

# Constraint Abstraction for Vectorless Power Grid Verification

Xuanxing Xiong and Jia Wang  
Department of Electrical and Computer Engineering  
Illinois Institute of Technology, Chicago, IL 60616, USA  
xxiong3@hawk.iit.edu, jwang@ece.iit.edu

## ABSTRACT

Vectorless power grid verification is a formal approach to analyze power supply noises across the chip without detailed current waveforms. It is typically formulated and solved as linear programs, which demand intensive computational power, especially for large-scale power grids. In this paper, we propose a constraint abstraction technique to reduce the computation cost of vectorless verification. The boundary condition of a subgrid is modeled by boundary constraints, which enable efficient calculation of conservative bounds of power supply noises in a divide-and-conquer manner. Experimental results show that the proposed approach achieves significant speedup over prior art while maintaining good solution quality.

## Categories and Subject Descriptors

B.7.2 [Integrated Circuits]: Design Aids

## General Terms

Algorithms, Verification, Performance

## Keywords

Power grid, voltage drop, vectorless verification

## 1. INTRODUCTION

As technology scaling continues, the performance and reliability of integrated circuits become increasingly susceptible to power supply noises, such as IR drops and Ldi/dt drops in the on-chip power grid. Reduced supply voltage levels in the grid can increase the gate delay, leading to timing violations and even logic failures. In order to ensure a reliable chip design, it is indispensable to verify that the power grid is robust, i.e., the power supply noises are acceptable for all possible runtime situations.

Nowadays, it is common practice to verify power grids by simulation. Typically, an equivalent RC/RLC circuit model of the grid is extracted from the layout, and designers perform simulations to evaluate the power supply noises based on the current waveforms drawn by the circuit. However, simulation-based verification has the following limitations. First, it is computationally prohibitive to simulate all possible current waveforms. In practice, some typical current waveforms are used for verification. Such current waveforms are either too pessimistic, or realistic but we are taking the risk of missing some important corner case. Second,

simulation-based approaches require detailed circuit implementation to provide the current waveforms, thus do not allow power grid verification at an early design stage, when grid correction can be easily incorporated.

To overcome these limitations, *vectorless power grid verification* was proposed in [13], and further studied in many later works. Instead of enumerating all possible current waveforms, it evaluates power supply noises based on partial current specifications defined by *current constraints*. Grid verification is often formulated as linear programs of finding the worst-case power supply noises at each node. The initial study [13] considered the DC analysis model, and it was extended to verify RC power grids in [8]. Since solving the linear programs is very expensive, [3] proposed to generate a reduced-size linear program for each node by using an approximate inverse technique; [17] and [19] designed convex dual algorithms to solve the linear programs efficiently; [1] utilized the dominance relations among nodal voltage drops to reduce the number of linear programs to be solved; [12] and [5] used macromodeling technique to simplify the linear programs for efficient incremental verification. Besides, [6] proposed to verify VDD network and ground network together with additional *current conservation constraints* in order to consider the mutual effects in between.

Moreover, vectorless verification of RLC power grids was proposed in [2]. A fast approach to compute conservative bounds of power supply noises was proposed in [4], and *transient current constraints* were introduced in [20,21] for more realistic noise predictions. In [7], the authors proposed *hierarchical power constraints* to model current excitation, and the resultant linear programs for vectorless verification can be solved by a sorting-deletion algorithm efficiently. Model order reduction was used in subsequent work [16] to further reduce the cost of formulating the linear programs.

Despite these available approaches, the computation cost of vectorless verification remains much higher than that of power grid simulation. Typically, for full-chip power grid verification, both the number of linear programs to be solved and the size of each linear program are proportional to the size of the grid. As a result, the computation cost increases dramatically when the grid size becomes larger. Macromodeling is an effective technique to reduce the computation cost for verifying large-scale power grids, but building the macromodels can also be expensive, and relatively small performance gain can be achieved as studied in [5].

In this paper, we consider vectorless verification of RC power grids following previous studies [3, 17, 19], and propose to reduce the computation cost by *constraint abstraction*. Since a localized region of the power grid can be verified based on the *boundary condition* (i.e., the power supply noises at the neighboring nodes), we propose novel *boundary constraints* to define a partial specification of boundary condition, and perform full-chip grid verification in a

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC'13, May 29 - June 07, 2013, Austin, TX, USA.

Copyright 2013 ACM 978-1-4503-2071-9/13/05 ...\$15.00.

divide-and-conquer manner to compute conservative bounds of power supply noises. The boundary constraints provide a high-level abstraction of the grid environment for the region of interest, and enables dramatic performance gain by significantly reducing the computation cost for verification. Experimental results with both synthetic power grids and IBM power grid benchmarks show that the proposed approach achieves up to about 17X speedup over the prior art [19], and the overestimation of power supply noises is reasonably small. In particular, a power grid with 562K nodes can be verified in about 1 hour.

The rest of the paper is organized as follows. The problem formulation and previous approaches are introduced in Section 2. The constraint abstraction approach is proposed in Section 3. After experimental results are shown in Section 4, we conclude the paper in Section 5.

## 2. PRELIMINARIES

### 2.1 Problem Formulation

Consider vectorless verification of power grids with an RC model, where each branch of the grid is represented by a resistor, and each node is connected to ground through a capacitor. External power supplies are modeled as ideal voltage sources connected to VDD pads, and some nodes have current sources (to ground), which represent the current drawn by the circuit. Let  $n$  be the number of nodes (except VDD pads) in the grid,  $\mathbf{v}(t)$  be the  $n \times 1$  vector of time-varying voltage drops at nodes, and  $\mathbf{i}(t)$  be the  $n \times 1$  vector of time-varying current sources connected to the grid (the elements corresponding to the nodes without current sources attached are constant 0s). Then the system equation of the grid can be written as

$$G\mathbf{v}(t) + C\dot{\mathbf{v}}(t) = \mathbf{i}(t), \quad (1)$$

where  $G$  is the  $n \times n$  conductance matrix, and  $C$  is an  $n \times n$  diagonal matrix of nodal capacitances.

The aforementioned model represents a VDD network of the power grid. In fact, for the ground network, similar model can also be built, and equation (1) still holds if  $\mathbf{v}(t)$  is redefined to be the vector of ground bounces. We refer to voltage drops in the VDD network and ground bounces in the ground network as *voltage noises*. Generally, the relationship between voltage noises  $\mathbf{v}(t)$  and current excitations  $\mathbf{i}(t)$  always satisfies the system equation (1), and voltage noises can be evaluated if some current specification is available.

We employ *current constraints* [13] to capture the feasible set of current excitations, so that power grid can be verified without detailed current waveforms, i.e., a vectorless approach. Specifically, two types of constraints are adopted: *local constraints* and *global constraints*. Local constraints define an upper bound on each current source,

$$0 \leq \mathbf{i}(t) \leq \mathbf{I}_L, \forall t,$$

where  $\mathbf{I}_L \geq 0$  is an  $n \times 1$  vector of peak current values. Global constraints define upper bounds on the sums of groups of current sources, and model the peak current drawn by circuit blocks. Let  $m$  be the number of global constraints, then global constraints can be expressed in matrix form as

$$U\mathbf{i}(t) \leq \mathbf{I}_G, \forall t,$$

where  $U$  is an  $m \times n$  incidence matrix consisting of 0s and

1s, and  $\mathbf{I}_G \geq 0$  is an  $m \times 1$  vector.

Vectorless verification can be performed for either DC analysis model [13] or transient analysis model (with time step  $\Delta t$ ) [8]. As identified in [19], the following maxVN-LCC (maximum Voltage Noise under Linear Current Constraints) problem is the key problem for vectorless verification of RC power grids.

**PROBLEM 1 (MAXVN-LCC).** *Consider an RC power grid of  $n$  nodes with conductance matrix  $G$  and capacitance matrix  $C$ . Let  $A = G$  for DC analysis model or  $A = G + \frac{C}{\Delta t}$  for transient analysis model with time step  $\Delta t$ . Given local and global current constraints with parameters  $\mathbf{I}_L$ ,  $\mathbf{I}_G$ , and  $U$ , solve for each node  $1 \leq j \leq n$ ,*

$$\text{Maximize } v_j \quad \text{s.t.} \quad (2)$$

$$A\mathbf{v} = \mathbf{i}, 0 \leq \mathbf{i} \leq \mathbf{I}_L, U\mathbf{i} \leq \mathbf{I}_G.$$

Here  $\mathbf{v}$  and  $\mathbf{i}$  are the decision variables of voltage noises and current sources, respectively, and  $v_j$  is the  $j$ th element of  $\mathbf{v}$ .

As  $A$  is known to be an  $n \times n$  symmetric positive definite M-matrix, so that  $A$  is invertible,  $A^{-1}$  is also symmetric and satisfies  $A^{-1} \geq 0$ . Then the optimization problem (2) can be decomposed into two sub-problems as follows:

$$\text{I: Compute } \mathbf{c}_j \text{ by solving } A\mathbf{x} = \mathbf{e}_j, \quad (3)$$

$$\text{II: Maximize } v_j = \mathbf{c}_j^T \mathbf{i} \quad \text{s.t.} \quad (4)$$

$$0 \leq \mathbf{i} \leq \mathbf{I}_L, U\mathbf{i} \leq \mathbf{I}_G,$$

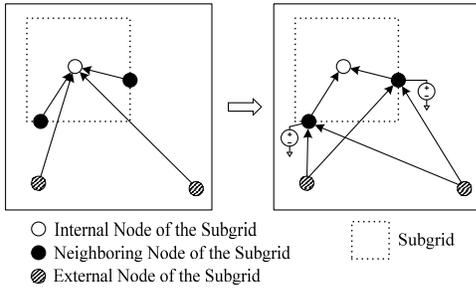
where  $\mathbf{c}_j$  is an  $n \times 1$  vector of coefficients, and  $\mathbf{e}_j$  is an  $n \times 1$  vector of 0s except for its  $j$ th element being 1. It is to be noted that  $\mathbf{c}_j$  is the  $j$ th column of  $A^{-1}$ . The first sub-problem is equivalent to power grid DC analysis with current vector  $\mathbf{e}_j$ , and the second sub-problem is a linear program.

### 2.2 Previous Approaches

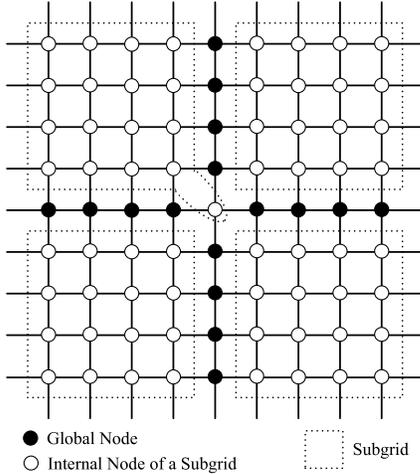
Obviously, a direct and exact approach involves solving sub-problems (3) and (4) with standard linear system solvers (or accurate power grid analysis algorithms) and linear programming (LP) solvers, respectively. However, such an approach consumes too much runtime to be practical for large-scale power grids. In order to expedite full-chip power grid verification, [3] proposed to compute an approximate  $\mathbf{c}_j$  with small number of nonzero elements, so that the linear program (4) can be much simplified and solved efficiently. Later works [17, 19] proposed convex dual algorithms to solve the linear program (4) fast, and utilized a random walk based preconditioned conjugate gradient (PCG) power grid analyzer to calculate  $\mathbf{c}_j$ . The dual algorithms achieve significant speedup over the standard LP solver, and the resultant verification approach provides better runtime efficiency than [3] while ensuring better solution accuracy. However, as the problem sizes of (3) and (4) are proportional to the number of nodes in the power grid, the computation cost of these approaches is often too high to enable time-efficient verification of large-scale power grids. Hence, it is of great interest to reduce the computation cost by exploring more efficient techniques.

## 3. CONSTRAINT ABSTRACTION

### 3.1 Methodology



**Figure 1: Illustration of conventional approaches (left) and the proposed constraint abstraction approach (right) for verifying an internal node inside a subgrid. The arrows represent the logical relation between nodes.**



**Figure 2: A simple partitioned power grid.**

Consider a localized region of the power grid, referred to as a *subgrid*. For each subgrid, we refer to the nodes inside the subgrid as *internal nodes*, the nodes which are outside the subgrid but connected with some internal nodes as *neighboring nodes*, and other nodes in the power grid as *external nodes*.

As illustrated in Figure 1, conventional approaches verify an internal node of a subgrid by considering the whole grid structure, including all neighboring nodes, external nodes, and the corresponding voltage supplies and current sources attached. The corresponding problem sizes of (3) and (4) increase as the size of the power grid increases. In order to verify the subgrid efficiently, we propose to treat neighboring nodes as *uncertain voltage sources* (see Figure 1), which can be modeled by *boundary constraints* (detailed in the next subsection) for realistic scenarios. As a result, the subgrid can be verified based on proper boundary constraints, without involving the external grid structure explicitly. The resultant problem sizes are roughly equal to the size of the subgrid, enabling significant reduction in computation cost. This approach is called *constraint abstraction*, since boundary constraints provide a high-level abstraction of the boundary condition of the subgrid.

### 3.2 Grid Partitioning & Boundary Constraints

We apply constraint abstraction for full-chip power grid verification in a *divide-and-conquer* manner. As illustrated

in Figure 2, the grid is partitioned into several disjoint *subgrids*, which are split by a small set of *global nodes* (i.e., the neighboring nodes of subgrids). Such a partition can be obtained by using the power grid partitioning technique proposed in [18]. Typically, a proper partition results in relatively small number of global nodes, and most nodes are internal nodes of subgrids.

For each subgrid, we use *boundary constraints* to model its boundary condition, i.e., the voltage noises at its neighboring global nodes, which are connected with the internal nodes inside the subgrid. Let  $\hat{m}$  be the number of neighboring global nodes of a subgrid, and  $\mathbf{v}_{ex}$  be the  $\hat{m} \times 1$  vector of voltage noises at these nodes. Then the boundary constraints are represented as

$$0 \leq \mathbf{v}_{ex} \leq \mathbf{v}_\ell, \text{ and } \sum_{1 \leq j \leq \hat{m}} v_{ex,j} \leq v_g, \quad (5)$$

where  $v_{ex,j}$  is the  $j$ th element of  $\mathbf{v}_{ex}$ ,  $\mathbf{v}_\ell$  is an upper bound vector on the voltage noises  $\mathbf{v}_{ex}$ , and  $v_g$  is an upper bound on the sum of these voltage noises.

In practice,  $\mathbf{v}_\ell$  can be either the exact worst-case voltage noises at neighboring global nodes, or some upper bounds, which are computed by verifying global nodes.  $v_g$  can be computed by solving the following linear program.

$$\begin{aligned} & \text{Maximize } \sum_{1 \leq j \leq \hat{m}} v_{ex,j} \quad \text{s.t.} \\ & \mathbf{A}\mathbf{v} = \mathbf{i}, 0 \leq \mathbf{i} \leq \mathbf{I}_L, \mathbf{U}\mathbf{i} \leq \mathbf{I}_G. \end{aligned} \quad (6)$$

Similar to solving the sub-problems (3) and (4) to compute the worst-case voltage noise, we calculate a coefficient vector to represent the objective function as an affine function of current sources. This can be achieved by solving a linear system  $\mathbf{A}\mathbf{x} = \mathbf{e}_g$ , where  $\mathbf{e}_g$  is a vector of 0s except that its elements corresponding to the neighboring global nodes are set to 1. With such a coefficient vector in hand, the optimal value of (6) can be obtained by solving a linear program like (4). Here, the computation cost is equivalent to verifying a node by solving (3) and (4).

Based on the above discussion, a partitioned power grid can be verified in the following steps.

1. For each global node, compute its worst-case voltage noise (or an upper bound of voltage noise).
2. For each subgrid, build boundary constraints, and then evaluate the worst-case voltage noises at internal nodes subject to boundary constraints.

In the first step, as there are only relatively small amount of global nodes, previous methods [3, 17, 19] can be applied. The verification of subgrids in the second step is detailed in the next subsection.

### 3.3 Verification of Subgrids

As summarized in Problem 1, we need to evaluate the worst-case voltage noises based on the generalized system equation  $\mathbf{A}\mathbf{v} = \mathbf{i}$  for both DC and transient models. Consider a subgrid with  $\hat{n}$  internal nodes and  $\hat{m}$  neighboring global nodes. The equation  $\mathbf{A}\mathbf{v} = \mathbf{i}$  is reduced to

$$\begin{bmatrix} A_{in} & A_{ex} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{in} \\ \mathbf{v}_{ex} \end{bmatrix} = \mathbf{i}_{in}, \quad (7)$$

where  $A_{in}$  is the  $\hat{n} \times \hat{n}$  conductance matrix of the subgrid,  $A_{ex}$  is a non-positive  $\hat{n} \times \hat{m}$  matrix representing the conductance links between internal nodes and neighboring global

nodes,  $\mathbf{v}_{in}$  and  $\mathbf{i}_{in}$  are the vectors of voltage noises and current sources at internal nodes, respectively.

Since  $A_{in}$  is also a symmetric positive definite  $\mathbf{M}$ -matrix, it is invertible, and  $A_{in}^{-1}$  is symmetric and non-negative. Then equation (7) can be rearranged as

$$\mathbf{v}_{in} = A_{in}^{-1}\mathbf{i}_{in} - A_{in}^{-1}A_{ex}\mathbf{v}_{ex}. \quad (8)$$

Let  $v_{in,j}$  be the  $j$ th element of  $\mathbf{v}_{in}$ ,  $\mathbf{c}_{in,j}$  be the  $j$ th column of  $A_{in}^{-1}$ , and  $\mathbf{c}_{ex,j}$  be the transpose of the  $j$ th row of  $-A_{in}^{-1}A_{ex}$ . (Note that both  $\mathbf{c}_{in,j}$  and  $\mathbf{c}_{ex,j}$  are non-negative, and  $\mathbf{c}_{ex,j} = -\mathbf{c}_{in,j}^T A_{ex}$ .) We have

$$v_{in,j} = \mathbf{c}_{in,j}^T \mathbf{i}_{in} + \mathbf{c}_{ex,j}^T \mathbf{v}_{ex}, \quad (9)$$

where the internal current vector  $\mathbf{i}_{in}$  is defined by local and global current constraints, and the boundary condition  $\mathbf{v}_{ex}$  is restricted by boundary constraints.

As a result, the subgrid can be verified by solving the following linear program for each internal node  $1 \leq j \leq \hat{m}$ .

$$\begin{aligned} \text{Maximize } v_{in,j} &= \mathbf{c}_{in,j}^T \mathbf{i}_{in} + \mathbf{c}_{ex,j}^T \mathbf{v}_{ex} \quad \text{s.t.} \quad (10) \\ 0 &\leq \mathbf{i}_{in} \leq \hat{\mathbf{I}}_L, \hat{U}\mathbf{i}_{in} \leq \hat{\mathbf{I}}_G, \\ 0 &\leq \mathbf{v}_{ex} \leq \mathbf{v}_\ell, \sum_{1 \leq j \leq \hat{m}} v_{ex,j} \leq v_g, \end{aligned}$$

where  $\hat{\mathbf{I}}_L$ ,  $\hat{\mathbf{I}}_G$  and  $\hat{U}$  are matrices for local and global current constraints related to the subgrid. It is to be noted that the decision variables  $\mathbf{i}_{in}$  and  $\mathbf{v}_{ex}$  are independent, thus (10) can be further decomposed into two linear programs and solved separately.

$$\text{Maximize } \mathbf{c}_{in,j}^T \mathbf{i}_{in} \quad \text{s.t.} \quad 0 \leq \mathbf{i}_{in} \leq \hat{\mathbf{I}}_L, \hat{U}\mathbf{i}_{in} \leq \hat{\mathbf{I}}_G, \quad (11)$$

$$\text{Maximize } \mathbf{c}_{ex,j}^T \mathbf{v}_{ex} \quad \text{s.t.} \quad 0 \leq \mathbf{v}_{ex} \leq \mathbf{v}_\ell, \sum_{1 \leq j \leq \hat{m}} v_{ex,j} \leq v_g. \quad (12)$$

**LEMMA 1.** *Let  $v_{in,j}^*$  be the exact worst-case voltage noise at internal node  $j$  (i.e., the optimal value of the linear program (2)), and  $v_{in,j}^+$  be the worst-case voltage noise computed through constraint abstraction (i.e., the optimal value of the linear program (10)), then  $v_{in,j}^* \leq v_{in,j}^+$ .*

**PROOF.** Let  $\mathbf{i}_{in}^*$  and  $\mathbf{v}_{ex}^*$  be the local current vector inside the subgrid and the voltage noises at neighboring global nodes corresponding to  $v_{in,j}^*$ , respectively, so that

$$v_{in,j}^* = \mathbf{c}_{in,j}^T \mathbf{i}_{in}^* + \mathbf{c}_{ex,j}^T \mathbf{v}_{ex}^*.$$

Let  $\mathbf{i}_{in}^+$  and  $\mathbf{v}_{ex}^+$  be the optimal solution of (10), so that

$$v_{in,j}^+ = \mathbf{c}_{in,j}^T \mathbf{i}_{in}^+ + \mathbf{c}_{ex,j}^T \mathbf{v}_{ex}^+.$$

Clearly, we have  $\mathbf{c}_{in,j}^T \mathbf{i}_{in}^* \leq \mathbf{c}_{in,j}^T \mathbf{i}_{in}^+$ , and  $\mathbf{c}_{ex,j}^T \mathbf{v}_{ex}^* \leq \mathbf{c}_{ex,j}^T \mathbf{v}_{ex}^+$ . It follows that  $v_{in,j}^* \leq v_{in,j}^+$ .  $\square$

This verification approach derives an upper bound of voltage noise at each internal node inside the subgrid. The difference between the computed upper bound  $v_{in,j}^+$  and the exact worst-case voltage noise  $v_{in,j}^*$  is called *overestimation*. As we will show in the experimental results, the amount of overestimation is very small, i.e., the computed voltage noise bound is fairly tight and realistic.

Except for solving (6) to obtain  $v_g$ , verifying the subgrid mainly involves computing  $A_{in}^{-1}$  and  $-A_{in}^{-1}A_{ex}$ , and solving the linear programs (11) and (12) for each internal node. By keeping the size of the subgrid within some

bound, the Cholesky decomposition of  $A_{in}$  can be computed, and then  $A_{in}^{-1}$  can be calculated row by row (or column by column since  $A_{in}^{-1}$  is symmetric) by solving a linear system like (3) through forward and back substitutions. As  $A_{ex}$  is a sparse matrix representing the links between internal nodes and neighboring global nodes, the matrix multiplication  $-A_{in}^{-1}A_{ex}$  can be performed efficiently. In practice, both  $A_{in}^{-1}$  and  $-A_{in}^{-1}A_{ex}$  are not stored explicitly, their rows (i.e.,  $\mathbf{c}_{in,j}^T$  and  $\mathbf{c}_{ex,j}^T$ ) are computed, used and discarded at runtime. The linear program (11) can be solved by standard LP solvers if the subgrid size is within some reasonable bound. As the linear program (12) only has one constraint defining an upper bound of the sum of all variables, it can be efficiently solved by sorting the coefficients in non-increasing order, and setting the corresponding variable to be the maximum feasible value sequentially.

### 3.4 Analysis of Computation Cost

In this subsection, we present the computation advantage of the proposed constraint abstraction approach over a direct one. Since the computation for  $-A_{in}^{-1}A_{ex}$  and (12) can be performed fairly efficiently, the major computation cost for verifying a subgrid is to solve (6), to calculate  $A_{in}^{-1}$ , and to solve linear program (11) for each internal node. In contrast, a direct approach computes  $A^{-1}$  (by solving (3)) and solves linear program (4) for all the nodes.

Suppose the cost of computing the Cholesky decomposition is  $f_1(N)$ , the cost of one forward and one back substitution is  $f_2(N)$ , and the cost of solving a linear program (like (4)) is  $f_3(N)$ , where  $N$  is the size of the matrix, and the number of decision variables in the linear program. Typically,  $f_3(N) \gg f_1(N) \gg f_2(N)$ .

Consider a direct approach with Cholesky decomposition for computing  $A^{-1}$ . The total computation cost is given by

$$f_1(n) + nf_2(n) + nf_3(n). \quad (13)$$

Let  $n_0$  be the number of global nodes,  $n_j, 1 \leq j \leq k$  be the number of internal nodes inside each subgrid, where  $\sum_{0 \leq j \leq k} n_j = n$ . The cost of verifying global nodes with the direct approach is  $f_1(n) + n_0 f_2(n) + n_0 f_3(n)$ . For each subgrid, the cost of solving (6) is  $f_2(n) + f_3(n)$ , the cost of computing  $A_{in}^{-1}$  and solving linear program (11) is  $f_1(n_j) + n_j f_2(n_j) + n_j f_3(n_j)$ . Hence, the computation cost of the proposed approach can be approximated as

$$\begin{aligned} &f_1(n) + (n_0 + k)f_2(n) + (n_0 + k)f_3(n) \\ &+ \sum_{1 \leq j \leq k} (f_1(n_j) + n_j f_2(n_j) + n_j f_3(n_j)). \end{aligned} \quad (14)$$

Expressions (13) and (14) provide a rough estimation of computation costs based on the size of the power grid and its subgrids. Generally, the time complexity of Cholesky decomposition is  $O(N^3)$ , the time complexity of forward and back substitutions is  $O(N^2)$ , and the complexity of linear programming by standard solvers can be in even higher order (e.g.,  $O(N^4)$ ). In practice, the sparsity of the conductance matrix  $A$  (as well as the constraint matrix  $U$ ), combined with efficient reordering, enables actual computation costs to be less than such theoretical bounds, but the cost of the direct approach still remains greater than quadratic. Hence, the computation cost of the proposed approach will be much smaller than that of the direct approach if the power grid is partitioned properly.

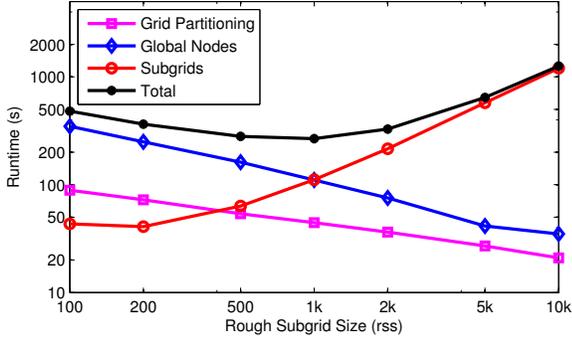


Figure 3: Runtime for verifying a synthetic power grid “pg4000” (90,643 nodes) with different rough subgrid sizes ( $rss$ ).

## 4. EXPERIMENTS

### 4.1 Experimental Setup

The proposed constraint abstraction approach has been implemented in C++. hMETIS [10] is utilized to partition the power grid into subgrids with relatively small number of global nodes. We use a user-specified *rough subgrid size* called  $rss$  to control the size of subgrids, and the number of subgrids is calculated by  $\text{round}(\frac{n}{rss})$ , where  $n$  is the number of nodes (after merging nodes connected by shorts) in the grid. The DualVN algorithm proposed in [19] is employed to verify global nodes, and to solve the linear program (6) for building boundary constraints, where its error tolerance for solving linear programs is set to be 0.1mV. CHOLMOD [15] and GotoBLAS2 [9] are employed to solve the involved linear systems through Cholesky factorization, and MOSEK [14] is used to solve the linear program (11) for verifying each subgrid. We choose the simplex method of MOSEK, since it is slightly faster than the interior point method for our experiments.

For performance comparison, we implemented the DirectVN algorithm of [19] by solving linear system (3) and linear program (4) with CHOLMOD and MOSEK, respectively. The DualVN algorithm [19] has also been implemented for verifying the whole power grid, while linear system (3) is solved by CHOLMOD. Experiments are carried out on a 64-bit Linux server with 2.67GHz Intel Xeon X5650 processor and 64GB memory. Although the server has 12 cores, only one core is used for experiments.

We employ the synthetic power grids used in [19] for performance tests, and also experiment with IBM power grid benchmarks [11]. As IBM power grids have multiple networks, we choose to verify the ground network in our experiments, while their VDD networks can be verified separately. Local current constraints are extracted from the grid description, and global current constraints are generated by scaling down the total amount of current drawn by groups of current sources. For each power grid, we specify 4 global constraints in our tests.

### 4.2 Exploring Rough Subgrid Size

Since the computation cost of the proposed approach is highly dependent on the sizes of subgrids as discussed in Section 3.4, we experiment with different rough subgrid sizes  $rss$  (ranging from 100 to 10K) for performance analysis.

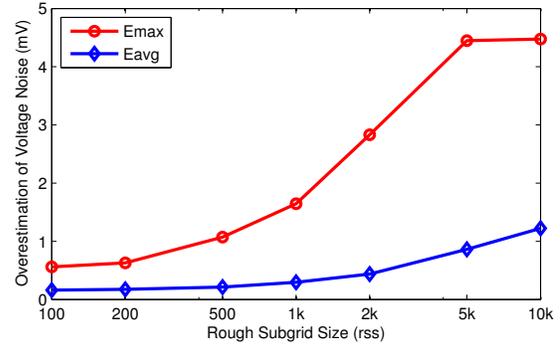


Figure 4: Solution accuracy of verifying a synthetic power grid “pg4000” (90,643 nodes) with different rough subgrid sizes ( $rss$ ).  $E_{max}/E_{avg}$ : the maximum/average overestimation of voltage noises in mV.

Note that the actual sizes of most subgrids would be slightly smaller than  $rss$ , because global nodes do not belong to any subgrid. The total runtime of the proposed approach can be roughly divided into three parts: the runtime to partition the power grid, the runtime to solve global nodes, and the runtime to verify subgrids. Figure 3 plots these runtime components for verifying a synthetic power grid with different  $rss$ . As  $rss$  increases, both the number of subgrids and the number of resultant global nodes decrease, so it takes less runtime to partition the power grid, and to verify global nodes. On the contrary, both the subgrid sizes and the total number of internal nodes increase as  $rss$ , thus solving the subgrids consumes more runtime when  $rss$  becomes larger. Obviously, there is a tradeoff between the computation costs of global nodes and subgrids. If the runtime for grid partitioning is sufficiently small, then the minimum runtime is achieved at the best tradeoff point, where the runtime of global nodes and the runtime of subgrids are roughly equal, e.g., the optimal  $rss$  is 1K for the test case shown in Figure 3.

Recall that constraint abstraction results in overestimation of the worst-case voltage noises as stated in Lemma 1. We also evaluate the effects of different  $rss$  on the amount of overestimation. As shown in Figure 4, both the maximum and the average overestimation increase when a larger  $rss$  is specified. This phenomenon is due to the fact that the boundary constraints become less effective when the subgrid sizes become larger.

To determine a proper  $rss$ , both runtime and solution accuracy need to be considered. Fortunately, the overestimation for most  $rss$  settings are sufficiently small, so that the  $rss$  which provides the best runtime efficiency can be employed for production runs.

### 4.3 Performance Results

The performance data are presented in Table 1. The runtime of DirectVN and DualVN are reported under columns 4 and 5 for performance comparison. The golden solution is produced by DirectVN for analysis of overestimation. Since DirectVN takes too much runtime to verify large power grids, the reported data with  $\approx$ s are estimations from 1000 random nodes. The DualVN algorithm is fairly efficient, and can solve the largest grid “pg10000” with 562K nodes in 19 hours. The proposed constraint abstraction approach further improves the runtime efficiency of vectorless verifica-

**Table 1: Performance Results of the Proposed Constraint Abstraction Approach.**  $n$ : the number of nodes after merging the nodes connected by shorts;  $v_{max}$ : the maximum voltage noise across the grid in mV;  $r_{ss}$ : rough subgrid size;  $n_0$ : the number of global nodes;  $E_{max}/E_{avg}$ : the maximum/average overestimation of voltage noises in mV; the runtime units “s”, “m”, “h” and “d” represent seconds, minutes, hours, and days, respectively.

Power Grids			DirectVN	DualVN [19]	Constraint Abstraction					Speedup Relative to	
Name	$n$	$v_{max}$	Runtime	Runtime	$r_{ss}$	$n_0$	$E_{max}$	$E_{avg}$	Runtime	DirectVN	DualVN
pg1000	5875	46.31	2.66 m	15.40 s	200	546	2.32	0.42	6.57 s	24.31	2.35
pg2000	22939	39.91	53.47 m	2.52 m	500	1459	1.51	0.28	33.22 s	96.58	4.55
pg2500	35668	28.80	2.06 h	5.85 m	500	2287	0.94	0.16	1.00 m	123.61	5.85
pg3000	51195	43.63	4.83 h	12.27 m	500	3392	0.81	0.19	1.71 m	168.98	7.16
pg4000	90643	54.38	15.93 h	38.73 m	1000	4270	1.65	0.29	4.16 m	<b>229.72</b>	<b>9.31</b>
pg5000	141283	≈45.91	≈2.15 d	1.25 h	1000	6855	≈1.02	≈0.44	7.48 m	≈ <b>413.33</b>	<b>10.02</b>
pg10000	562363	≈23.11	≈44.07 d	18.58 h	2000	19427	≈1.88	≈1.44	1.10 h	≈ <b>963.08</b>	<b>16.92</b>
ibmpg1	10242	677.67	4.26 m	26.09 s	200	775	6.16	0.37	12.84 s	19.89	2.03
ibmpg2	65228	357.92	3.95 h	16.65 m	500	4513	5.00	0.44	2.45 m	96.90	6.81
ibmpg3	150687	≈179.91	≈3.47 d	1.29 h	1000	6853	≈1.58	≈0.42	8.93 m	≈ <b>559.49</b>	<b>8.70</b>
ibmpg4	478094	≈3.42	≈14.12 d	15.37 h	2000	20103	≈0.19	≈0.12	1.24 h	≈ <b>273.33</b>	<b>12.40</b>
ibmpg5	291382	≈42.44	≈19.51 d	6.13 h	1000	13674	≈0.59	≈0.12	29.29 m	≈ <b>959.33</b>	<b>12.56</b>
ibmpg6	430337	≈109.96	≈55.97 d	12.47 h	2000	11975	≈1.20	≈0.57	46.27 m	≈ <b>1741.73</b>	<b>16.17</b>

tion, achieving up to about 17X speedup over DualVN. As a result, “pg10000” can be verified in about 1 hour. Generally, grid partitioning takes a few seconds to a few minutes depending on the grid size and the number of desired subgrids. Most runtime of the proposed approach is spent on solving global nodes and subgrids. Moreover, for most test cases, the maximum overestimation is within 3mV, and the average overestimation is much smaller than 1mV. In addition, it is worth noting that the proposed approach is more effective when the grid size becomes larger, making it suitable for large-scale power grid verification.

The major merit of constraint abstraction is that it enables significant reduction in computation cost with small amount of overestimation. In practice, it can be used for fast estimation of power supply noises across the chip. The risky regions can be identified in small amount of runtime, and then more accurate methods can be applied if it is necessary.

## 5. CONCLUSION

In this paper, we presented constraint abstraction for vectorless verification of power grids. Boundary constraints are built to model the boundary condition of subgrids, enabling efficient verification of power grids in a divide-and-conquer manner. Experimental results confirmed the efficiency of the proposed approach, and it may also be extended to transient verification of RLC power grids.

## 6. REFERENCES

- [1] N. Abdul Ghani and F. N. Najm. Power grid verification using node and branch dominance. In *Proc. Design Automation Conf. (DAC)*, pages 682–687, Jun. 2011.
- [2] N. H. Abdul Ghani and F. N. Najm. Handling inductance in early power grid verification. In *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, pages 127–134, Nov. 2006.
- [3] N. H. Abdul Ghani and F. N. Najm. Fast vectorless power grid verification using an approximate inverse technique. In *Proc. Design Automation Conf. (DAC)*, pages 184–189, Jul. 2009.
- [4] N. H. Abdul Ghani and F. N. Najm. Fast vectorless power grid verification under an RLC model. *IEEE Trans. Computer-Aided Design*, 30(5):691–703, May 2011.
- [5] Abhishek and F. N. Najm. Incremental power grid verification. In *Proc. Design Automation Conf. (DAC)*, pages 151–156, Jun. 2012.
- [6] M. Avci and F. N. Najm. Early P/G grid voltage integrity verification. In *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, pages 816–823, Nov. 2010.
- [7] 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. Physical Design (ISPD)*, pages 159–166, Mar. 2011.
- [8] I. A. Ferzli, F. N. Najm, and L. Kruse. A geometric approach for early power grid verification using current constraints. In *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, pages 40–47, Nov. 2007.
- [9] GotoBLAS2. <http://www.tacc.utexas.edu/tacc-projects/gotoblas2>.
- [10] hMETIS. <http://glaros.dtc.umn.edu/gkhome/metis/hmetis/overview>.
- [11] IBM Power Grid Benchmarks. <http://dropzone.tamu.edu/pli/pgbench/>.
- [12] D. Kouroussis, I. A. Ferzli, and F. N. Najm. Incremental partitioning-based vectorless power grid verification. In *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, pages 358–364, Nov. 2005.
- [13] D. Kouroussis and F. N. Najm. A static pattern-independent technique for power grid voltage integrity verification. In *Proc. Design Automation Conf. (DAC)*, pages 99–104, Jun. 2003.
- [14] MOSEK. <http://www.mosek.com/>.
- [15] SuiteSparse. <http://www.cise.ufl.edu/research/sparse/suitesparse/>.
- [16] Y. Wang, X. Hu, C.-K. Cheng, G. K. H. Pang, and N. Wong. A realistic early-stage power grid verification algorithm based on hierarchical constraints. *IEEE Trans. Computer-Aided Design*, 31(1):109–120, Jan. 2012.
- [17] X. Xiong and J. Wang. An efficient dual algorithm for vectorless power grid verification under linear current constraints. In *Proc. Design Automation Conf. (DAC)*, pages 837–842, Jun. 2010.
- [18] X. Xiong and J. Wang. A hierarchical matrix inversion algorithm for vectorless power grid verification. In *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, pages 543–550, 2010.
- [19] X. Xiong and J. Wang. Dual algorithms for vectorless power grid verification under linear current constraints. *IEEE Trans. Computer-Aided Design*, 30(10):1469–1482, Oct. 2011.
- [20] X. Xiong and J. Wang. Vectorless verification of RLC power grids with transient current constraints. In *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, pages 548–554, 2011.
- [21] X. Xiong and J. Wang. Verifying RLC power grids with transient current constraints. *IEEE Trans. Computer-Aided Design*, 2013.