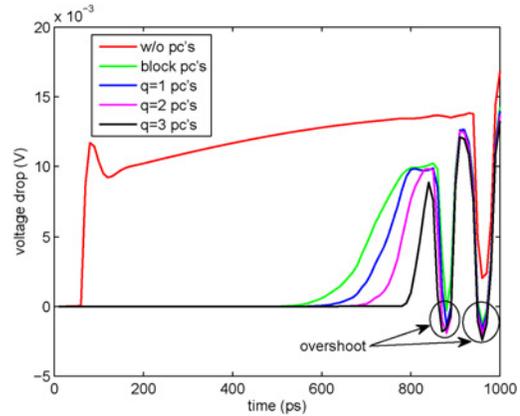


Fig. 7. Worst-case voltage drop on one node in PG-3 under different constraints. The circles label the regions where voltage overshoots occur.

Besides, it can be seen from Table II that the runtime of SPGA increases almost linearly with problem size. This agrees with the complexity analysis in Section IV-D.

C. Impact of Power Constraints (Hierarchy)

In this experiment, we show the impact of power constraints (pcs) on voltage drop predictions. We study different kinds of constraint structures: 1) ones including current constraints only

TABLE II
COMPARISON OF STANDARD LP SOLVER AND SPGA

	LP Problem		Standard LP Solver							Proposed Algorithm			Speedup
	Var's	Con's	Setup	Solving			Voltage			Setup	Solving	Voltage	
PG-1	29.7K	1.5K	0.136 s	4.22 s iter = 50	16.1 s iter = 200	23.8 s iter = 300	6.47 mV iter = 50	18.1 mV iter = 200	20.9 mV iter = 300	0.136 s	3.79e-3 s	20.9 mV	176×
PG-2	154K	7.8K	5.80 s	27.2 s iter = 50	177 s iter = 300	302 s iter = 500	5.79 mV iter = 50	14.9 mV iter = 300	19.2 mV iter = 500	5.80 s	3.68e-2 s	19.2 mV	53×
PG-3	655K	33K	19.8 s	145 s iter = 50	803 s iter = 300	1395 s iter = 500	4.07 mV iter = 50	9.32 mV iter = 300	11.7 mV iter = 500	19.8 s	0.266 s	13.4 mV	>70.5×
PG-4	9.39M	475K	753 s	142 s iter = 1	743 s iter = 5	Fail iter = 10	790 mV iter = 1	5.63e5 mV iter = 5	Fail iter = 10	753 s	3.83 s	15.0 mV	∞

TABLE III
IMPACT OF POWER CONSTRAINTS

		Without pcs		Block pcs		$q = 1$		$q = 2$		$q = 3$		Over-estimation
		Voltage	Time	Voltage	Time	Voltage	Time	Voltage	Time	Voltage	Time	
PG-1	One node	17.8 mV	0.136 s	16.4 mV	0.136 s	16.3 mV	0.136 s	16.1 mV	0.136 s	15.8 mV	0.137 s	8.54%–12.7%
	Complete	31.4 mV	40.2 s	27.8 mV	40.2 s	26.2 mV	40.3 s	24.9 mV	40.4 s	23.5 mV	40.7 s	12.9%–33.6%
PG-2	One node	35.9 mV	6.12 s	16.4 mV	6.19 s	13.6 mV	6.36 s	11.4 mV	6.37 s	8.77 mV	6.44 s	119%–309%
	Complete	69.8 mV	9.41e3 s	29.5 mV	9.53e3 s	24.0 mV	9.81e3 s	20.9 mV	9.82e3 s	17.6 mV	9.89 s	137%–297%
PG-3	One node	16.8 mV	21.0 s	14.3 mV	21.3 s	14.0 mV	21.8 s	13.7 mV	22.2 s	13.3 mV	22.6 s	17.5%–26.3%
PG-4	One node	20.1 mV	749 s	16.5 mV	753 s	16.2 mV	759 s	15.8 mV	765 s	15.4 mV	771 s	21.8%–30.5%

TABLE IV
COMPARISON OF MOR-BASED COEFFICIENT COMPUTATION WITH DIRECT ONES

	Direct Method		MOR Method			Relative Error		Speedup
	Voltage Drop	CPU Time	Reduced Order	Voltage Drop	CPU Time	Coefficients	Voltage Drop	
PG-1	16.8 mV	0.411 s	20	16.9 mV	0.0895 s	1.31e-1	8.57e-3	4.59×
PG-2	19.3 mV	19.9 s	30	19.3 mV	5.85 s	3.32e-5	3.23e-7	3.40×
PG-3	13.5 mV	54.3 s	40	13.5 mV	20.1s	1.76e-3	6.19e-5	2.70×
PG-4	14.9 mV	1587	50	14.5 mV	772 s	1.91e-1	3.03e-2	2.06×

(without pcs); 2) ones including block level pcs (block pcs); 3) ones including one-level inter-block pcs ($q = 1$ pcs); 4) ones including two-level inter-block pcs ($q = 2$ pcs); and 5) ones including three-level inter-block pcs ($q = 3$ pcs). The worst-case voltage drop and CPU time under every constraint structure are reported in Table III. For PG-1 and PG-2, we also perform *complete* predictions (see Section IV-A). The over-estimation in the table is calculated by comparing the voltage drop under constraint 1 with that under constraints 2, 3, 4, and 5. From Table III, we conclude that the worst-case voltage drop will be significantly over-estimated if we do not consider power constraints. On the other hand, the runtime for all constraint structures is roughly the same, because the setup time (instead of the solving time) dominates the overall runtime.

Remarks: the complete prediction in Table III is executed sequentially. As the predictions for nodes ($i_{nd} \in \Omega$) are independent of each other, Algorithm 4 can be executed in parallel. In this case, the runtime for complete prediction can be greatly reduced.

To provide an intuitive picture, we also plot the current waveforms and voltage drops at some particular nodes for PG-2 and PG-3, as shown in Figs. 4–7. From Figs. 4 and 6, we can see that the current waveforms satisfy $1 > 2 > 3 > 4 > 5$. Similarly, Figs. 5 and 7 show that the voltage drops also satisfy $1 > 2 > 3 > 4 > 5$. In Fig. 7, there exist voltage overshoots (i.e., the voltage of one node is larger than external voltage

supply). This is because that the mean value of inductance is relatively large compared with that of capacitance for PG-3. Thus the inductance effects play an important role in the voltage drops, which may result in negative voltage drops.

D. Comparison of MOR-Based with Direct Coefficient Computation

In this experiment, we compare the direct coefficient computation method (as in [19]) with Algorithm 1. The results are reported in Table IV. The reduced order we used in Algorithm 1 is recorded in the table. The relative error of coefficients is calculated by

$$err_c = \frac{\| [c_{i,1}, \dots, c_{i,k_i}] - [c'_{i,1}, \dots, c'_{i,k_i}] \|_2}{\| [c'_{i,1}, \dots, c'_{i,k_i}] \|_2} \quad (24)$$

where $c_{i,k}$ is the output of Algorithm 1 and $c'_{i,1}$ is the counterpart calculated by direct method. The relative error of voltage drop is calculated by

$$err_v = \frac{|x_i - x'_i|}{|x'_i|} \quad (25)$$

where x_i is the voltage drop calculated with $c_{i,k}$ and x'_i is the voltage drop calculated with $c'_{i,k}$.

We conclude from Table IV that coefficient computation based on MOR is much faster than direct method, while at

the same time provides an accurate voltage drop prediction. Note that the relative error of voltage drop is much smaller than that of coefficients. The reason is that when we employ Algorithm 3 to solve the LP problems, only the order of \bar{c}_k s has influence on the result. In other words, although model reduction may introduce inaccuracy in coefficients, the voltage drop prediction will be still accurate as long as the order of \bar{c}_k s is not changed significantly.

VI. CONCLUSION

A novel early-stage power grid verification algorithm based on hierarchical constraints has been proposed in this paper. Power constraints are introduced to provide more realistic worst-case voltage drop predictions. MOR-based coefficient computation is proposed to reduce the computation complexity of formulating the LP problems. The hierarchical constraint structure enables SPGA which *dramatically* reduces the complexity of solving the LP problems. Experimental results have shown that: 1) power constraints reduces the over-pessimism in worst-case voltage drop predictions; 2) MOR-based methods reduce the CPU time for formulating the LP problems with negligible errors; and 3) SPGA dramatically reduces the CPU time for solving the LP problems and is guaranteed to deliver the optimal solutions.

APPENDIX A

PROOF OF LEMMA 2

$\max\{x_i(k_\tau \Delta t)\}$ is solved under the same constraints as (14), with the objective function being $x_i(k_\tau \Delta t) = \sum_{k=1}^{k_\tau} \hat{c}_{i,k} u(k \Delta t)$. Here $\hat{c}_{i,k}$ is the i th row of $\mathcal{M}^{k_\tau-k} \mathcal{N}$. It is readily verified that $c_{i,k+k_\tau-k_\tau} = \mathcal{M}^{k_\tau-(k+k_\tau-k_\tau)} \mathcal{N} = \hat{c}_{i,k}$. Assume $\max\{x_i(k_\tau \Delta t)\}$ is reached when the current patterns are $u^{(t)}(k \Delta t)$. Let $u^{(t)}(k \Delta t) = 0$ for $k = 1, \dots, k_t - k_\tau$ and $u^{(t)}(k \Delta t) = u^{(\tau)}((k - k_t + k_\tau) \Delta t)$ for $k = k_t - k_\tau + 1, \dots, k_t$. We have that the current waveform $u^{(t)}(k \Delta t)$ also satisfies (10)–(13) if $u^{(\tau)}(k \Delta t)$ satisfies (10)–(13). On the other hand

$$\begin{aligned} x_i(k_t \Delta t) &= \sum_{k=1}^{k_t} c_{i,k} u^{(t)}(k \Delta t) = \sum_{k=k_t-k_\tau+1}^{k_t} c_{i,k} u^{(t)}(k \Delta t) \\ (k' = k - k_t + k_\tau) &= \sum_{k'=1}^{k_\tau} c_{i,k'+t-k_\tau} u^{(t)}((k' + t - k_\tau) \Delta t) \\ &= \sum_{k'=1}^{k_\tau} \hat{c}_{i,k'} u^{(\tau)}(k' \Delta t) = \max\{x_i(k_\tau \Delta t)\}. \end{aligned}$$

Therefore, $\max\{x_i(k_t \Delta t)\} \geq \max\{x_i(k_\tau \Delta t)\}$.

APPENDIX B

PROOF OF LEMMA 3

Assume $x_{i_{nd}}$ reaches its maximum value at a point $u^{(1)}$ with at least one variable u_j associated with negative coefficient c_j being positive. Then we construct a new point $u^{(2)}$ by setting

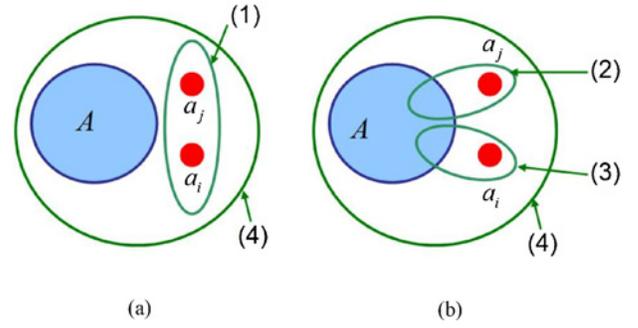


Fig. 8. Illustrative pictures for the proof of Theorem 1. (a) First case. (b) Second case.

u_j to zero. It is readily verified that $u^{(2)}$ also satisfies the constraints in (20) because for any subset \mathcal{L} we have $\sum_{i \in \mathcal{L}} u_k^{(2)} \leq \sum_{i \in \mathcal{L}} u_k^{(1)}$. On the other hand, $x_{i_{nd}}|_{u=u^{(2)}} > x_{i_{nd}}|_{u=u^{(1)}}$, which conflicts with the assumption that $u^{(1)}$ is optimal.

APPENDIX C

PROOF OF THEOREM 1

Consider $\forall A \subseteq \{1, \dots, k_m\}$, $a_i, a_j \in \{1, \dots, k_m\} \setminus A$. There exist four types of constraints in total that involve more than one party among A , a_i and a_j : 1) ones that involve a_i and a_j but do not involve A ; 2) ones that involve a_j and A but do not involve a_i ; 3) ones that involve a_i and A but do not involve a_j ; and 4) ones that involve a_j , a_i and A . Among these four types, 1 and 2 cannot exist simultaneously (according to Property 1). Similarly, 1 and 3 cannot exist simultaneously. Therefore, we only have to consider two general cases: 1) (1) and (4), and 2) (2), (3), and (4). See Fig. 8.

For 1), let the total number of type-(1) constraints be x and the total number of type-(4) constraints be y . Assume WLOG that

$$\underbrace{\mathcal{L}_{i_1} \subset \dots \subset \mathcal{L}_{i_x}}_{\text{type-(1)}} \subset \underbrace{\mathcal{L}_{j_1} \subset \mathcal{L}_{j_y}}_{\text{type-(2)}}.$$

Let the subsets of A involved in constraints j_1, \dots, j_y be A_{j_1}, \dots, A_{j_y} , respectively. We have $A_{j_1} \subseteq \dots \subseteq A_{j_y}$. Accordingly

$$\begin{aligned} f(A \cup a_i) - f(A) &= \min\{f(a_i), \ell_{j_1} - f(A_{j_1}), \dots, \ell_{j_y} - f(A_{j_y})\} \\ f(A \cup a_j \cup a_i) - f(A \cup a_j) &= \min\{f(a_i), \ell_{i_1} - f(a_j), \dots, \ell_{i_x} - f(a_j), \\ &\quad \ell_{j_1} - f(A_{j_1} \cup a_j), \dots, \ell_{j_y} - f(A_{j_y} \cup a_j)\}. \end{aligned}$$

Because $\ell_{j_1} - f(A_{j_1} \cup a_j) \leq \ell_{j_1} - f(A_{j_1}), \dots, \ell_{j_y} - f(A_{j_y} \cup a_j) \leq \ell_{j_y} - f(A_{j_y})$, we have $f(A \cup a_j \cup a_i) - f(A \cup a_j) \leq f(A \cup a_i) - f(A)$.

For 2), let the total number of type-(2) constraints be y , the total number of type-(3) constraints be x , and the total number of type-(4) constraints be z . Assume WLOG that

$$\underbrace{\mathcal{L}_{i_1} \subset \dots \subset \mathcal{L}_{i_x}}_{\text{type-(3)}} \subset \underbrace{\mathcal{L}_{l_1} \subset \mathcal{L}_{l_z}}_{\text{type-(4)}}; \quad \underbrace{\mathcal{L}_{j_1} \subset \dots \subset \mathcal{L}_{j_y}}_{\text{type-(2)}} \subset \underbrace{\mathcal{L}_{l_1} \subset \mathcal{L}_{l_z}}_{\text{type-(4)}}.$$

Let the subsets of A involved in the constraints i_1, \dots, i_x and l_1, \dots, l_z be A_{i_1}, \dots, A_{i_x} and A_{l_1}, \dots, A_{l_z} , respectively. We have $A_{i_1} \subseteq \dots \subseteq A_{i_x} \subseteq A_{l_1} \subseteq \dots \subseteq A_{l_z}$. Accordingly

$$\begin{aligned} f(A \cup a_i) - f(A) &= \min\{f(a_i), \ell_{i_1} - f(A_{i_1}), \dots, \ell_{i_x} - f(A_{i_x}), \\ &\quad \ell_{l_1} - f(A_{l_1}), \dots, \ell_{l_z} - f(A_{l_z})\} \\ f(A \cup a_j \cup a_i) - f(A \cup a_j) &= \min\{f(a_i), \ell_{i_1} - f(A_{i_1}), \dots, \\ &\quad \ell_{i_x} - f(A_{i_x}), \ell_{l_1} - f(A_{l_1 \cup a_j}), \dots, \ell_{l_z} - f(A_{l_z \cup a_j})\}. \end{aligned}$$

Because $\ell_{l_1} - f(A_{l_1 \cup a_j}) \leq \ell_{l_1} - f(A_{l_1}), \dots, \ell_{l_z} - f(A_{l_z \cup a_j}) \leq \ell_{l_z} - f(A_{l_z})$, we have $f(A \cup a_j \cup a_i) - f(A \cup a_j) \leq f(A \cup a_i) - f(A)$.

Therefore, $f(A \cup a_j \cup a_i) - f(A \cup a_j) \leq f(A \cup a_i) - f(A)$ always holds. According to Lemma 1, we have the $f(\circ)$ being a submodular function and the feasible space of (21) is a submodular polyhedron.

APPENDIX D PROOF OF THEOREM 2

The LP problem (21) satisfies the following properties: 1) the objective function is a linear combination of variables; 2) all the coefficients \bar{c}_k s are positive; and 3) the feasible space defined by the constraints is a submodular polyhedron. According to the theory of submodular optimization [refer to Theorem (19) in [33] for details], the solution solved by SPGA (Algorithm 3) is optimal.

REFERENCES

- [1] S. Nassif and J. Kozhaya, "Fast power grid simulation," in *Proc. IEEE/ACM Des. Autom. Conf.*, Jun. 2000, pp. 156–161.
- [2] T. Chen and C. Chen, "Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative methods," in *Proc. IEEE/ACM Des. Autom. Conf.*, Jun. 2001, pp. 559–562.
- [3] J. Kozhaya, S. Nassif, and F. Najm, "A multigrid-like technique for power grid analysis," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 21, no. 10, pp. 1148–1160, Oct. 2002.
- [4] M. Zhao, R. Panda, S. Sapatnekar, and D. Blaauw, "Hierarchical analysis of power distribution networks," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 21, no. 2, pp. 159–168, Feb. 2002.
- [5] S. Pant, D. Blaauw, V. Zolotov, S. Sundareswaran, and R. Panda, "A stochastic approach to power grid analysis," in *Proc. IEEE/ACM Des. Autom. Conf.*, Jun. 2004, pp. 171–176.
- [6] Q. Zhu, *Power Distribution Network Design for VLSI*. New York: Wiley-IEEE, 2004.
- [7] H. Qian, S. Nassif, and S. Sapatnekar, "Power grid analysis using random walks," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 24, no. 8, pp. 1204–1224, Aug. 2005.
- [8] P. Ghanta, S. Vrudhula, R. Panda, and J. Wang, "Stochastic power grid analysis considering process variations," in *Proc. IEEE/ACM Conf. Des. Autom. Test Eur.*, Mar. 2005, pp. 964–969.
- [9] E. Chiprout, "Fast flip-chip power grid analysis via locality and grid shells," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2005, pp. 485–488.
- [10] Y. Wang, C. Lei, G. Pang, and N. Wong, "MFTI: Matrix-format tangential interpolation for modeling multiport systems," in *Proc. IEEE/ACM Des. Autom. Conf.*, Jun. 2010, pp. 683–686.
- [11] Z. Feng and P. Li, "Multigrid on GPU: Tackling power grid analysis on parallel SIMT platforms," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2008, pp. 647–654.
- [12] D. Kouroussis and F. Najm, "A static pattern-independent technique for power grid voltage integrity verification," in *Proc. IEEE/ACM Des. Autom. Conf.*, Jun. 2003, pp. 99–104.
- [13] H. Qian, S. Nassif, and S. Sapatnekar, "Early-stage power grid analysis for uncertain working modes," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 24, no. 5, pp. 676–682, May 2005.
- [14] N. Ghani and F. Najm, "Handling inductance in early power grid verification," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2006, pp. 127–134.
- [15] I. Ferzli, F. Najm, and L. Kruse, "A geometric approach for early power grid verification using current constraints," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2007, pp. 40–47.
- [16] A. Ghani and F. Najm, "Fast vectorless power grid verification using an approximate inverse technique," in *Proc. IEEE/ACM Des. Autom. Conf.*, Jul. 2009, pp. 184–189.
- [17] X. Xiong and J. Wang, "An efficient dual algorithm for vectorless power grid verification under linear current constraints," in *Proc. IEEE/ACM Des. Autom. Conf.*, Jun. 2010, pp. 837–842.
- [18] P. Du, X. Hu, S. Weng, A. Shayan, X. Chen, E. Engin, and C. Cheng, "Worst-case noise prediction with non-zero current transition times for early power distribution system verification," in *Proc. Int. Symp. Quality Electron. Des.*, 2010, pp. 624–631.
- [19] C. K. Cheng, P. Du, A. B. Kahng, G. K. H. Pang, Y. Wang, and N. Wong, "More realistic power grid verification based on hierarchical current and power constraints," in *Proc. Int. Symp. Phys. Des.*, 2011, pp. 159–166.
- [20] Z. Feng and Z. Zeng, "Parallel multigrid preconditioning on graphics processing units (GPUs) for robust power grid analysis," in *Proc. IEEE/ACM Des. Autom. Conf.*, Jun. 2010, pp. 661–666.
- [21] M. Aydonat and F. Najm, "Power grid correction using sensitivity analysis," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2010, pp. 808–815.
- [22] M. Avci and F. Najm, "Early P/G grid voltage integrity verification," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2010, pp. 816–823.
- [23] N. Evmoropoulos, M. Rammou, G. Stamoulis, and J. Moondanos, "Characterization of the worst-case current waveform excitations in general RLC-model power grid analysis," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2010, pp. 824–830.
- [24] Z. Zeng and P. Li, "Locality-driven parallel power grid optimization," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 28, no. 8, pp. 1190–1200, Aug. 2009.
- [25] S. Zhao, K. Roy, and C. Koh, "Decoupling capacitance allocation and its application to power-supply noise-aware floorplanning," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 21, no. 1, pp. 81–92, Jan. 2002.
- [26] H. Su, S. Sapatnekar, and S. Nassif, "An algorithm for optimal decoupling capacitor sizing and placement for standard cell layouts," in *Proc. IEEE Int. Symp. Phys. Des.*, Apr. 2002, pp. 68–73.
- [27] K. Wang and M. Marek-Sadowska, "On-chip power-supply network optimization using multigrid-based technique," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 24, no. 3, pp. 407–417, Mar. 2005.
- [28] S. Sapatnekar and H. Su, "Analysis and optimization of power grids," *IEEE Des. Test Comput.*, vol. 20, no. 3, pp. 7–15, May–Jun. 2003.
- [29] T. Wang and C. Chen, "Optimization of the power/ground network wire sizing and spacing based on sequential network simplex algorithm," in *Proc. IEEE Int. Symp. Quality Electron. Des.*, Mar. 2002, pp. 157–162.
- [30] R. Plemmons, "M-matrix characterizations: I-nonsingular M-matrices," *Linear Algebra Its Applicat.*, vol. 18, no. 2, pp. 175–188, 1977.
- [31] C. Ho, A. Ruehli, and P. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits Syst.*, vol. 22, no. 6, pp. 504–509, Jun. 1975.
- [32] S. Fujishige, *Submodular Functions and Optimization*. Amsterdam, The Netherlands: Elsevier Science, 2005.
- [33] J. Edmonds, "Submodular functions, matroids, and certain polyhedra," in *Combinatorial Optimization-Eureka, You Shrink!* New York: Springer-Verlag, 2003, pp. 11–26.
- [34] S. Fujishige, "Structures of polyhedra determined by submodular functions on crossing families," *Math. Programming*, vol. 29, no. 2, pp. 125–141, 1984.
- [35] A. Odabasioglu, M. Celik, and L. Pileggi, "Passive and reduced order interconnect macromodeling algorithm," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 17, no. 8, pp. 645–654, Aug. 1998.
- [36] J. Phillips, L. Daniel, and L. Silveira, "Guaranteed passive balancing transformations for model order reduction," in *Proc. 39th Annu. Des. Autom. Conf.*, 2002, pp. 52–57.
- [37] B. Bond and L. Daniel, "Guaranteed stable projection-based model reduction for indefinite and unstable linear systems," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2008, pp. 728–735.

- [38] Y. Wang, Z. Zhang, C. Koh, G. Pang, and N. Wong, "PEDS: Passivity enforcement for descriptor systems via Hamiltonian-symplectic matrix pencil perturbation," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2010, pp. 800–807.
- [39] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.



Yuanzhe Wang received the B.Eng. and M.Phil. degrees in electrical engineering from Tianjin University, Tianjin, China, and the University of Hong Kong, Pokfulam, Hong Kong, in 2009 and 2011, respectively. He is currently working toward the Ph.D. degree in electrical and computer engineering with Carnegie Mellon University, Pittsburgh, PA.

His current research interests include computer-aided design of very large-scale integrated circuits, with emphasis on compressed sensing, power/ground network analysis, model order reduction,

and analog/radio frequency circuit simulation.



Xiang Hu received the B.E. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 2005 and 2007, respectively. He is currently working toward the Ph.D. degree in electrical and computer engineering with the University of California at San Diego, La Jolla.

His current research interests include analysis and optimization of power distribution networks, and circuit simulation.



Chung-Kuan Cheng (S'82–M'84–SM'95–F'00) received the B.S. and M.S. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, and the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, in 1984.

From 1984 to 1986, he was a Senior CAD Engineer with Advanced Micro Devices, Inc., Sunnyvale, CA. In 1986, he joined the University of California at San Diego (UCSD), La Jolla, where he is currently a Professor with the Department of Computer

Science and Engineering and an Adjunct Professor with the Department of Electrical and Computer Engineering. He served as a Chief Scientist with Mentor Graphics, Wilsonville, OR, in 1999. He was appointed an Honorary Guest Professor of Tsinghua University, Beijing, China, from 2002 to 2008. His current research interests include medical modeling and analysis, network optimization, and design automation on microelectronic circuits.

Dr. Cheng was an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN from 1994 to 2003. He was a recipient of best paper awards from the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN in 1997 and 2002, and received the NCR Excellence in Teaching Award, School of Engineering, UCSD, in 1991 and the IBM Faculty Awards in 2004, 2006, and 2007.



Grantham K. H. Pang (S'84–M'86–SM'01) received the Ph.D. degree from the University of Cambridge, Cambridge, MA, in 1986.

He was with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, from 1986 to 1996, and joined the Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam, Hong Kong, in 1996. His current research interests include visual surveillance, machine vision for surface defect detection, optical communications, control system design, intelligent control, and intelligent transportation systems.



Ngai Wong (S'98–M'02) received the B.Eng. degree with first class honors and the Ph.D. degree, both in electrical and electronic engineering, from the University of Hong Kong, Pokfulam, Hong Kong, in 1999 and 2003, respectively.

He was an Intern with Motorola, Inc., Kowloon, Hong Kong, from 1997 to 1998, specializing in product testing. He was a Visiting Scholar with Purdue University, West Lafayette, IN, in 2003. Currently, he is an Associate Professor with the Department of Electrical and Electronic Engineering, University of Hong Kong. His current research interests include very large scale integration (VLSI) linear/nonlinear modeling and simulation, model order reduction, digital filter design, sigma-delta modulators, and numerical algorithms in communication and VLSI applications.