



A Multilevel Spectral Framework for Scalable Vectorless Power/Thermal Integrity Verification

ZHIQIANG ZHAO and ZHUO FENG, Stevens Institute of Technology, USA

Vectorless integrity verification is becoming increasingly critical to the robust design of nanoscale integrated circuits. This article introduces a general vectorless integrity verification framework that allows computing the worst-case voltage drops or temperature (gradient) distributions across the entire chip under a set of local and global workload (power density) constraints. To address the computational challenges introduced by the large power grids and three-dimensional mesh-structured thermal grids, we propose a novel spectral approach for highly scalable vectorless verification of large chip designs by leveraging a hierarchy of almost linear-sized spectral sparsifiers of input grids that can well retain effective resistances between nodes. As a result, the vectorless integrity verification solution obtained on coarse-level problems can effectively help compute the solution of the original problem. Our approach is based on emerging spectral graph theory and graph signal processing techniques, which consists of a graph topology sparsification and graph coarsening phase, an edge weight scaling phase, as well as a solution refinement procedure. Extensive experimental results show that the proposed vectorless verification framework can efficiently and accurately obtain worst-case scenarios in even very large designs.

CCS Concepts: • **Hardware** → **Electronic design automation**; **Very large scale integration design**; **Power and energy**; **Power and thermal analysis**;

Additional Key Words and Phrases: Vectorless integrity verification, spectral graph theory, spectral graph coarsening, graph signal processing

ACM Reference format:

Zhiqiang Zhao and Zhuo Feng. 2022. A Multilevel Spectral Framework for Scalable Vectorless Power/Thermal Integrity Verification. *ACM Trans. Des. Autom. Electron. Syst.* 28, 1, Article 11 (December 2022), 25 pages. <https://doi.org/10.1145/3529534>

1 INTRODUCTION

Aggressive VLSI technology scaling has led to ever-increasing power densities as well as temperatures on chip, imposing grand challenges in designing **integrated circuit (IC)** systems [31]. For example, the rapidly increasing power dissipations and aggressively lowering supply voltages result in a massive amount of the current drawn from the power supply, which in turn make power grid integrity verification tasks increasingly challenging [8, 11, 12, 15, 16, 19, 27, 33, 39]. Meanwhile, higher power densities also inevitably raise chip temperatures, which further lead to

This work is supported in part by the National Science Foundation under Grants CCF-2041519 (CAREER), CCF-2021309 (SHF), and CCF-2011412 (SHF).

Authors' address: Z. Zhao and Z. Feng, Stevens Institute of Technology, 1 Castle Point Terrace, Hoboken, New Jersey 07030; emails: {zzhao76, zhuo.feng}@stevens.edu.

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. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

1084-4309/2022/12-ART11 \$15.00

<https://doi.org/10.1145/3529534>

(1) greater power grid IR drops and interconnect RC delays, (2) higher leakage power consumption caused by the exponentially increasing sub-threshold currents, (3) slower devices caused by the degraded carrier mobility, and (4) shorter device life and poorer package reliability. Therefore, it is necessary for compute-intensive full-chip thermal analysis and integrity verification: At circuit level, estimating temperature variations and peak temperature across the chip are essential for accurate timing and power analysis of digital designs [31], whereas evaluating peak temperature and temperature gradients for critical circuit modules becomes increasingly important for reducing mismatches and improving performance of analog and mixed-signal circuits [23]; at system level, thermal modeling and analysis can be leveraged to guide dynamic voltage and frequency scaling for reducing thermal violations, achieving desired temperature levels and thereby reducing workload runtimes [7].

Traditional vector-based power grid and thermal integrity verification methods rely on running numerous circuit simulations using over-pessimistic input vectors to locate the worst-case voltage or thermal profiles. However, vector-based methods require the underlying workloads or power densities to be known in advance [17, 22, 36], which may not always be practical. For example, at early chip design phase it is usually impossible to obtain a precise estimation of underlying power densities due to the lack of accurate workloads modeling. As a result, vector-based power grid and thermal verification methods may not provide useful guidance for improving the chip design performance (reliability).

To address the aforementioned limitations, vectorless verification methods have been proposed as alternatives, which have been adopted for power grid integrity verifications in recent years. Recent methods have exploited optimization approaches for finding the worst-case scenarios under given workload constraints [8, 11, 12, 15, 16, 19, 27, 33, 39], leveraging sparse approximate inverse technique [15], efficient dual algorithm [39], and node elimination [16]. Despite these significant improvements, the overall power grid verification cost can still be extremely high, especially for large-scale tasks. To significantly improve the efficiency, scalable multilevel vectorless verification methods based on **geometric multigrid (GMG)** operations and the PDE-constrained optimization framework have been proposed [11, 12, 21]. However, such methods require the underlying power grid designs to have (nearly) regular structures so that GMG operations can be performed effectively, which may become a major limitation when applied to nanoscale PDN designs with highly irregular power grid structures. Motivated by the existing GMG-based multilevel vectorless verification methods, Reference [42] introduces a more versatile multilevel vectorless verification framework exploiting the recent graph-theoretic **algebraic multigrid (AMG)** algorithmic framework [24] as well as a hierarchy of spectrally sparsified power grids.

Motivated by the recent vectorless methods [11, 42, 44], this work introduces a general multilevel vectorless verification framework applicable to both power and thermal integrity verification problems to enable efficient estimations of worst-case voltage drop or thermal profiles under complex power or workload constraints (uncertainties). Our approach leverages a recent spectral graph reduction framework for generating a hierarchy of spectral graph sparsifiers of decreasing sizes [43], an iterative edge weight scaling scheme and a solution refinement procedure. The main contribution of this work is briefly summarized as follows:

- (1) We propose a more general multilevel framework for vectorless power and thermal integrity verification that allows estimating nearly worst-case scenarios under various kinds of complex workload or power density uncertainties and constraints.
- (2) We introduce a novel power (thermal) grid simplification method based on recent spectral graph sparsification and coarsening techniques as well as graph signal processing for achieving good scalability for large problems.

- (3) Extensive experimental results have been demonstrated for large-scale power and thermal integrity verification tasks considering various power density (workload) constraints, as well as the flexible tradeoffs between the verification cost and solution quality.

Comparing to the previous works [42, 44], the following innovations have been made: (1) We propose a general multilevel framework for vectorless integrity verification that is applicable to input problems with different structures and densities, including power grids and thermal problems; (2) a spectral graph coarsening framework is leveraged for input grid simplification; and (3) the proposed framework is evaluated on large-scale power and thermal integrity verification tasks considering various power density (workload) constraints, as well as the flexible tradeoffs between the verification cost and solution quality.

The rest of this article is organized as follows. Section 2 provides a brief introduction to prior IC power grid and thermal modeling and analysis methods as well as emerging spectral graph sparsification and graph signal processing techniques. In Section 3, the proposed scalable vectorless integrity verification method is described in details. Section 4 demonstrates extensive experiment results for both power grids and thermal chip designs with various power density (workload) constraints, which is followed by the conclusion of this work in Section 5.

2 BACKGROUND

2.1 Vectorless Power Grid Integrity Verification

The chip steady-state analysis of an n -node power grid can be formulated into following equation by utilizing nodal analysis [11, 15]:

$$T x = b, \quad (1)$$

where T represents a conductance matrix representing all the interconnected resistors in the grid, x is $n \times 1$ node voltage vector, and b is the right-hand-side current vector;

Traditional vectorless power grid integrity verification aims to identify the maximum voltage drops or current densities under linear current constraints [12, 15], where current constraints are introduced to capture current loading variations and correlations in given chip designs. Two types of constraints are considered in a typical vectorless verification problem: local constraints for setting the lower and upper bounds of the power density for each source while global constraints for setting the lower and upper bounds for blocks of sources.

Given the power grid, the proposed vectorless integrity verification tasks compute the maximum voltage drop by solving the following **linear programming (LP)** for each individual node i :

$$\begin{aligned} & \text{maximize : } x_i = e_i^\top T^{-1} b, \text{ for } i = 1, \dots, n \\ & \text{subject to:} \\ & \text{local constraints : } b^L \leq b \leq b^U, \\ & \text{global constraints : } B^L \leq M b \leq B^U, \end{aligned} \quad (2)$$

where n is the number of nodes in the power grid and e_i is an elementary unit vector with i th entry to be 1 and others being zeros. Since the conductance matrix T is an M -matrix, the T^{-1} only contains non-negative sensitivity values. The b^L (B^L) and b^U (B^U) represent the lower bounds and upper bounds of individual power sources (blocks), while M is an $m \times n$ matrix that only contains 0s and 1s for defining m global (block) constraints. After getting the worst-case vector b_{wst} through the above optimization procedure, we can simply compute the maximum voltage drop x_{wst} using $x_{wst} = e_i^\top T^{-1} b_{wst}$.

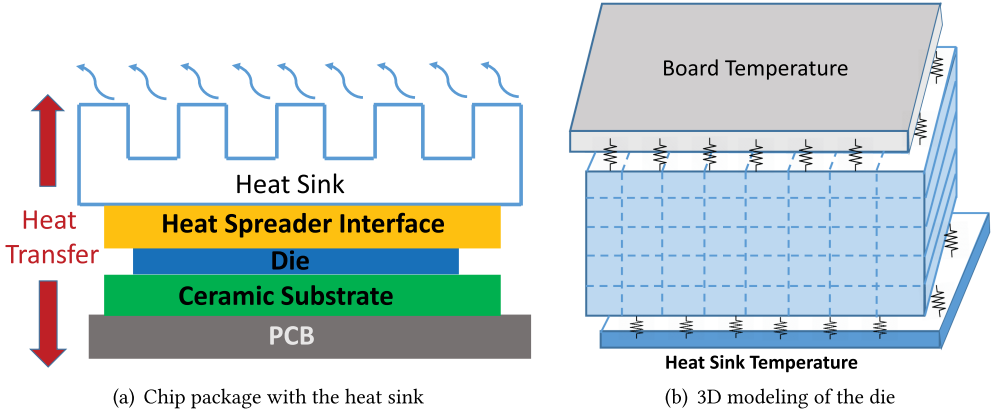


Fig. 1. Thermal modeling of the chip package.

2.2 Vectorless Thermal Integrity Verification

A diagram of an IC chip in a C4 package is shown in Figure 1(a), showing two major heat transfer paths: one is through the heat sink to the ambient surroundings, and the other is from the chip package to the board. The equivalent thermal circuit of the die is usually modeled as a three-dimensional (3D) mesh grid with thermal conductance computed according to the materials as well as a discretization scheme, as shown in Figure 1(b). The heat diffusion in an IC can be modeled by the following PDE equation [22, 30]:

$$\rho c_p \frac{\partial F(\vec{r}, t)}{\partial t} = \nabla(k(\vec{r}, t) \nabla F(\vec{r}, t)) + p(\vec{r}, t), \quad (3)$$

subject to the boundary condition:

$$k(\vec{r}, F) \frac{\partial F(\vec{r}, t)}{\partial n_j} + h_j F(\vec{r}, t) = f_j(\vec{r}, t), \quad (4)$$

where ρ is the material density (kg/m^3), c_p is the specific heat [$\text{J}/(\text{kg} \cdot ^\circ\text{C})$], F is the temperature ($^\circ\text{C}$), \vec{r} is the location in the 3D space, k is the thermal conductivity of the material [$\text{W}/(\text{m}^2 \cdot ^\circ\text{C})$], $p(\vec{r}, t)$ is the power density of heat sources (W/m^3), n_j is the outward direction normal to the boundary surface j , h_j is the heat transfer coefficient [$\text{W}/(\text{m}^2 \cdot ^\circ\text{C})$], and f_j is an arbitrary function at the surface j . Similarly to the power grid analysis, the chip steady-state analysis of an n -node thermal grid can be formulated as Equation (1) [22, 44], while T becomes the thermal conductance matrix of the 3D thermal grid, b is the right-hand-side vector modeling the underlying workload (power density) distribution, and x becomes the unknown temperature vector to be computed.

Vectorless thermal integrity verification seeks to identify the maximum temperature or temperature gradient by solving LP problems similarly to the ones in Equation (2). However, *factorization of the thermal matrix* obtained from 3D mesh-structured grids can be much more costly than factorizing the conductance matrices for power grid vectorless verification tasks [40, 42], due to the high computational complexity of existing direct solution methods, such as LU and Cholesky decomposition methods [9]. For example, our results show that factorizing a matrix with one million rows (columns) using the state-of-the-art Cholesky solver [9] may take over 30 minutes and consume 18 GB memory.

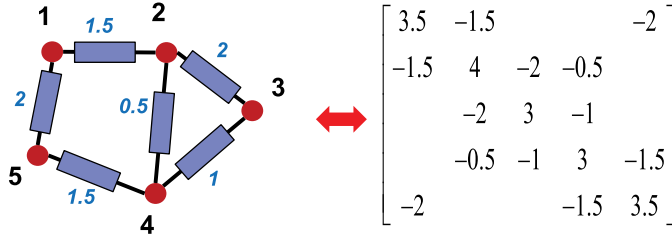


Fig. 2. A resistor network (conductance value of each element is shown) and its graph Laplacian matrix.

2.3 Relationship between Power Grid and Thermal Integrity Verification

The steady-state vectorless power grid and thermal integrity verification are essentially similar which target on solving the linear programming problems formulated in Equation (2). The main difference is that the formulated conductance matrix T has different structures and densities for power grids and thermal grids. For example, T will be much denser for thermal grids compared to the power grids, thus harder to solve in real applications. Previous works, such as References [11, 12, 16, 21], only work on specific types of problems due to different structures and densities of the input grids. In this work, we introduce a general framework for vectorless integrity verification regardless of the input grid types. This is realized by utilizing the graph simplification techniques (spectral sparsification and coarsening) in the framework, which can handle large-scale power grid and thermal integrity verification tasks.

2.4 Spectral Sparsification and Graph Signal Processing

For a weighted, undirected graph $G = (V_G, E_G, w_G)$, where E_G represents the edge set of G , V_G represents the vertex set, and w_G is a function that gives every edge a positive weight, the following Laplacian matrix can be defined:

$$L_G(p, q) = \begin{cases} -w_G(p, q) & \text{if } (p, q) \in E_G \\ \sum_{(p, t) \in E_G} w_G(p, t) & \text{if } (p, q) = (p, p) \\ 0 & \text{if otherwise} \end{cases} \quad (5)$$

It can be shown that the graph Laplacian matrix is a symmetric diagonally dominant matrix, which can be considered as an admittance matrix of a resistive circuit network [37], as is shown in Figure 2. Given any real vector $x \in \mathbb{R}^{|V_G|}$, the quadratic form of graph G can be defined as

$$x^T L_G x = \sum_{(p, q) \in E_G} w_G(p, q) (x(p) - x(q))^2. \quad (6)$$

Spectral graph sparsification aims to find an ultra-sparse graph proxy (sparsifier) that has much fewer edges but the same set of nodes and similar spectral properties (e.g., Laplacian eigenvalues and eigenvectors) as the original graph. A sparsifier $P = (V_G, E_P, w_P)$ is said to be σ -spectrally similar to the original graph G if they have similar quadratic forms, or the following condition is satisfied for all real vectors $x \in \mathbb{R}^{|V_G|}$ [1], as shown in Figure 3,

$$\frac{x^T L_P x}{\sigma} \leq x^T L_G x \leq \sigma x^T L_P x. \quad (7)$$

The relative condition number can be computed by

$$\kappa(L_G, L_P) = \lambda_{\max} / \lambda_{\min} \leq \sigma^2, \quad (8)$$

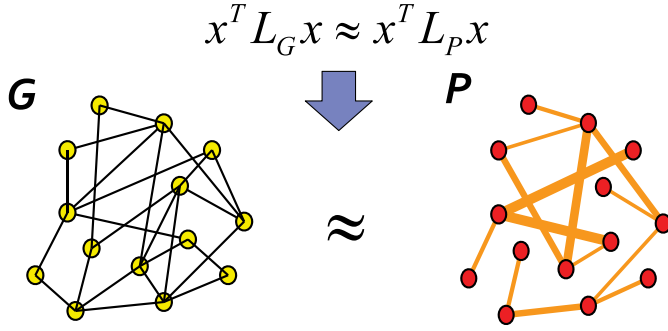


Fig. 3. Two spectrally similar graphs G and P.

where λ_{\max} (λ_{\min}) is the largest (smallest non-zero) eigenvalue of matrix $L_P^+ L_G$ with L_P^+ denoting the Moore–Penrose pseudoinverse of L_P matrix. It has been shown that a spectrally similar sub-graph can approximately preserve the distances or effective resistances between vertices [38] so that much faster graph-based algorithms can be developed based on these “spectrally” sparsified networks. Also, a graph sparsifier with a smaller relative condition number (higher spectral similarity) can lead to faster convergence when used as preconditioners in iterative methods, such as Krylov-subspace iterative methods.

There is an analogy between traditional signal processing or classical Fourier analysis and graph signal processing [35]: (1) The signals at different time points in classical Fourier analysis correspond to the signals at different nodes in an undirected graph; (2) The more slowly oscillating functions in time domain correspond to the graph Laplacian eigenvectors associated with lower eigenvalues or the more slowly varying (smoother) components across the graph. The recent spectral graph sparsification process [13, 14] aims to maintain as few as possible edges for preserving the slowly varying or “low-frequency” signals of the original graphs, which therefore can be regarded as a “low-pass” graph filter. As a result, spectrally sparsified graphs will be able to preserve the eigenvectors associated with low eigenvalues more accurately than high eigenvalues.

2.5 Spectral Graph Coarsening

Compared to the solid theoretical works on the graph sparsification, graph coarsening is harder to understand due to the lack of matured theory support. Different coarsening methods have been proposed and studied in the past decades, but most of them are based on heuristics. One widely used coarsening algorithm is match-based graph coarsening. For example, METIS [18], which has been widely used for graph partitioning and embedding, is a heavy edge matching-based graph coarsening method; another popular way for graph coarsening is based on AMG-inspired schemes [24, 34], which usually forms the Galerkin operator for generating the coarsened graphs; Reference [4] recently introduces a graph neural network-based framework to learn the edge weights of the coarsened graphs. Among existing coarsening methods, spectral graph coarsening has been proven to be highly effective due to the preservation of key graph spectral (structural) properties [25, 26]. A variety of spectral graph coarsening schemes have been proposed in recent years: Reference [10] proposed a Kron reduction scheme based on Schur complement; Purohit et al. [32] introduced CoarseNet to coarsen graphs while preserving the largest eigenvalue of its adjacency matrix such that the diffusion characteristics of the original graph can be kept; Loukas and Vnderghenst [25] proposed a theoretical framework based on restricted spectral similarity, which is a modification of the previous spectral similarity metric for spectral graph sparsification; Bravo-Hermsdorff and Gunderson [2] proposed a probabilistic framework for graph coarsening,

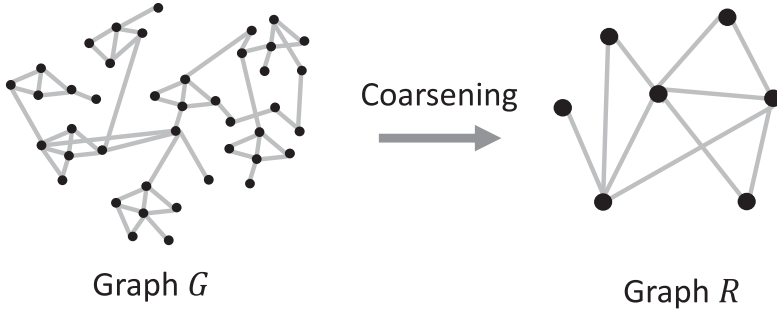


Fig. 4. An example illustrating the graph coarsening process.

with the goal of preserving the inverse Laplacian of the coarsened graph; and Reference [46] introduces a spectral coarsening and scaling algorithm for preserving the first few eigenvalues and eigenvectors of the original graph Laplacian matrix.

3 A SPECTRAL APPROACH TO VECTORLESS INTEGRITY VERIFICATION

To aggressively simplify circuit networks (e.g., the 3D thermal grids) and thereby addressing the computational challenges in vectorless integrity verification without sacrificing the approximation accuracy, in this work emerging graph signal processing and spectral graph algorithms [14, 35, 43] will be leveraged for vectorless power and thermal integrity verification tasks. For example, since full chip temperature distributions can be considered as the “low-frequency” graph signals on thermal grids obtained after applying a “low-pass” graph filter on the original input power sources, the spectrally coarsened thermal grids will be able to well preserve the original temperature distributions and thus facilitate vectorless thermal verification tasks.

3.1 Graph Coarsening via Local Spectral Embedding

Graph coarsening tries to find a smaller graph $R = (V_R, E_R, w_R)$ to approximate the original graph $G = (V_G, E_G, w_G)$ with the graph mapping operator $H_G^R \in \mathbb{R}^{|V_R| \times |V_G|}$:

$$L_R = H_G^R L_G (H_G^R)^\top, \quad (9)$$

where H_G^R is a coarsening matrix containing only 0 and 1. Figure 4 shows an example illustrating the spectral graph coarsening process. The coarsening process can be considered as a surjective mapping of the original node set: $(H_G^R)_{p,q} = 1$ if node q in graph G is aggregated to super-node p in graph R , and $(H_G^R)_{p',q} = 0$ for all nodes $p' \in \{v \in R : v \neq p\}$. The reduced graph R and the original graph G satisfy restricted spectral similarity shown as the following condition [25, 26]:

$$\begin{aligned} \frac{1}{\sigma'} \|x\|_{L_G} &\leq \|x_R\|_{L_R} \leq \sigma' \|x\|_{L_G}, \\ \forall x_R &\in U^R, \quad \forall x_G \in U^G, \end{aligned} \quad (10)$$

where $U^R = [u_R^{(1)}, u_R^{(2)}, \dots, u_R^{(k)}]$ and $U^G = [u_G^{(1)}, u_G^{(2)}, \dots, u_G^{(k)}]$ include the first k eigenvectors of L_R and L_G correspondingly.

One of the properties for the mapping operator is that it is a locality-preserving operator, so how to construct the H_G^R is the key problem for preserving important spectral properties (e.g., the first few eigenvalues and eigenvectors of the graph Laplacian matrix). One possible way is to directly find node clusters on the low-dimensional representatives of graph G by performing spectral graph embedding with the first k eigenvectors of L_G . However, it requires the eigenvector calculations

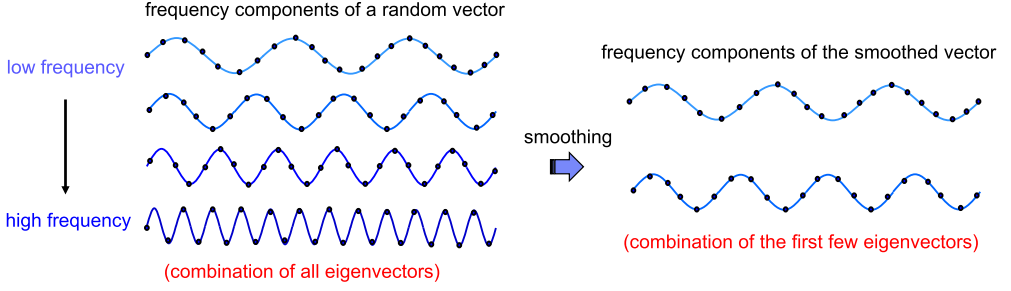


Fig. 5. Smoothing a random vector on a path graph.

for the original graph Laplacians, which is not feasible for very large graphs. In this work, we leverage an efficient yet effective local spectral embedding scheme to identify node clusters based on emerging graph signal processing techniques [35].

So far we know that many standard iterative methods, such as Gauss–Seidel and Jacobi methods, possess the smoothing property [3]. This property allows these methods effectively eliminating the high-frequency or oscillatory components of the signal while leaving the low-frequency or smooth components relatively unchanged. From the perspective of graph signal analysis, these iterative methods are low-pass graph filtering functions that can quickly filter out the “high-frequency” components of the random graph signal or the eigenvectors corresponding to high eigenvalues of the graph Laplacian [35].

To better understand the concept, an example is demonstrated by smoothing a random vector on a path graph, as shown in Figure 5. We consider a random vector (graph signal) x that can be expressed with a linear combination of eigenvectors u_i , for $i = 1, \dots, n$, of a path-graph Laplacian, shown as follows:

$$x = \sum_{i=1}^n \alpha_i u_i. \quad (11)$$

After applying the smoothing function on x , a smoothed vector \tilde{x} can be obtained in linear time by only preserving slow-varying signals, which can be considered as a linear combination of the eigenvectors corresponding to the first few bottom eigenvalues, shown as follows:

$$\tilde{x} = \sum_{i=1}^k \tilde{\alpha}_i u_i, \quad k \ll n. \quad (12)$$

In this work, we apply a few (e.g., 5 to 10) Gauss–Seidel iterations for solving the linear system of equations $L_G \mathbf{x}^{(i)} = \mathbf{0}$ to a set of k initial random vectors $\mathbf{K} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k)})$ that are orthogonal to the all-one vector $\mathbf{1}$ satisfying $\mathbf{1}^\top \mathbf{x}^{(i)} = 0$. Based on the smoothed vectors in \mathbf{K} , each node is embedded into a k -dimensional space such that nodes p and q are considered spectrally similar if their low-dimensional embedding vectors $\mathbf{x}_p \in \mathbb{R}^k$ and $\mathbf{x}_q \in \mathbb{R}^k$ are highly correlated. Consequently, spectrally similar nodes p and q can be then aggregated together for node reduction purpose. Here the node distance is measured by the spectral node affinity $a_{p,q}$ for neighboring nodes p and q [5, 24]:

$$a_{p,q} = \frac{\|(\mathbf{K}_{p,:}, \mathbf{K}_{q,:})\|^2}{(\mathbf{K}_{p,:}, \mathbf{K}_{p,:})(\mathbf{K}_{q,:}, \mathbf{K}_{q,:})}, \quad (13)$$

$$\text{with } (\mathbf{K}_{p,:}, \mathbf{K}_{q,:}) = \sum_{i=1}^k (\mathbf{x}_p^{(i)} \cdot \mathbf{x}_q^{(i)}).$$

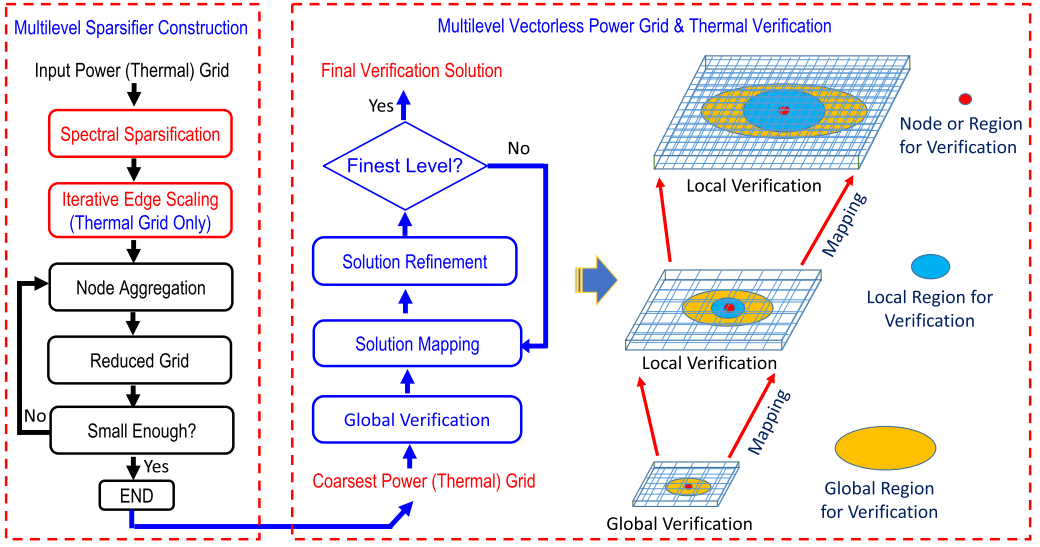


Fig. 6. Multilevel vectorless power grid and thermal integrity verification.

3.2 A Multilevel Vectorless Verification Framework

In this work, we propose a multilevel vectorless verification framework shown in Figure 6. Our approach is based on the latest graph coarsening algorithm [24, 43, 45] for generating coarse-level (sparsified) grids according to the original power (thermal) grid problem, as well as the recent multilevel vectorless power grid integrity verification framework [11, 42]. The proposed framework includes two phases: a multilevel sparsifier construction phase as well as a multilevel vectorless verification phase, which are described as follows:

Phase I: *Multilevel sparsifier construction*

- Apply spectral sparsification for the input grid to generate initial grid sparsifier.
- Perform iterative edge weight scaling on the initial sparsifier to compensate the loss of edges. The detailed introduction for scaling procedure is illustrated in Section 3.3
- Apply spectral graph coarsening on the scaled sparsifier for generating multilevel grid sparsifiers.
- Map power constraints to each coarsened level by leveraging coarsening mapping operator.
- Factorize the sparsifiers at each level for adjoint sensitivity computations.

Phase II: *Multilevel vectorless verification*

- Calculate the adjoint sensitivities at each level with the matrix factors.
- Identify a global critical region on the coarsest grid and local critical regions on each finer level.
- Perform vectorless verification on the global critical region at the coarsest level.
- Map solution vector to next finer level and improve the solution accuracy by performing local solution refinements on local critical regions at each level until reaching the finest level.

3.3 Spectral Graph Sparsification and Edge Scaling

A perturbation-based spectral graph sparsification engine [13, 14] is leveraged to dramatically sparsify the topology of the original grid during the vectorless verification. The spectral

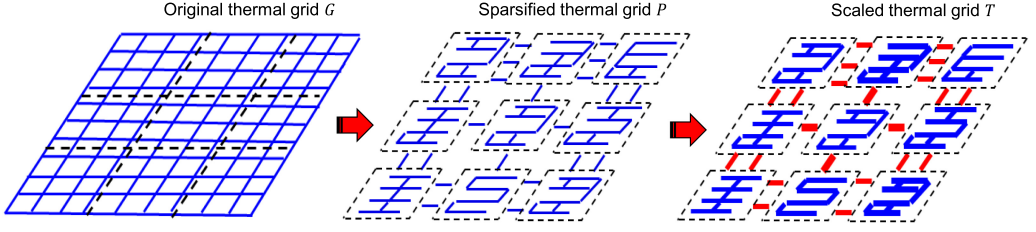


Fig. 7. Iterative edge scaling for sparsified thermal grids.

sparsification step will efficiently remove the redundant edges in the input grid and only keep the edges that are spectrally critical to the structure of the graph. In doing so, it can effectively control the multilevel grid densities while maintaining good spectral approximation quality, such as effective resistances between nodes. It is noted that preserving effective resistance is equivalent to preserving the adjoint sensitivities to be applied for setting up LPs, which is the key for accurate vectorless verification. Meanwhile, the treelike structures of the sparsified grids will immediately reduce the matrix factorization time. The low-degree nodes in the sparsifier grid can be potentially merged together to further reduce the number of variables in LPs. Therefore, both the matrix factors and adjoint sensitivities can be computed in a more efficient way without sacrificing the final solution quality (e.g., worst-case vector).

ALGORITHM 1: Algorithm for Iterative Edge Weight Scaling

Input: The error tolerance ϵ , the number of partitions k , the original graph G and the initial spectrally sparsified graph $P^{(0)}$.

Output: The spectrally sparsified graph P with scaled edge weights.

- 1: Generate a random vector \mathbf{b} that is orthogonal to the all-one vector.
 - 2: Partition the original graph G into k blocks P_1, P_2, \dots, P_k using multilevel graph partitioning method [20].
 - 3: Let graph $P = P^{(0)}$.
 - 4: Construct matrices $T_G = L_G + g_{min}I$ and $T_P = L_P + g_{min}I$ by adding a small value g_{min} similar to the ambient thermal conductance to each diagonal entry of L_G and L_P for graph signal filtering purpose.
 - 5: Solve $T_G \mathbf{x} = \mathbf{b}$ and $T_P \tilde{\mathbf{x}} = \mathbf{b}$ and compute $\mathbf{err} = \frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|}$.
 - 6: **while** $\mathbf{err} > \epsilon$ **do**
 - 7: **for** partition P_i , $i = 1, \dots, k$, **do**
 - 8: calculate $y_i = \sum_{t \in P_i} \mathbf{x}[t]$, $\tilde{y}_i = \sum_{t \in P_i} \tilde{\mathbf{x}}[t]$, and $\alpha_i = \frac{\tilde{y}_i}{y_i}$ for all nodes;
 - 9: **end for**
 - 10: **for** each edge $(p, q) \in E_P$ **do**
 - 11: **if** $p, q \in P_i$, scale up $w_P(p, q)$ by a factor of α_i ;
 - 12: **if** $p \in P_i$ and $q \in P_j$ with $i \neq j$, scale up $w_P(p, q)$ by a factor of $(\alpha_i + \alpha_j)/2$;
 - 13: **end for**
 - 14: Update graph P , L_P and T_P with the latest edge weights;
 - 15: Solve $T_P \tilde{\mathbf{x}} = \mathbf{b}$ and update the mismatch $\mathbf{err} = \frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|}$;
 - 16: **end while**
 - 17: Return the latest spectrally sparsified graph P .
-

Motivated by recent graph signal processing techniques [35], an iterative edge weight scaling scheme is leveraged to gradually scale up the edge weight in the sparsified thermal grid, as shown in Figure 7. This scheme will compensate for the thermal conductance loss due to the missing edges

by matching the “low-frequency” behaviors of the original thermal grids, such that approximation quality after spectral sparsification can be further improved.

We define the non-decreasing eigenvalues and corresponding eigenvectors of the Laplacian matrix L_G to be $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ and u_1, u_2, \dots, u_n , respectively. The spectral decomposition of L_G can be expressed as follows:

$$L_G = \sum_{i=1}^n \lambda_i u_i u_i^\top, \quad (14)$$

while the spectral decomposition of L_G^+ can be expressed as follows:

$$L_G^+ = \left(\sum_{i=2}^n \lambda_i u_i u_i^\top \right)^+ = \sum_{i=2}^n \frac{1}{\lambda_i} u_i u_i^\top. \quad (15)$$

By adding a small grounded thermal conductance g_{min} to each node in graph G , which is equivalent to add a small value g_{min} to each diagonal element in L_G , we have the following:

$$T_G = L_G + g_{min}I = \sum_{i=1}^n (g_{min} + \lambda_i) u_i u_i^\top, \quad (16)$$

where $I = \sum_{i=1}^n u_i u_i^\top$ is the identify matrix. Similarly to Equation (15), the spectral decomposition of T_G^{-1} can be expressed as follows:

$$T_G^{-1} = (L_G + g_{min}I)^{-1} = \sum_{i=2}^n (g_{min} + \lambda_i)^{-1} u_i u_i^\top. \quad (17)$$

Given a random vector b , it can be expressed with the Laplacian eigenvectors as follows:

$$b = \sum_{i=1}^n \beta_i u_i^\top. \quad (18)$$

To get the solution of $T_G x = b$, we can have $x = T_G^{-1} b$, which can be further expressed as follows:

$$\begin{aligned} x &= (L_G + g_{min}I)^{-1} b = \left(\sum_{i=1}^n (g_{min} + \lambda_i) u_i u_i^\top \right)^{-1} b \\ &= \sum_{i=1}^n \frac{1}{g_{min} + \lambda_i} u_i u_i^\top b = \frac{1}{g_{min}} \sum_{i=1}^n \frac{\beta_i}{1 + r \lambda_i} u_i, \end{aligned} \quad (19)$$

where $r = 1/g_{min}$. Equation (19) indicates that when a small g_{min} is applied, the eigenvectors associated with small eigenvalues or only “low-frequency” components in b will remain in x ; however, a relatively large g_{min} ($r \approx 0$) will allow more higher frequencies to be included in x and thus lead to $x \approx b$.

Based on the above analysis, T_G^{-1} can be considered as a “low-pass” filter matrix for a given graph signal b : By properly choosing g_{min} values it is possible to only keep “low-frequency” components in x , while the graph signal’s “high-frequency” (highly oscillating) components will be filtered out. Since chip temperature distributions mainly contain “slowly varying” components due to relative small ambient thermal conductance values, which can be viewed as “low-frequency” signals, it becomes possible to leverage emerging spectral sparsification techniques [13, 14] to only maintain a small number of edges in the sparsified thermal grids while still preserving accurate thermal profiles, since spectrally sparsified graphs can very well preserve “low-frequency” graph signals.

Based on the above intuition, Algorithm 1 is proposed for scaling up edge weights in the sparsified thermal grid by matching the “low-frequency” responses filtered by the original thermal grids.

3.4 Solution Refinement

We define the non-decreasing eigenvalues and corresponding eigenvectors of the Laplacian matrix L_P to be $0 = \tilde{\lambda}_1 \leq \tilde{\lambda}_2 \leq \dots \leq \tilde{\lambda}_n$ and $\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_n$, respectively, where graph P is a sparsified graph of graph G . The spectral decomposition of L_P can be expressed as follows:

$$L_P = \sum_{i=1}^n \tilde{\lambda}_i \tilde{u}_i \tilde{u}_i^\top. \quad (20)$$

Assume that k smallest eigenvalues and corresponding eigenvectors of L_G can be well preserved in L_P , while the remaining higher eigenvalues and eigenvectors are not. Then the spectral decomposition of L_P can be approximately written as follows:

$$L_P \approx \sum_{i=1}^k \lambda_i u_i u_i^\top + \sum_{k+1}^n \tilde{\lambda}_i \tilde{u}_i \tilde{u}_i^\top. \quad (21)$$

Based on Equation (19), the solution from the original grid can be expressed as

$$x = (L_G + g_{min}I)^{-1}b = \sum_{i=1}^n \frac{u_i u_i^\top b}{g_{min} + \lambda_i}. \quad (22)$$

While the identify matrix I can be written as follows:

$$I = \sum_{i=1}^n \tilde{u}_i \tilde{u}_i^\top \approx \sum_{i=1}^k u_i u_i^\top + \sum_{i=k+1}^n \tilde{u}_i \tilde{u}_i^\top. \quad (23)$$

Similarly, the solution \tilde{x} obtained with L_P can be written as follows:

$$\begin{aligned} \tilde{x} &= (L_P + g_{min}I)^{-1}b = \sum_{i=1}^n \frac{\tilde{u}_i \tilde{u}_i^\top b}{g_{min} + \tilde{\lambda}_i} \\ &\approx \sum_{i=1}^k \frac{u_i u_i^\top b}{g_{min} + \lambda_i} + \sum_{i=k+1}^n \frac{\tilde{u}_i \tilde{u}_i^\top b}{g_{min} + \tilde{\lambda}_i}. \end{aligned} \quad (24)$$

Based on Equations (22) and (24), the solution error due to spectral sparsification and scaling becomes

$$\Delta x = x - \tilde{x} \approx \sum_{i=k+1}^n \left(\frac{u_i u_i^\top b}{g_{min} + \lambda_i} - \frac{\tilde{u}_i \tilde{u}_i^\top b}{g_{min} + \tilde{\lambda}_i} \right). \quad (25)$$

From Equation (25) we can see that the solution error using spectrally sparsified graphs can be expressed as a linear combination of high eigenvectors corresponding to large Laplacian eigenvalues, which can also be viewed as a linear combination of high-frequency signals on graphs [35]. To further improve the solution obtained on sparsified grids, “low-pass” graph signal filters, such as weighted Jacobi iterative method, can be utilized to filter out the high-frequency error. The solution refinement method is described in Algorithm 2. The inputs include the input grid conductance matrix T_G that can be decomposed into a diagonal matrix D_G and the remainder matrix Q_G , the solution vectors $\tilde{x}_1, \dots, \tilde{x}_k$ obtained on the sparsified conductance matrix T_P , the right-hand-side vectors b_1, \dots, b_k as well as the weight factor γ and iteration number N_{max} for signal filtering.

ALGORITHM 2: Solution Refinement Algorithm

Input: $T_G = D_G + Q_G$, $\tilde{x}_1, \dots, \tilde{x}_k$, b_1, \dots, b_k , γ , N_{max} ;

- 1: For each of the approximate solution vector \tilde{x}_j in $\tilde{x}_1, \dots, \tilde{x}_k$, do
- 2: **for** $i = 1$ **to** N_{max} **do**
- 3: $\tilde{x}_j^{(i+1)} = (1 - \gamma)\tilde{x}_j^{(i)} + \gamma D_G^{-1}(b_j - Q_G \tilde{x}_j^{(i)})$
- 4: **end for**
- 5: Return the smoothed solution vectors $\tilde{x}_1, \dots, \tilde{x}_k$.

3.5 Example: A Two-level Verification Framework

In the following, we will describe a two-level vectorless verification approach, while multilevel schemes can be defined in a similar way.

Local and global constraints mapping. Instead of directly solving the linear programming problem in Equation (2), we first reduce the input grid to a much smaller one by the multilevel graph coarsening scheme. At the same time, the power constraints on the original problem needs to be modified after the grid coarsening on each level. Assume the power constraints on level h to be $(b^U)^h$ (upper constraints) and $(b^L)^h$ (lower constraints). Specifically, we have $(b^U)^0 = b^U$ and $(b^L)^0 = b^L$ when $h = 0$. More generally, power constraints can be directly mapped from fine level h to coarse level $h + 1$ with graph mapping operator H_h^{h+1} obtained as follows:

$$\begin{aligned} \text{upper bound : } (b^U)^{h+1} &= H_h^{h+1} (b^U)^h, \\ \text{lower bound : } (b^L)^{h+1} &= H_h^{h+1} (b^L)^h, \end{aligned}$$

where $(b^U)^{h+1}$, $(b^L)^{h+1}$, $(b^U)^h$, and $(b^L)^h$ denote the upper bound and lower bound of power sources for coarse and fine grids, respectively. The global constraints mapping can be defined in a similar manner by choosing the global constraints on the coarse grid to be the sum of each block's lower and upper bounds on the fine grid.

Solution mapping and refinement. To reduce the verification cost on the coarse level, the global critical region C_{glb} will be identified based on the adjoint sensitivity threshold [11], such that C_{glb} will only include the most important power sources. Given a sensitivity threshold ϵ_{th} , we will only include the power sources that have adjoint sensitivity values greater than ϵ_{th} into C_{glb} for setting up LPs:

$$\begin{aligned} \text{maximize : } x_{wst}^{h+1} &= \sum_{\forall b_i^{h+1} \in C_{glb}} s_i^{h+1} b_i^{h+1} \\ \text{subject to local and global constraints:} & \\ \text{local : } (b^L)^{h+1} &\leq b^{h+1} \leq (b^U)^{h+1}, \\ \text{global : } (B^L)^{h+1} &\leq M^{h+1} b^{h+1} \leq (B^U)^{h+1}. \end{aligned} \tag{26}$$

The solution b_{wst}^{h+1} will be further mapped back to the fine level using the mapping operator $(H_h^{h+1})^\top$ by

$$\tilde{b}_{wst}^h = (H_h^{h+1})^\top b_{wst}^{h+1}. \tag{27}$$

Since directly mapping the solution of the coarse level problem to a finer level may lead to increased solution error, a local solution refinement procedure at the finer level becomes essential as suggested in Reference [11]. To this end, we propose to incrementally improve the solution quality on the finer grid by setting up a new LP for a much smaller local critical region and combining the updated LP solution with mapped coarse-level solution to gain good efficiency and accuracy. The

key steps in the proposed solution mapping and refinement procedure for a vectorless integrity verification problem are as follows:

- (1) Set the normalized sensitivity threshold $\epsilon_{loc} = \beta \epsilon_{glb}$ with the scaling factor $\beta > 1$ to obtain a much smaller local critical region C_{loc} ;
- (2) Set up a new LP problem for the local critical region C_{loc} to obtain the solution vector \bar{b}_{wst}^h .
- (3) For the current sources (variables) that belong to C_{loc} , update their solution with \bar{b}_{wst}^h ; for the sources (variables) that belong to C_{glb} but not C_{loc} , reuse the mapped solution \bar{b}_{wst}^h .
- (4) Compute the refined solution by: $x_{wst} = \tilde{x}_{wst} + \epsilon_{glb} \times s_{max} |b^h - \bar{b}_{wst}^h|_1$.

3.6 Algorithm Flow and Complexity Analysis

The detailed multilevel vectorless integrity verification algorithm has been described in Algorithm 3. The complexity for graph coarsening is $O(m)$ with m denoting the number of resistors in the chip model. The complexity for spectral sparsification and edge scaling is $O(m \log n)$ with n denoting the number of nodes in the grid, which is nearly linear. The cost for solving LPs will depend on the algorithm to be adopted as well as the sizes of critical regions for setting up the LPs, which can be well controlled by the proposed multilevel verification framework. Since only ultra-sparse (treelike) spectral sparsifiers of the original input grids are needed for vectorless verification by leveraging the proposed solution refinement procedure, the proposed method can significantly improve the overall algorithm scalability, as shown in our experiment results in Section 4.

ALGORITHM 3: Multilevel Vectorless Integrity Verification

Input: original power or thermal grid, user-defined local and global power constraints b^U , b^L and M , initial normalized sensitivity threshold ϵ_{th} , and sensitivity scaling factor $\beta > 1$

Output: worst-case voltage drop or thermal profile of the original input grid.

- 1: Extract spectrally sparsified grid for the original input grid.
 - 2: For thermal grid, update sparsified grid using iterative edge weight scaling method (Algorithm 1).
 - 3: Multilevel coarse grid construction:
 - (a) Construct all hierarchy levels from finest to coarsest level;
 - (b) Get local and global power constraints b^U , b^L and M for each level using coarsening operators.
 - 4: Factorize each coarse-level grid for adjoint sensitivity calculation.
 - 5: Perform global verification at the coarsest level K :
 - (a) Find global critical region C_{glb}^K for a given sensitivity threshold ϵ_K , and set up LP to get worst case vector b_{wst}^K
 - 6: Perform solution mapping and refinement on finer to finest levels:
 - 7: $k \leftarrow K$
 - 8: **while** $k > 1$ **do**
 - 9: Map solution vector to finer level by: $\tilde{b}_{wst}^{k-1} = (H_{k-1}^k)^\top b_{wst}^k$
 - 10: Set sensitivity threshold $\epsilon_{k-1} = \beta \epsilon_k$ and identify C_{loc}^{k-1} .
 - 11: Setup a new LP for C_{loc}^{k-1} to obtain solution vector \bar{b}_{wst}^{k-1} .
 - 12: Combine the latest LP and interpolated solutions to form b_{wst}^{k-1} .
 - 13: $k \leftarrow k - 1$
 - 14: **end while**
 - 15: Calculate the worst-case voltage drop or thermal distribution using the worst-case power source vector.
-

Table 1. Statistics of Two Microprocessor Designs

Design	Processor A	Processor B
Power Consumption (W)	28	50
Die Area (mm ²)	195	302
Num. of Metal Layers	4	6
Num. of Material Layers	11	15
Equivalent Heat Transfer Coefficients (10 ³ W/m ² K)	3.3 (heat sink)	3.3 (heat sink)
	2.0 (package)	2.0 (package)

Table 2. Specifications of the Power Grid Test Cases and Thermal Test Cases

Power Grid Specs.				Thermal Grid Specs.			
CKT	N.#	P.#	L.#	CKT	N.#	P.#	L.#
<i>ibmpg3</i>	0.85M	90K	2	T1	25K	2.5K	2
<i>ibmpg4</i>	1.0M	100K	2	T2	0.1M	10K	2
<i>ibmpg6</i>	1.7M	170K	2	T3	0.2M	10K	2
<i>ibmpg7</i>	1.5M	150K	2	T4	0.4M	40K	2
<i>thupg1</i>	5.0M	500K	3	T5	0.9M	90K	2
<i>thupg2</i>	9.0M	900K	3	T6	1.6M	0.16M	2
—	—	—	—	T7	2.0M	0.20M	2

4 EXPERIMENTAL RESULTS

In this section, we present the experiment results of the proposed vectorless power grid verification method on different power grid designs and thermal verification method for two microprocessor designs [22]. The proposed multilevel vectorless integrity verification method has been implemented in MATLAB and C++. The LP problems are solved by the state-of-the-art LP solver [29] and all experiment results have been obtained using a single CPU core of a computing platform running 64-bit RHEL 6.0 with 2.67-GHz 12-core CPU and 48-GB DRAM memory.

4.1 Experimental Setup

The test cases used for power grid verification include industrial power grid designs with different sizes up to 9 million nodes [28, 41]. The design details of the two microprocessors used for generating the thermal grids are shown in Table 1. The original design of the two microprocessors is from Reference [22], and we reuse the same design in this work. The heat conductance paths are modeled using equivalent heat transfer coefficients.

The specifications of the power grid and thermal test cases are shown in Table 2, where *N.#* and *P.#* denote the numbers of grid nodes and power sources, respectively. *L.#* represents the number of levels generated when using the multilevel verification methods. When setting up experiments, three methods for vectorless power grid and thermal integrity verification are applied, including single level (direct) method [6], multilevel method without (w/o) sparsification, and the proposed multilevel method with (w/) sparsification.

4.2 Experimental Results for Power Grid Verification

4.2.1 Result of Solution Quality. As we mentioned in the previous sections, the spectral graph sparsification method can well preserve the effective resistances of the original power grid, which will guarantee a good solution quality during vectorless verifications. Figure 8 shows very

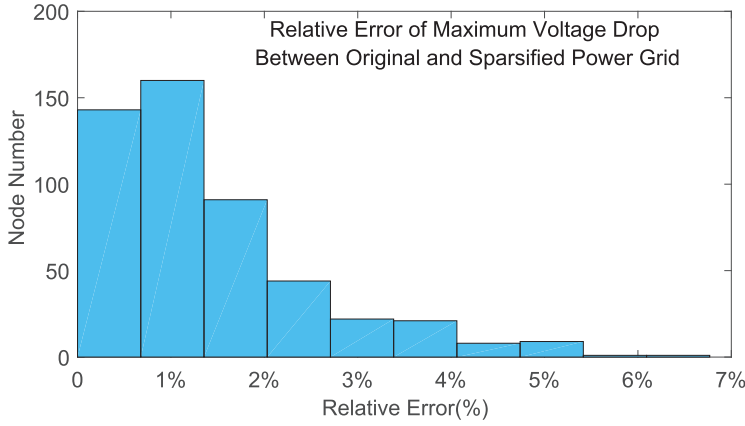


Fig. 8. Relative errors of vectorless verification w/ sparsified grid.

Table 3. Results of the Proposed Vectorless Power Grid Integrity Verification Method

CKT	Single Level			Multilevel w/o Sparsification				Multilevel w/ Sparsification				
	T_{chol}^o	T_{sol}^o	T_{lp}^o	T_{chol}^m	T_{sol}^m	T_{lp}^m	Err(%)	T_{chol}^s	T_{sol}^s	T_{lp}^s	Err(%)	κ
<i>ibmpg3</i>	17.4 s	1.39 s	1.79 s	37.12 s	0.87 s	0.21 s	2.34%	1.40 s	0.029 s	0.11 s	2.58%	27
<i>ibmpg4</i>	22.4 s	1.57 s	2.10 s	48.04 s	1.16 s	0.35 s	3.42%	3.72 s	0.06 s	0.08 s	2.26%	39
<i>ibmpg6</i>	16.3 s	2.84 s	3.19 s	30.30 s	0.67 s	0.45 s	1.23%	5.41 s	0.10 s	0.20 s	2.27%	24
<i>ibmpg7</i>	30.3 s	2.44 s	3.15 s	66.50 s	1.57 s	0.32 s	3.98%	4.86 s	0.08 s	0.15 s	1.00%	36
<i>thupg1</i>	92.2 s	9.20 s	11.43 s	433.54 s	5.06 s	27.15 s	1.00%	11.09 s	0.21 s	10.03 s	1.00%	42
<i>thupg2</i>	963.2 s	43.16 s	282.30 s	2527.46 s	398.05 s	45.10 s	1.00%	47.23 s	0.46 s	17.93 s	1.00%	41

satisfactory results for the vectorless verifications with the sparsified power grids (single level), where relative errors are reported for vectorless verification results when using the original power grid and the sparsified power grid.

4.2.2 Result of Runtime Efficiency. Two parts are included for adjoint sensitivity calculation of the vectorless power grid verification: a matrix factorization phase and a linear equation solving phase using matrix factors. For example, for a given power grid conductance matrix T_G , to calculate the voltage sensitivity given a node i with respect to all current sources, the following procedure will be leveraged: First, a unit vector b will be set up with only the i th entry being 1 and other entries being 0. Then, the linear system of equation $T_G x = b$ will be solved based on matrix factors. Once the solution x is obtained, it can be used as the sensitivity vector for setting up the LPs. It should be noted that we only need to factorize the conductance matrix once for each level and matrix factors can be reused many times during the vectorless verification process. Since the conductance matrix of a sparsified grid can be factorized and solved in almost linear time, adjoint sensitivity computations based upon sparsified grid can be much more efficient than the original grid problem. As shown in Figure 9, the runtime results of sensitivity calculation for the original and sparsified power grids (single level) have been illustrated, where the runtime for Cholesky matrix factorization and linear equation solving are reported. It shows that sensitivity calculations are much faster with the sparsifier grids.

Comprehensive verification results using different approaches are shown in Table 3, with speedup numbers shown in Table 4. “Single Level,” “Multilevel w/o Sparsification,” and “Multilevel w/ Sparsification” denote the verification methods using single level (direct), multilevel method without and with sparsification methods, respectively; T_{chol}^* , T_{sol}^* , and T_{lp}^* denote the runtime for Cholesky factorization, matrix resolve using matrix factors and total LP solve including all levels,

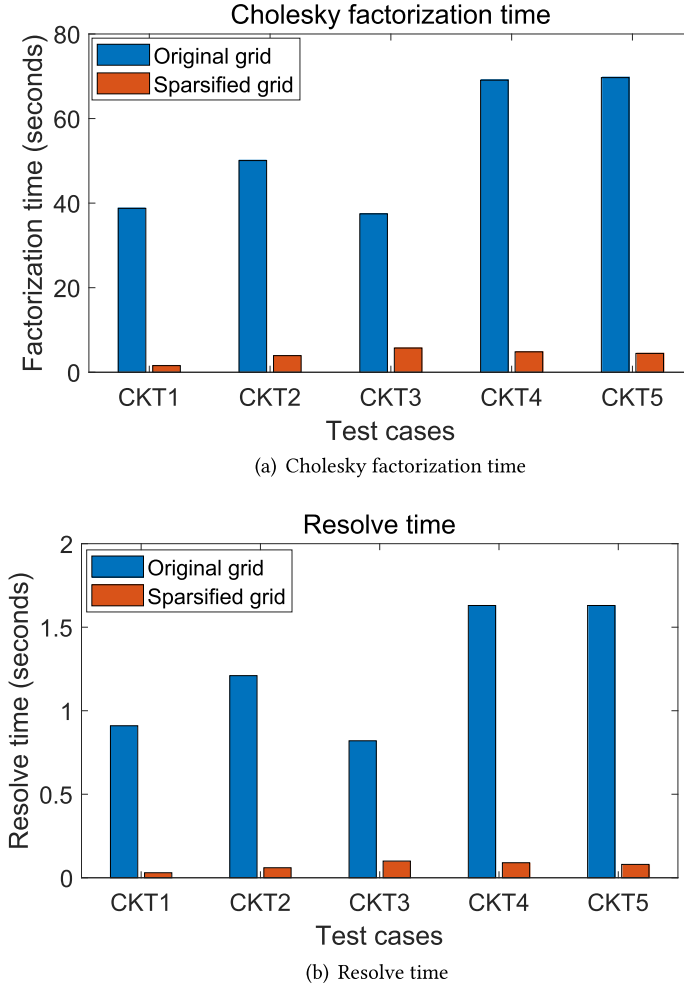


Fig. 9. Sensitivity computation time for the original and sparsified grids.

Table 4. Runtime Results of the Proposed Method

CKT	Sp. Over Single Level			Sp. Over Original Multilevel		
	$\frac{T_{chol}^o}{T_{chol}^s}$	$\frac{T_{sol}^o}{T_{sol}^s}$	$\frac{T_{lp}^o}{T_{lp}^s}$	$\frac{T_{chol}^m}{T_{chol}^s}$	$\frac{T_{sol}^m}{T_{sol}^s}$	$\frac{T_{lp}^m}{T_{lp}^s}$
<i>ibmpg3</i>	12.4×	47.8×	16.3×	26.5×	30×	2.0×
<i>ibmpg4</i>	6.0×	26.2×	26.3×	12.9×	19.3×	4.4×
<i>ibmpg6</i>	3.0×	28.4×	16.0×	5.6×	6.7×	2.3×
<i>ibmpg7</i>	6.2×	30.5×	21.0×	13.7×	19.6×	2.1×
<i>thupg1</i>	8.3×	43.8×	1.1×	40.0×	24.1×	2.7×
<i>thupg2</i>	20.4×	93.8×	15.7×	53.7×	865×	2.5×

respectively; *Err* denotes the relative error of maximum voltage drop compared to single level method, and κ denotes the relative condition number.

For all test cases, verification using “Multilevel w/ Sparsification” method is the fastest among all the methods. Meanwhile, huge speedups for Cholesky factorization and matrix resolve are

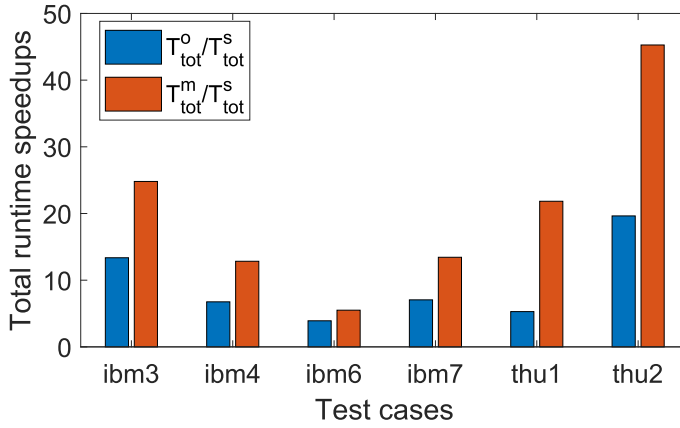


Fig. 10. Total runtime speedups of Multilevel w/ Sparsification method comparing to the other two methods.

achieved when comparing to the other two methods while maintaining good verification accuracy. It is noted that Cholesky factorization and matrix resolve time for “Multilevel w/o Sparsification” method are slowest among all three methods. It is observed that solving LPs using the proposed method is over $2\times$ faster than using the “Multilevel w/o Sparsification” method, showing that the proposed method can play very important roles in reducing overall computational cost during vectorless verifications, especially for large power grid designs.

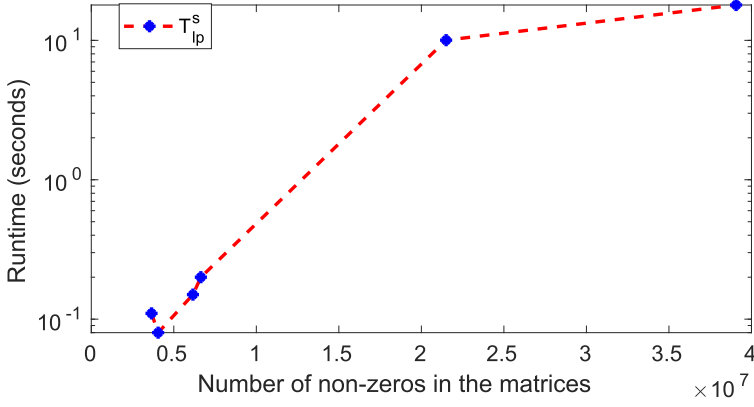
Figure 10 shows the total runtime speedups of Multilevel w/ Sparsification method comparing to the other two methods, indicating the dramatically speedups when comparing to the other two methods.

Figure 11 shows the nearly linear runtime scalability of the proposed method, where both the LP solve time and total runtime have been demonstrated.

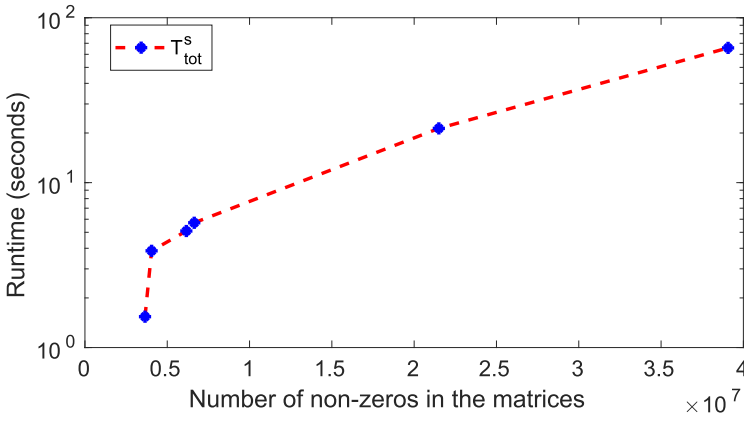
4.2.3 Tradeoff Analysis between Accuracy and Efficiency. Figure 12 shows how vectorless verification solution quality (error) and the LP runtime will change with different relative condition numbers (κ). As observed in our experimental results, the relative errors grow rather slowly with increasing condition numbers ($\kappa < 200$), while the LP solution time can be dramatically reduced. The larger condition number represents a sparser power grid sparsifier, while the sparser power grid will result in the coarsened grids with less number of current variables after node and constraint aggregations at coarse levels. This is equivalent to reducing the number of optimization variables in the LP problem, and it will eventually achieve a faster verification procedure with the cost of less accurate approximations. By doing so, we are able to flexibly explore the tradeoffs between the vectorless verification runtime and solution quality by choosing the sparsity of the input grids.

4.3 Experimental Results for Thermal Verification

4.3.1 Iterative Edge Scaling and Solution Refinement. To demonstrate the effectiveness of the proposed edge scaling and solution refinement schemes, four solution (temperature) vectors are calculated for a 3D thermal grid and its spectral sparsifiers: (a) the true solution vector obtained using the original thermal grid, (b) the approximate solution vector computed using the sparsifier without edge scaling, (c) the approximate solution vector obtained using the sparsifier with edge scaling, and (d) the refined (smoothed) solution vector using the sparsifier with edge scaling. We plot the relative errors (shown as l^2 -norm) with the number of iterations during the iterative edge



(a) LP runtime scalability of the proposed method.



(b) Total runtime scalability of the proposed method.

Fig. 11. Runtime scalability of the proposed method.

scaling procedure by comparing the solution vectors (c) and (d) against the true solution vector (a) in Figure 13(a). Meanwhile, we plot the histogram distributions of relative errors of the solution vectors (b)–(d) at the final iteration by comparing them against the true solution vector (a), as shown in Figure 13(b). We can see that the solution errors between the sparsified thermal grid and the original 3D thermal grid can be significantly reduced by leveraging the proposed iterative edge scaling scheme, and further mitigated by the proposed solution refinement procedure.

4.3.2 Result of Verification Quality. As we mentioned earlier, low-frequency components of the original thermal grid solutions can be well preserved after spectral graph sparsification, which is the key to achieving high-quality solutions for vectorless verification tasks. Figures 14 and 15 show the worst-case thermal distributions of processors A and B using (a) the direct method, (b) the multilevel vectorless verification method w/o sparsification, and (c) the multilevel vectorless verification method w/spectral sparsification, respectively. As we can see from the figures, the thermal distributions using three methods are very close to each other, indicating that rather accurate vectorless verification results can be obtained using multilevel spectrally sparsified thermal grids.

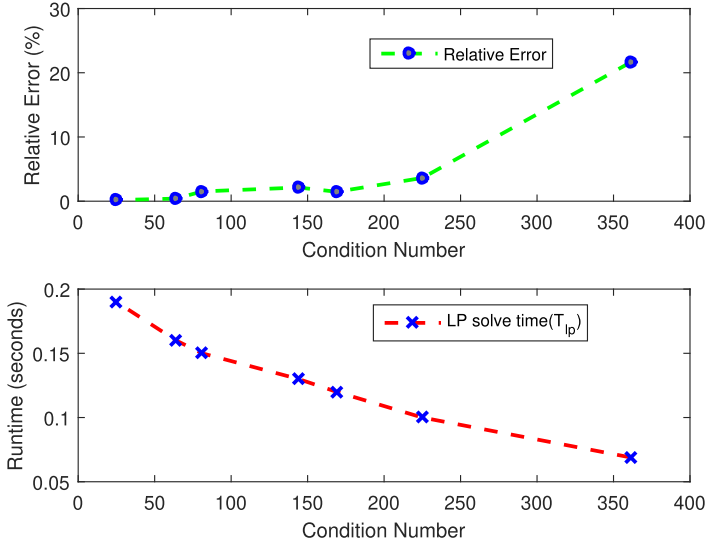


Fig. 12. Result of the tradeoff analysis using the proposed method.

Table 5. Results of the Proposed Multilevel Vectorless Thermal Integrity Verification Method (Two-level Scheme Is Used)

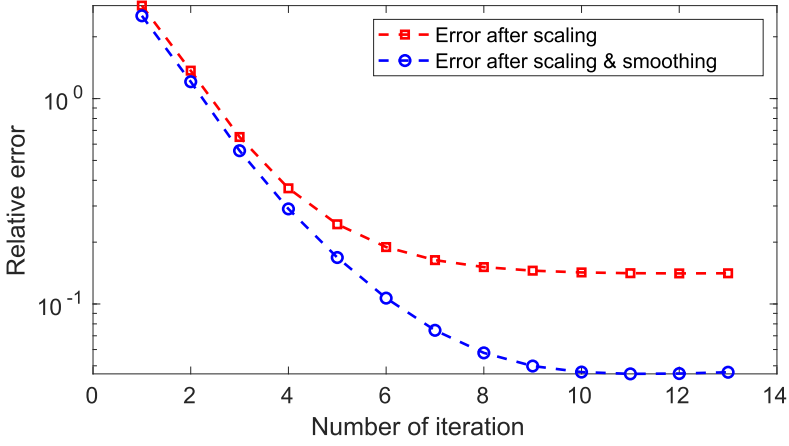
CKT	(a) Single Level			(b) Multilevel w/o Sparsification				(c) Multilevel w/ Sparsification				
	T_{chol}^o	T_{sol}^o	T_{lp}^o	T_{chol}^m	T_{sol}^m	T_{lp}^m	$Err(\%)$	T_{chol}^s	T_{sol}^s	T_{lp}^s	$Err(\%)$	κ
T1	0.94 s	2.30 s	2.71 s	1.24 s	3.21 s	3.12s	1.0%	0.03 s	0.13 s	2.02 s	5.0%	2,073
T2	5.89 s	14.79 s	10.36 s	8.12 s	20.48 s	5.80s	2.1%	0.29 s	0.72 s	6.81 s	3.8%	2,400
T3	24.26 s	55.20 s	20.08 s	33.91 s	86.07 s	25.90 s	4.0%	1.13 s	2.90 s	4.33 s	4.0%	1,435
T4	38.03 s	99.91 s	60.56 s	50.91 s	131.97 s	24.15 s	2.0%	4.61 s	10.46 s	14.48 s	5.0%	2,193
T5	110.17 s	262.53 s	159.83 s	148.11 s	335.97 s	21.43 s	1.0%	20.09 s	43.50 s	9.36 s	2.0%	2,469
T6	1.18K s	33.60K s	0.87K s	1.25K s	33.99K s	0.79K s	1.0%	51.70 s	167.00 s	133.83 s	1.0%	2,141
T7	1.32K s	32.27K s	1.76K s	1.42K s	28.91K s	1.70K s	1.0%	65.23 s	187.36 s	181.76 s	2.0%	3,073

Table 6. Runtime Results of the Proposed Method

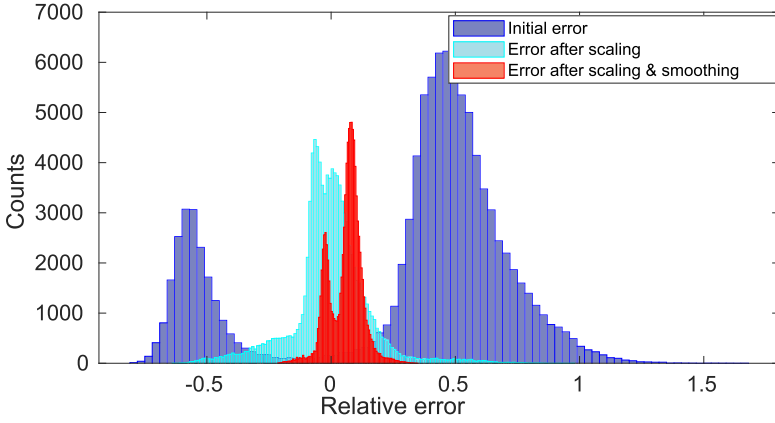
CKT	Sp. Over Single Level			Sp. Over Original Multilevel		
	$\frac{T_{chol}^o}{T_{chol}^s}$	$\frac{T_{sol}^o}{T_{sol}^s}$	$\frac{T_{lp}^o}{T_{lp}^s}$	$\frac{T_{chol}^m}{T_{chol}^s}$	$\frac{T_{sol}^m}{T_{sol}^s}$	$\frac{T_{lp}^m}{T_{lp}^s}$
T1	31×	18×	1.4×	41×	25×	1.6×
T2	20×	20×	1.6×	28×	28×	0.85×
T3	22×	19×	4.6×	30×	30×	6.0×
T4	9×	10×	4×	11×	13×	1.7×
T5	6×	6×	18×	8×	8×	2.3×
T6	23×	201×	6.5×	24×	204×	5.9×
T7	20×	172×	9.7×	22×	154×	9.1×

4.3.3 Comprehensive Results. Vectorless thermal integrity verification results using different methods are shown in Table 5. Except for T_{chol}^* , all other time are computed by summing up the runtimes for verifying 100 randomly chosen nodes.

Table 6 shows the runtime speedups of Cholesky factorization, adjoint sensitivity calculation and LP solving when comparing the proposed multilevel method with the other two methods. It is observed that “Multilevel w/ Sparsification” method is consistently much faster than the other



(a) Relative errors vs number of iterations.



(b) Relative error distributions.

Fig. 13. Relative errors with iterative edge scaling and solution refinement scheme.

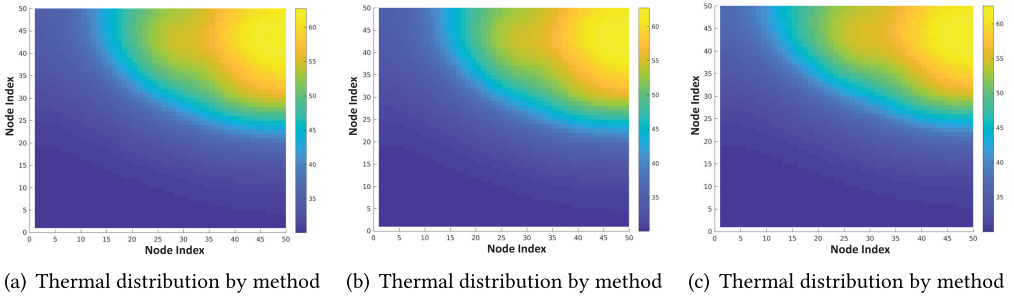


Fig. 14. Worst-case temperature distributions of processor A.

two methods, especially for larger test cases. And the total runtime speedups can be up to $100\times$ for the larger thermal grids when using the proposed method, as shown in Figure 16. The “Multilevel w/o Sparsification” is the slowest method due to the fast growing matrix densities at coarse levels. The overall LP solution time T_{lp} for the proposed method is also the smallest, indicating that our

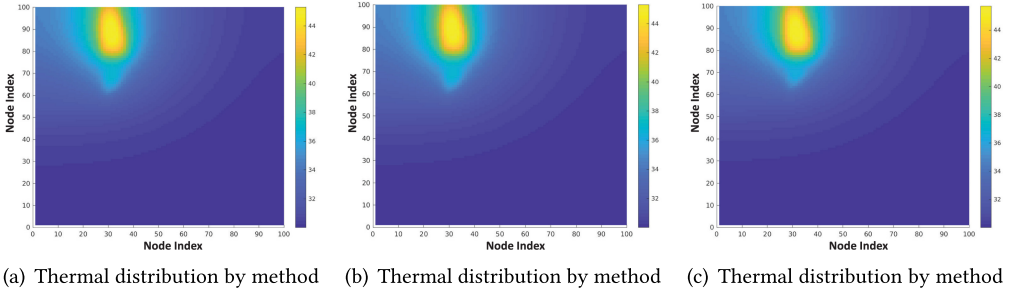


Fig. 15. Worst-case temperature distributions of processor B.

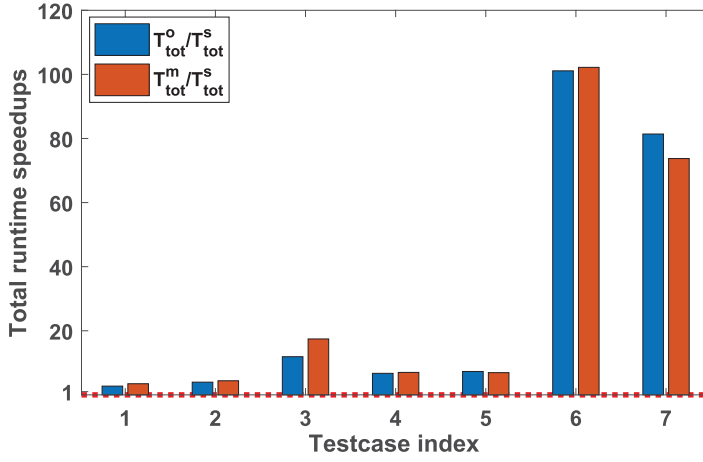
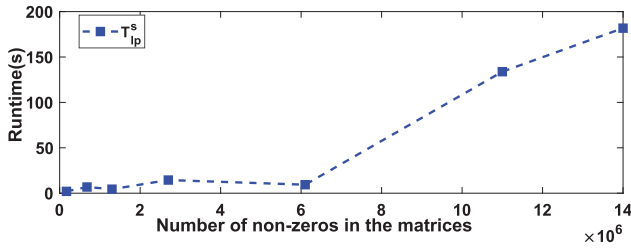
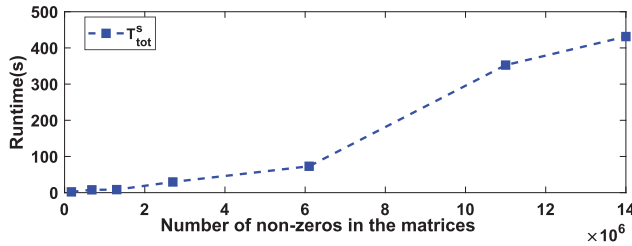


Fig. 16. Total runtime speedups of Multilevel w/Sparsification method comparing to the other two methods.



(a) LP solve time with the number of non-zero in matrices



(b) Total verification time with the number of non-zeros in matrices

Fig. 17. Verification time with various problem sizes.

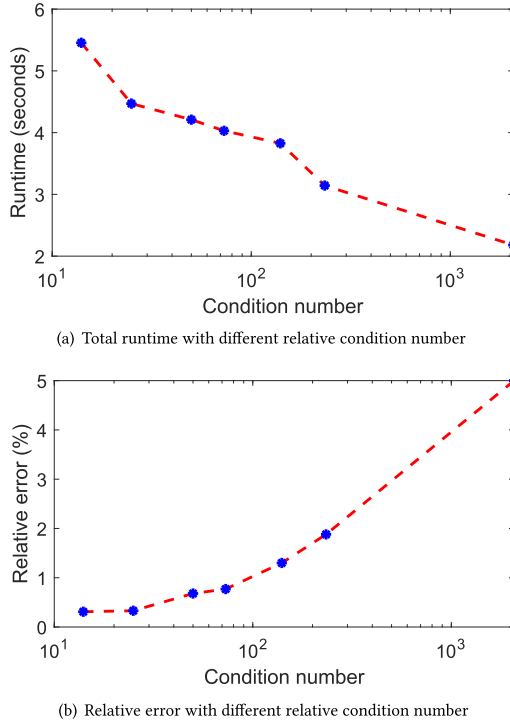


Fig. 18. Tradeoff analysis using the proposed method.

method can effectively reduce the number of decision variables in LP and thus result in much lower computational cost in vectorless thermal verification tasks.

Meanwhile, the proposed method scales very comfortably with even very large 3D thermal grids, since both the LP solve time and total verification time increase almost linearly with the 3D thermal grid sizes, as shown in Figure 17.

Figure 18 shows how vectorless thermal verification solution quality (error) and the total runtime will change with different relative condition numbers (κ). The relative errors grow slowly with increasing condition numbers, while the runtime is decreased, indicating that the proposed method allows to flexibly explore the tradeoffs between the vectorless verification runtime and solution quality.

5 CONCLUSION

We present a highly scalable multilevel vectorless power grid and thermal integrity verification framework for computing chip worst-case voltage drop or thermal profiles without knowing exact distribution of underlying power sources or workloads. Recent theoretical results in spectral graph sparsification and coarsening, as well as graph signal processing techniques enable us to develop much faster and more scalable vectorless integrity verification algorithms, while achieving flexible tradeoffs between computing efficiency and solution quality. Extensive experiment results for various chip designs have been demonstrated, indicating that the proposed scalable vectorless verification method can always efficiently obtain highly accurate results for large chip designs.

REFERENCES

- [1] J. Batson, D. Spielman, N. Srivastava, and S. Teng. 2013. Spectral sparsification of graphs: Theory and algorithms. *Commun. ACM* 56, 8 (2013), 87–94.
- [2] Gecia Bravo Hermisdorff and Lee Gunderson. 2019. A unifying framework for spectrum-preserving graph sparsification and coarsening. In *Advances in Neural Information Processing Systems*, Volume 32, 7736–7747.
- [3] William L. Briggs, Van Emden Henson, and Steve F. McCormick. 2000. *A Multigrid Tutorial*. SIAM.
- [4] Chen Cai, Dingkan Wang, and Yusu Wang. 2021. Graph Coarsening with Neural Networks. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=uxpzitPEooJ>.
- [5] Jie Chen and Ilya Safro. 2011. Algebraic distance on graphs. *SIAM J. Sci. Comput.* 33, 6 (2011), 3468–3490.
- [6] Yanqing Chen, Timothy A. Davis, William W. Hager, and Sivasankaran Rajamanickam. 2008. Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate. *ACM Trans. Math. Softw.* 35, 3 (2008), 1–14.
- [7] Ryan Cochran and Sherief Reda. 2010. Consistent runtime thermal prediction and control through workload phase detection. In *Proceedings of the 47th Design Automation Conference*. ACM, 62–67.
- [8] D. Kouroussis, I. Ferzli, and F. Najm. 2005. Incremental partitioning-based vectorless power grid verification. In *Proceedings of the IEEE International Conference on Computer-Aided Design (ICCAD'05)*. 358–364.
- [9] T. Davis. 2008. CHOLMOD: Sparse Supernodal Cholesky Factorization and Update/downdate. Retrieved from <http://www.cise.ufl.edu/research/sparse/cholmod/>.
- [10] Florian Dorfler and Francesco Bullo. 2012. Kron reduction of graphs with applications to electrical networks. *IEEE Trans. Circ. Syst. I: Regul. Pap.* 60, 1 (2012), 150–163.
- [11] Z. Feng. 2013. Scalable multilevel vectorless power grid voltage integrity verification. *IEEE Trans. VLSI Syst.* 21, 8 (August 2013), 1526–1539.
- [12] Z. Feng. 2013. Scalable vectorless power grid current integrity verification. In *Proceedings of the Design Automation Conference (DAC'13)*. 86:1–86:8.
- [13] Z. Feng. 2016. Spectral graph sparsification in nearly-linear time leveraging efficient spectral perturbation analysis. In *Proceedings of the Design Automation Conference (DAC'16)*.
- [14] Zhuo Feng. 2018. Similarity-aware spectral sparsification by edge filtering. In *Proceedings of the 55nd Design Automation Conference (DAC'18)*.
- [15] N. Ghani and F. Najm. 2009. Fast vectorless power grid verification using an approximate inverse technique. In *Proceedings of the Design Automation Conference (DAC'09)*. 184–189.
- [16] A. Goyal and F. Najm. 2011. Efficient RC power grid verification using node elimination. In *Proceedings of the Design, Automation and Test in Europe Conference (DATE'11)*. 257–260.
- [17] Wei Huang, Shougata Ghosh, Sivakumar Velusamy, Karthik Sankaranarayanan, Kevin Skadron, and Mircea R. Stan. 2006. HotSpot: A compact thermal modeling methodology for early-stage VLSI design. *IEEE Trans. VLSI Syst.* 14, 5 (2006), 501–513.
- [18] George Karypis and Vipin Kumar. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.* 20, 1 (1998), 359–392.
- [19] D. Kouroussis and F. Najm. 2003. A static pattern-independent technique for power grid voltage integrity verification. In *Proceedings of the Design Automation Conference (DAC'03)*. 99–104.
- [20] Dominique LaSalle, Md Mostofa Ali Patwary, Nadathur Satish, Narayanan Sundaram, Pradeep Dubey, and George Karypis. 2015. Improving graph partitioning for modern graphs and architectures. In *Proceedings of the 5th Workshop on Irregular Applications: Architectures and Algorithms*. ACM, 14.
- [21] R. Lewis and S. Nash. 2005. Model problems for the multigrid optimization of systems governed by differential equations. *SIAM J. Sci. Comput.* 26, 6 (2005), 1811–1837. <https://doi.org/10.1137/S1064827502407792>
- [22] Peng Li, Lawrence T. Pileggi, Mehdi Asheghi, and Rajit Chandra. 2006. IC thermal simulation and modeling via efficient multigrid-based approaches. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 25, 9 (2006), 1763–1776.
- [23] Mark Po-Hung Lin, Hongbo Zhang, Martin D. F. Wong, and Yao-Wen Chang. 2011. Thermal-driven analog placement considering device matching. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 30, 3 (2011), 325–336.
- [24] O. Livne and A. Brandt. 2012. Lean algebraic multigrid (LAMG): Fast graph Laplacian linear solver. *SIAM J. Sci. Comput.* 34, 4 (2012), B499–B522.
- [25] Andreas Loukas. 2019. Graph reduction with spectral and cut guarantees. *J. Mach. Learn. Res.* 20, 116 (2019), 1–42.
- [26] Andreas Loukas and Pierre Vandergheynst. 2018. Spectrally approximating large graphs with smaller graphs. In *International Conference on Machine Learning*. PMLR, 3237–3246.
- [27] F. Najm. 2012. Overview of vectorless/early power grid verification. In *Proceedings of the IEEE International Conference on Computer-Aided Design (ICCAD'12)*. 670–677.
- [28] S. R. Nassif. 2008. IBM Power Grid Benchmarks. Retrieved from <http://dropzone.tamu.edu/~pli/PGBench/>.
- [29] Gurobi Optimization. 2016. Gurobi Optimizer. Retrieved from www.gurobi.com.

- [30] M. Necati Ozisik. 2002. *Boundary Value Problems of Heat Conduction*. Courier Corporation.
- [31] Massoud Pedram and Shahin Nazarian. 2006. Thermal modeling, analysis, and management in VLSI circuits: Principles and methods. *Proc. IEEE* 94, 8 (2006), 1487–1501.
- [32] Manish Purohit, B. Aditya Prakash, Chanhyun Kang, Yao Zhang, and V. S. Subrahmanian. 2014. Fast influence-based coarsening for large networks. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1296–1305.
- [33] H. Qian, S. R. Nassif, and S. S. Sapatnekar. 2005. Early-stage power grid analysis for uncertain working modes. *IEEE Trans. Comput.-Aid. Des.* 24, 5 (2005), 676–682.
- [34] Dorit Ron, Ilya Safro, and Achi Brandt. 2011. Relaxation-based coarsening and multiscale graph organization. *Multisc. Model. Simul.* 9, 1 (2011), 407–423.
- [35] David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Sign. Process. Mag.* 30, 3 (2013), 83–98.
- [36] Kevin Skadron, Mircea R. Stan, Karthik Sankaranarayanan, Wei Huang, Sivakumar Velusamy, and David Tarjan. 2004. Temperature-aware microarchitecture: Modeling and implementation. *ACM Trans. Arch. Code Optim.* 1, 1 (2004), 94–125.
- [37] D. Spielman. 2010. Algorithms, graph theory, and linear equations in laplacian matrices. In *Proceedings of the International Congress of Mathematicians*.
- [38] D. Spielman and N. Srivastava. 2008. Graph sparsification by effective resistances. In *Proceedings of the ACM Symposium on Theory of Computing (STOC'08)*. 563–568.
- [39] X. Xiong and J. Wang. 2010. An efficient dual algorithm for vectorless power grid verification under linear current constraints. In *Proceedings of the Design Automation Conference (DAC'10)*. 837–842.
- [40] Jianlei Yang, Yici Cai, Qiang Zhou, and Wei Zhao. 2015. A selected inversion approach for locality driven vectorless power grid verification. *IEEE Trans. VLSI Syst.* 23, 11 (2015), 2617–2628.
- [41] J. Yang and Z. Li. [n. d.]. THU Power Grid Benchmarks. Retrieved from <http://tiger.cs.tsinghua.edu.cn/PGBench/>.
- [42] Zhiqiang Zhao and Zhuo Feng. 2017. A spectral graph sparsification approach to scalable vectorless power grid integrity verification. In *Proceedings of the 54th Annual Design Automation Conference*. ACM, 68.
- [43] Zhiqiang Zhao and Zhuo Feng. 2019. Effective-resistance preserving spectral reduction of graphs. In *Proceedings of the 56th Annual Design Automation Conference 2019*. 1–6.
- [44] Zhiqiang Zhao and Zhuo Feng. 2020. A spectral approach to scalable vectorless thermal integrity verification. In *Proceedings of the Design, Automation & Test in Europe*. IEEE.
- [45] Zhiqiang Zhao, Yongyu Wang, and Zhuo Feng. 2017. SAMG: Sparsified graph-theoretic algebraic multigrid for solving large symmetric diagonally dominant (SDD) matrices. In *Proceedings of the 36th International Conference on Computer-Aided Design (ICCAD'17)*. ACM.
- [46] Zhiqiang Zhao, Ying Zhang, and Zhuo Feng. 2021. Towards scalable spectral embedding and data visualization via spectral coarsening. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 869–877.

Received 3 December 2021; revised 3 March 2022; accepted 29 March 2022