ePlace-MS: Electrostatics-Based Placement for Mixed-Size Circuits

Jingwei Lu, *Member, IEEE*, Hao Zhuang, *Student Member, IEEE*, Pengwen Chen, Hongliang Chang,
Chin-Chih Chang, Yiu-Chung Wong, Lu Sha, Dennis Huang, Yufeng Luo, Chin-Chi Teng,
and Chung-Kuan Cheng, *Fellow, IEEE*

***Abstract*— We propose an electrostatics-based placement algorithm for large-scale mixed-size circuits (ePlace-MS). ePlace-MS is generalized, flat, analytic and nonlinear. The density modeling method eDensity is extended to handle the mixed-size placement. We conduct detailed analysis on the correctness of the gradient formulation and the numerical solution, as well as the rationale of dc removal and the advantages over prior density functions. Nesterov's method is used as the nonlinear solver, which shows high yet stable performance over mixed-size circuits. The steplength is set as the inverse of Lipschitz constant of the gradient function, while we develop a backtracking method to prevent overestimation. An approximated nonlinear preconditioner is developed to minimize the topological and physical differences between large macros and standard cells. Besides, we devise a simulated annealer to legalize the layout of macros and use a second-phase global placement to reoptimize the standard cell layout. All the above innovations are integrated into our mixed-size placement prototype ePlace-MS, which outperforms all the related works in literature with better quality and efficiency. Compared to the leading-edge mixed-size placer NTUplace3, ePlace-MS produces up to 22.98% and on average 8.22% shorter wirelength over all the 16 modern mixed-size benchmark circuits with the same runtime.**

***Index Terms*—Analytic placement, electrostatic analogy, fast Fourier transform (FFT), Lipschitz constant, Nesterov's method, nonlinear optimization, Poisson's equation, preconditioning, spectral methods.**

I. Introduction

Placement remains crucial and challenging in VLSI physical design [22], [31]. Placement performance impacts downstream design phases of clock tree synthesis [27], dynamic power minimization [28], global and detailed routing [30], etc. The advent of billion-transistor integration [14] makes the placement performance dominant on the overall quality. Placement quality is usually evaluated by the total half-perimeter wirelength (HPWL), which correlates with timing [29], [48], routability [10], [41], and power [26], subject to the constraint of zero overlap among circuit components. Such problem formulation is broadly used among research developments [3], [5], [15], [17], [18], [45], [46] and well honored by public placement benchmarks [33], [34], [47]. Over thousands of predesigned IP blocks, macros and memory units are embedded to shorten the total design turnaround, where the topological and physical differences with standard cells are huge. The high design complexity and complication continuously challenge the capability of mixed-size placers.

Prior mixed-size placement algorithms form three categories. Two-stage methods use two separated phases of floorplanning and placement. Location and orientation of macros are determined and fixed at first, while placement follows to optimize only standard cells in the global scale. MP-tree [6] packs macros along the chip boundaries. A constraint-graph (CG) algorithm [4] uses mathematical programming to optimize displacement, macro positions, and orientations. However, the limited information of standard cell distribution misguides the floorplanner at early stages, inducing suboptimal floorplan solution to degrade the overall quality.

Constructive (floorplan-guided) approaches combine the advantages of floorplan and placement. The floorplanner simultaneously optimizes both macros and soft blocks (clusters of standard cells). An incremental placement then spread standard cells in local scale. Capo [39] repeatedly invokes a fixed-outline floorplanner over the top-down placement framework, providing guidance to macro shifting. FLOP [47] groups cells into soft blocks to produce initial floorplan solution. Incremental global [46] and detailed placement further spread and legalize the standard cells within local scale. Nonetheless, the intrinsic limitation of partitioning and clustering usually induce suboptimal solutions in the placement perspective. Optimization space of standard cell placement could be substantially shrunk with quality loss hard to recover.

One-stage solution remains popular among most modern placement algorithms [13], [15], [17], [18], [21], [46]. Macros and standard cells are being placed simultaneously where the limitations discussed above can be well avoided. FastPlace3.0 [46] performs selective grid resizing to accommodate large macros with more whitespace. ComPLx [17] shreds macros into small objects with sizes similar to that of the standard cells. After placement finishes, each macro is reconstructed based on the gravity center of instances belonging to it. APlace3 [15] reshapes the smoothing curve of the density function to distinguish the smoothness of macro movement with that of standard cells. NTUplace3 [13]

Manuscript received July 19, 2014; revised September 15, 2014; accepted November 13, 2014. Date of publication January 13, 2015; date of current version April 17, 2015. This work was supported by the National Science Foundation (NSF) under Grant CCF-1017864. This paper was recommended by Associate Editor C. Sze.

J. Lu, H. Zhuang, and C.-K. Cheng are with the Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92093-0404 USA (e-mail: jlu@cs.ucsd.edu).

P. Chen is with the Department of Applied Mathematics, National Chung Hsing University, Taichung 402, Taiwan.

H. Chang, C.-C. Chang, Y.-C. Wong, L. Sha, D. Huang, Y. Luo, and C.-C. Teng are with Cadence Design Systems, Inc., San Jose, CA 95134 USA.

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

Digital Object Identifier 10.1109/TCAD.2015.2391263

0278-0070 © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on June 12,2025 at 02:50:26 UTC from IEEE Xplore. Restrictions apply.

incorporates rotational and flipping components into the gradient function, which enables simultaneous optimization on the location of all the movable objects as well as the orientation of macros. As mentioned in [47], macro and standard cell co-placement challenges the capability of modern analytic placement approaches. Despite largest search space, nevertheless, the substantial topological and physical differences between macros and standard cells might introduce gradient imbalance and cause the solution hard to converge.

In this paper, we develop a generalized one-stage, flat, and analytic nonlinear algorithm for mixed-size placement, based on the infrastructure of our early works Fast Fourier Transform based PLacement (FFTPL) [24] and ePlace [23] for standard-cell-based placement, as well as its extension to the mixed-size circuits [25]. As the major difficulty of mixed-size placement remains in the broad spectrum of topological and physical attributes among all the movable objects (i.e., standard cells and large macros), our innovation of nonlinear preconditioning well equalizes them in the solver's perspective. As a generalized algorithm, ePlace-MS handles standard cells and macros in exactly the same way (see macro shifting when declustering [13], soft block formation by standard cells [46], [47], special macro density smoothing [15], [18], macro shredding [17], etc.) to ensure high and stable performance over various integrated circuits with potentially quite different structures of the design. Our contributions are listed as follows.

1) We extend our prior density function eDensity [24], [25] to model mixed-size integrated circuits in a generalized way. Besides, we provide detailed analysis and proof on eDensity with:
   a) rationale of its dc removal;
   b) correctness of the density gradient formulation;
   c) correctness of the numerical solution;
   d) advantages over literature density functions.
2) We extend Nesterov's method as the nonlinear solver to handle mixed-size placement, with steplength dynamically predicted via Lipschitz constant. Moreover, we develop a backtracking method to effectively prevent steplength overestimation.
3) We develop an approximated nonlinear preconditioner to resolve the substantial topological and physical gap between standard cells and macros. The solution quality is significantly improved with negligible runtime overhead.
4) We devise an annealing-based macro legalizer providing direct control to the macro shifting. A second-phase standard cell-only global placement (cGP) is proposed to resolve the quality overhead induced during macro legalization.
5) We integrate all the innovations into ePlace-MS, an electrostatics-based prototype for mixed-size placement, with promising experimental results obtained on the modern mixed-size (MMS) [47] circuits. Specifically, ePlace-MS outperforms the leading placer NTUplace3 [13] with 8.22% shorter wirelength and the same runtime on average of all the 16 MMS benchmarks [47].

The remainder is organized as follows. Section II introduces the background knowledge. Section III provides an overview of ePlace-MS. Section IV analyzes our placement density function eDensity on the mixed-size circuits. Section V discusses the nonlinear placement optimization. Section VI proposes a nonlinear preconditioner to equalize macros with standard cells. Section VII introduces our annealing-based macro legalization. Section VIII discusses the second-phase cGP. Experiments and results are shown in Section IX. We conclude in Section X.

II. Essential Concepts

Given a placement instance $G = (V, E, R)$ with a set V of n movable objects (standard cells and macros), nets E and placeable region R, the placement is formulated as a constrained optimization problem. The constraint desires a solution $\mathbf{v} = (\mathbf{x}, \mathbf{y})^T = (x_1, \ldots, x_n, y_1, \ldots, y_n)^T$ to accommodate all the objects with sufficient sites but zero overlap or density violation. Global placement uniformly decomposes the entire region R into $m \times m$ rectangular grids (bins) which are denoted as grid set B. For every grid $b \in B$, the density $\rho_b(\mathbf{v})$ should not exceed a predetermined upper-bound ρ_t, which is named target density and usually design specific. The objective is set as minimizing the total HPWL of all the nets. Let $\text{HPWL}_e(\mathbf{v})$ denote the HPWL of the net e, the total HPWL is expressed as

$$\begin{aligned}\text{HPWL}(\mathbf{v}) &= \sum_{e\in E} \text{HPWL}_e(\mathbf{v}) \\ &= \sum_{e\in E}\left(\max_{i,j\in e}|x_i - x_j| + \max_{i,j\in e}|y_i - y_j|\right).\end{aligned} \tag{1}$$

Here i and j are any two of all the objects connected by the net e. As a result, the nonlinear optimization of global placement is formulated as (2) and proved to be NP-complete [9]

$$\min_{\mathbf{v}} \ \text{HPWL}(\mathbf{v}) \text{ s.t. } \rho_b(\mathbf{v}) \le \rho_t, \ \forall b \in B. \tag{2}$$

Analytic methods conduct placement using gradient-based optimization. As the function $\text{HPWL}(\mathbf{v})$ in (1) is not differentiable, various quadratic [7], [20] and nonlinear [12], [35] wirelength smoothing techniques have been proposed in literature to ensure the differentiability thus analytic placement to work. In this paper, we use both the log-sum-exp (LSE) [35] model and the weighted-average (WA) [12] model (as two options for ePlace-MS) to approximate HPWL via a closed-form smooth function $W(\mathbf{v})$, as shown in

$$W_{e_x}(\mathbf{v}) = \gamma\left(\log\sum_{i\in e}\exp(x_i/\gamma) + \log\sum_{i\in e}\exp(-x_i/\gamma)\right) \tag{3}$$

$$W_{e_x}(\mathbf{v}) = \frac{\sum_{i\in e} x_i \exp(x_i/\gamma)}{\sum_{i\in e}\exp(x_i/\gamma)} - \frac{\sum_{i\in e} x_i \exp(-x_i/\gamma)}{\sum_{i\in e}\exp(-x_i/\gamma)}. \tag{4}$$

Here, $W_e(\mathbf{v}) = W_{e_x}(\mathbf{v}) + W_{e_y}(\mathbf{v})$ and $W(\mathbf{v}) = \sum_e W_e(\mathbf{v})$. γ is used to control the modeling accuracy, i.e., smaller γ will improve the accuracy of approximation but decrease the function smoothness, vice versa. In this paper, we use ePlace-MS-WA and ePlace-MS-LSE to denote the usage of each wirelength model in our algorithm, respectively.

A density penalty function helps incorporate the $|B|$ constraints in (2) in order to achieve analyticity. Modern quadratic placers [17], [18], [21], [45], [46] are all based on the density force formulation proposed in [7], where anchor points are inserted to drag cells away from over-filled region, modeling the density force as a constant term in the gradient function. In contrast, nonlinear placers formulate the density gradient as a component independent to wirelength. APlace3 [15] and

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on June 12,2025 at 02:50:26 UTC from IEEE Xplore. Restrictions apply.

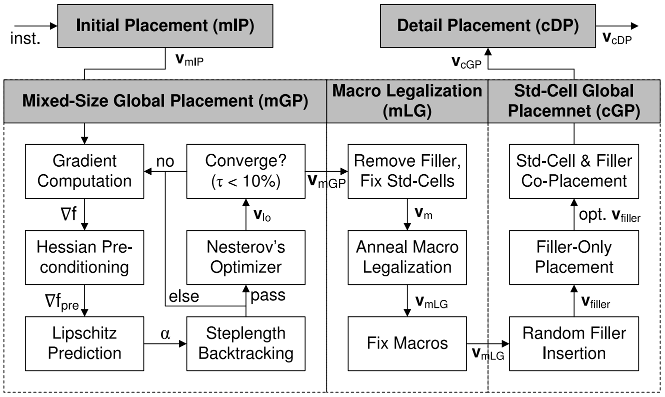
inst.
Initial Placement (mIP)
$\mathbf{v}_{\text{mIP}}$
Detail Placement (cDP)
$\mathbf{v}_{\text{cDP}}$
$\mathbf{v}_{\text{cGP}}$
Mixed-Size Global Placement (mGP)
Macro Legalization (mLG)
Std-Cell Global Placemnet (cGP)
Gradient Computation
no
Converge? (τ < 10%)
$\mathbf{v}_{\text{mGP}}$
Remove Filler, Fix Std-Cells
Std-Cell & Filler Co-Placement
∇f
$\mathbf{v}_{\text{lo}}$
$\mathbf{v}_{\text{m}}$
opt. $\mathbf{v}_{\text{filler}}$
Hessian Pre-conditioning
Nesterov's Optimizer
Anneal Macro Legalization
Filler-Only Placement
∇f_{pre}
else
pass
$\mathbf{v}_{\text{mLG}}$
$\mathbf{v}_{\text{filler}}$
Lipschitz Prediction
α
Steplength Backtracking
Fix Macros
$\mathbf{v}_{\text{mLG}}$
Random Filler Insertion

Fig. 1. Flowchart of ePlace-MS.

NTUplace3 [13] ensure differentiability of the density distribution via bell-shape curving [35] with local smoothness, while mPL6 [3] uses Helmholtz function targeting global density smoothness. In this paper, we use the electrostatics-based density modeling method eDensity developed in our prior work FFTPL [24], which will be thoroughly analyzed in Section IV with proof of correctness. The numerous grid density constraints are all relaxed by using this single penalty function, while we have the unconstrained placement optimization problem defined as

$$\min_{\mathbf{v}} f(\mathbf{v}) = W(\mathbf{v}) + \lambda N(\mathbf{v}) \tag{5}$$

where $N(\mathbf{v})$ [defined in (6)] is the density function and λ is the penalty factor for adjusting the ratio between wirelength and density.

III. Placement Overview

Fig. 1 shows the flowchart of ePlace-MS. Given a placement instance, ePlace-MS quadratically minimizes the total wirelength at the first stage of mixed-size initial placement (mIP). The initial solution $\mathbf{v}_{\text{mIP}}$ is of low wirelength but high overlap. Based on the target density ρ_t, our mixed-size global placer (mGP) populates extra whitespace with unconnected fillers, then iteratively co-optimizes all the objects (standard cells, macros, and fillers) together. After mGP, we remove all the fillers, fix the standard-cell layout, then invoke the annealing engine macro legalization (mLG) to legalize the location of all the macros. In the second-phase global placement (cGP), we retrieve all the fillers and distribute them appropriately, then free standard cells and co-place them with fillers to further reduce the wirelength. Finally, in the standard-cell detailed placement (cDP), we invoke the detailed placer in [13] to legalize and discretely optimize the standard-cell layout.

ePlace-MS does not allow rotation or flipping of any objects due to the lithography issue. However, it has the flexibility to smoothly integrate the rotational and flipping gradients [13] to guide placement optimization iteratively. Deadspace allocation is also not considered in this paper, while it can be effectively realized in ePlace-MS via appropriate macro inflation.

ePlace-MS maximally expands the design space for mGP with the major optimization effort budgeted on mGP, since all the objects (standard cells, macros, and fillers) are allowed to move and can be optimized simultaneously. In contrast, the design spaces for mLG and cGP are relatively shrunk, as only macros or standard cells are allowed to move with other objects fixed thus acting as constraints, which actually constrains the search space of mixed-size placement solution.

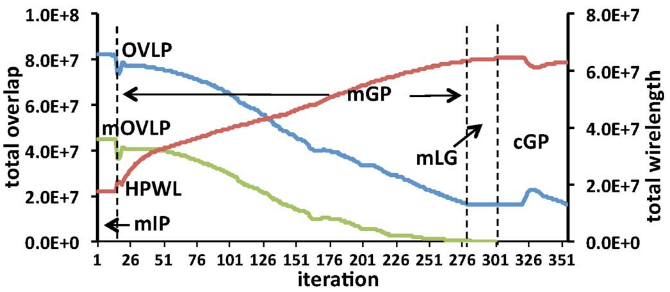

Fig. 2. Total HPWL, total object overlap (OVLP), and total macro overlap (mOVLP) at different stages and iterations of ePlace-MS-WA on the MMS ADAPTEC1 benchmark. Overlap between macros are cleaned at mLG where mOVLP decreases to zero. Remaining OVLP will all be cleaned at cDP (following cGP).

Specifically, only minor layout perturbation is expected to perform changes within local scale. As Fig. 2[1] shows, the constrained optimization focuses on the mGP stage and terminates when overlap is small enough. The entire placement framework is built upon our recent work of FFTPL [24] and ePlace [23] with similar initialization and iterative adjustment of parameters. Grid dimension m is statically determined as $m = \lceil \log_2 \sqrt{n'} \rceil$ and upper-bounded by 1024, where $n' = |V'|$ is the number of movable macros, standard cells and fillers [23], [24]. Penalty factor λ is initially set as (10) of [5]. We iteratively update $\lambda_k = \mu_k \lambda_{k-1}$ in mGP to balance the wirelength and density forces, where $\mu_k = 1.1^{-(\Delta\text{HPWL}_k/\Delta\text{HPWL}_{\text{REF}})+1.0}$ based on the HPWL variation $\Delta\text{HPWL}_k = \text{HPWL}(\vec{v}_k) - \text{HPWL}(\vec{v}_{k-1})$. In practice, we set $\Delta\text{HPWL}_{\text{REF}} = 3.5 \times 10^5$ and bound μ_k by $[0.75, 1.1]$. Density overflow τ is used as the stopping criterion. We terminate mGP when $\tau \leq 10\%$ and cGP when $\tau \leq 7\%$, respectively. Wirelength coefficient γ is used to smooth the HPWL. We set the smoothing parameter as $\gamma = 8.0 w_b \times 10^{20/9 \times (\tau - 0.1) - 1.0}$ to encourage global movement at early iterations and convergence at later iterations. Fillers are used to balance the electrostatic dc component in the global scale. The total area of fillers equals the total whitespace multiplies target density then subtracted by the total area of all the movable objects. All the fillers are equally sized to be the average physical dimensions of all the standard cells. More details of parameter adjustment or filler formation can be found in [23] and [24].

IV. Density Function Analysis

In this section, we prove the correctness of the density gradient formulation and numerical solution, and analyze the rationale of dc removal and the advantages over previous placement density methods. Given its high performance on standard cell circuits, we extend the density function to handle the mixed-size placement in a generalized way. Fig. 3 shows the progression via a density-only mixed-size placement by ePlace-MS, where standard cells and macros are smoothly co-optimized toward even density distribution. Tables II and III show that our density function has the best performance with shortest wirelength and smallest density overflow versus all the mixed-size placers in [13], [17], [46], and [47].

A. Density Function

A placement density function is developed in [24] based on the electrostatic analogy, which is therefore

[1] Here OVLP denotes physical overlap among all the objects. Computation costs $O(n \log n)$ time via scanline and segment-tree data structure.

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on June 12,2025 at 02:50:26 UTC from IEEE Xplore. Restrictions apply.

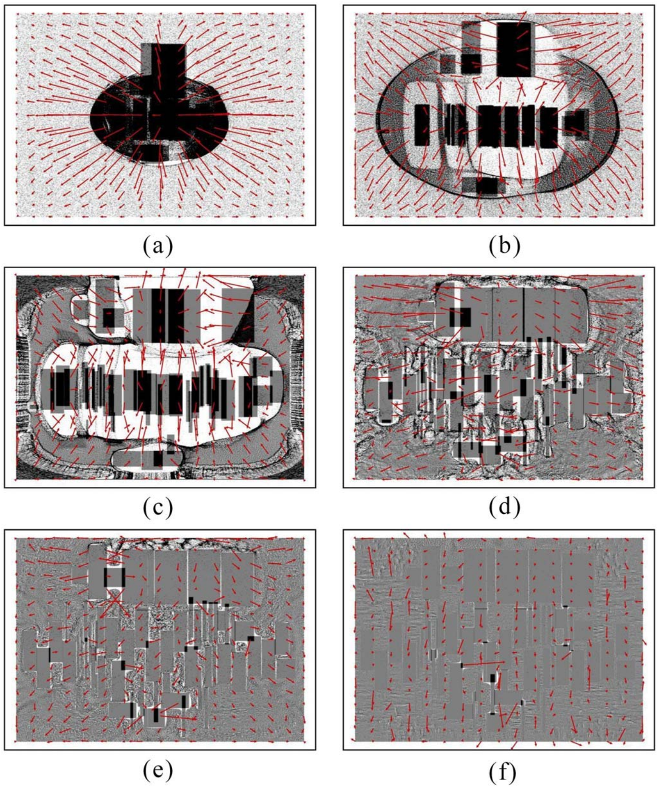

Fig. 3. Snapshots of the density distribution by eDensity via mixed-size placement on the MMS ADAPTEC1 benchmark. The placement is driven by only density forces (denoted by red arrows) with the magnitude of the grid density characterized by grayscale. Total potential energy and total density overflow are denoted by N and τ, respectively. (a) Iter = 4, $N = 11.84\text{e}11$, and $\tau = 74.67\%$. (b) Iter = 5, $N = 18.63\text{e}10$, and $\tau = 48.03\%$. (c) Iter = 7, $N = 30.33\text{e}9$, and $\tau = 26.61\%$. (d) Iter = 12, $N = 97.97\text{e}7$, and $\tau = 12.78\%$. (e) Iter = 20, $N = 10.70\text{e}7$, and $\tau = 6.45\%$. (f) Iter = 200, $N = 37.85\text{e}5$, and $\tau = 3.04\%$.

named eDensity. Modeling every object as a positive charge, the density function $N(\mathbf{v})$ shown in (6) is modeled as the total electric potential energy. The electric force keeps spreading all the charges apart from each other, thus reducing the total potential energy toward zero in the end. The electrostatic equilibrium state is coupled with even placement density distribution and will be eventually reached. Compared to all the previous mixed-size placement algorithms [3], [13], [15], [17], [21], [39], [46], [47], our density function achieves the minimum density overflow as shown in Table III, indicating the fewest violations to the target density thus the best performance of our density function

$$N(\mathbf{v}) = \frac{1}{2}\sum_{i\in V} N_i(\mathbf{v}) = \frac{1}{2}\sum_{i\in V} q_i\psi_i(\mathbf{v}). \tag{6}$$

Here q_i is the electric quantity of the charge i, it equals the area of the respective object i. ψ_i is the local potential. Also, as the system energy equals the sum of mutual potential energy between all the pairs of charges, we have a factor of $1/2$ for the energy of each single charge. A well-defined Poisson's equation in (7) correlates the density distribution $\rho(x, y)$ with the potential distribution $\psi(x, y)$, where x and y are spatial coordinates. We enforce Neumann boundary condition (i.e., zero gradient at the boundary of the density function or placement domain) to prevent objects from moving outside the placement region R. Specifically, the horizontal density gradients along the two vertical boundaries are equivalent to zero, vice versa, such that movement toward the placement boundaries will be gradually slowed down and finally stopped

$$\begin{cases} \nabla\cdot\nabla\psi(x, y) = -\rho(x, y) \\ \hat{\mathbf{n}}\cdot\nabla\psi(x, y) = \mathbf{0},\ (x, y)\in\partial R \\ \iint_R \rho(x, y) = \iint_R \psi(x, y) = 0. \end{cases} \tag{7}$$

Here $\hat{\mathbf{n}}$ is the outer normal vector at the boundary ∂R. We use $\boldsymbol{\xi}(x, y) = \nabla\psi(x, y)$ to denote the electric field distribution. The electric force on each charge i equals $q_i\boldsymbol{\xi}_i(\mathbf{v})$, where $\boldsymbol{\xi}_i = (\xi_{i_x}, \xi_{i_y})$ is the local field vector and can be decomposed into its horizontal (ξ_{i_x}) and vertical (ξ_{i_y}) components.

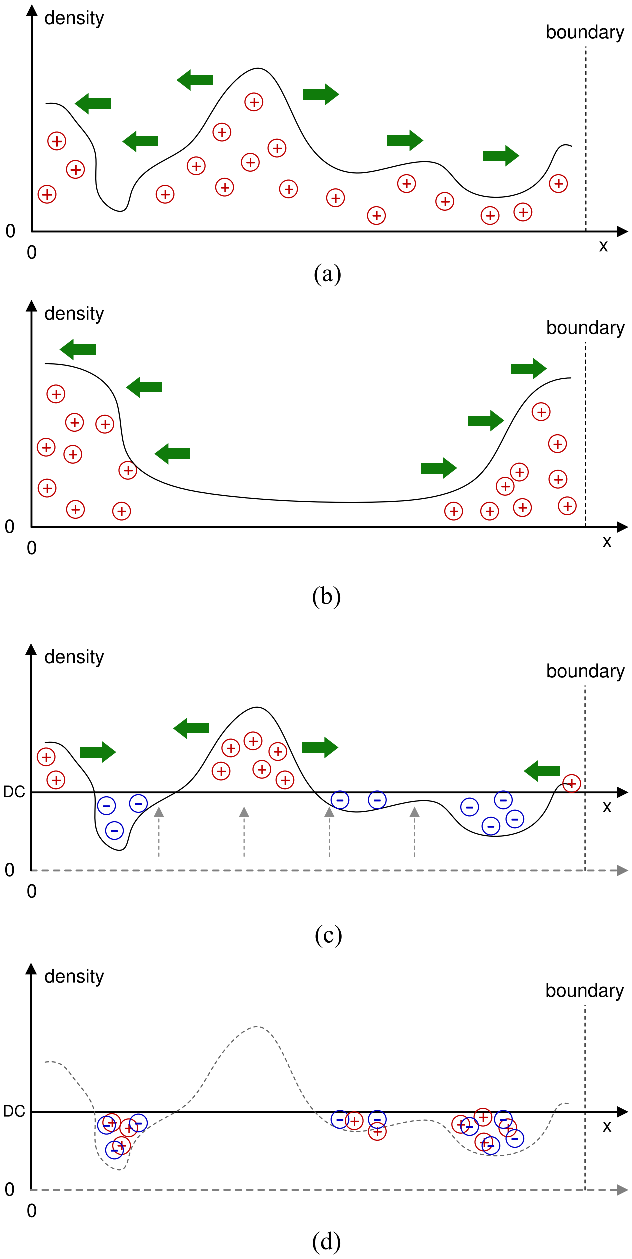

Fig. 4. 1-D example showing the correlation of electrostatic equilibrium with even density distribution after removing the dc component (ρ_{avg}) from the spatial density distribution ($\rho(x, y)$). (a) Initial charge density distribution. (b) Electrostatic equilibrium state with all the charges being pushed to the boundaries (moving beyond the placement domain is physically prohibited here). (c) Initial charge density distribution with the dc component (ρ_{avg}) removed. (d) Electrostatic equilibrium state with globally even density distribution.

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on June 12,2025 at 02:50:26 UTC from IEEE Xplore. Restrictions apply.

Our density function $N(\mathbf{v})$ is generalized. In contrast to prior nonlinear placers [13], [15], there is no special handling or smoothing applied to movable macros or fixed blocks. Please refer to Section IV-E for a more detailed advantage analysis. The global smoothness of $N(\mathbf{v})$ [by (6) and (7)] indicates that the local movement of any object will change the potential map in the global scale. The potential energy of all the objects will thus be changed by the movement of any single object i.

B. DC Removal

eDensity correlates even distribution of placement objects with the electrostatic equilibrium state. However, since all the objects are mapped to positively charged particles (with electric quantity set as the object area) as Fig. 4(a) shows, the electric forces applied on all the charges are purely repulsive. Such repulsive force will keep pushing all the charges toward infinity. On the other hand, if moving outside the placement domain is physically prohibited, the equilibrium state will have all the charges stay along the boundary lines in the end, as Fig. 4(b) illustrates, causing uneven placement density distribution. To resolve this problem, we remove the zero-frequency (i.e., dc) component from the spatial density distribution $\rho(x, y)$, in order to couple the electrostatic equilibrium state with an even charge density distribution. Besides, the constant terms in $\psi(x, y)$ and $\boldsymbol{\xi}(x, y)$ produced during the integral operations become zero and can be ignored. This also satisfies $\iint_R \rho(x, y) = \iint_R \psi(x, y) = 0$ in (7). We use ρ_{avg} to denote the dc of the global density distribution, i.e., the quantity to be reduced from the original density ρ_b of each grid, since dc of $\rho(x, y)$ equals the average grid density of B. As a result, we have $\sum_b \rho_b = 0$ after removing dc (ρ_{avg}) from ρ_b, $\forall b \in B$. Fig. 4(c) shows that removal of dc component reduces the sum of all the charges to zero and thus introduces negative charges to the low density regions, while attractive force is generated between objects besides the original repulsive force. The combination of all the repulsive and attractive force vectors thus guide the placement toward an even density distribution shown in Fig. 4(d), where there is no charge anywhere within the domain and the system potential energy is reduced to zero.

C. Correctness of Gradient Formulation

As discussed in Section IV-A, we use $q_i\boldsymbol{\xi}_i$ as the gradient of the density function $N(\mathbf{v})$ with respect to the horizontal movement of the charge i. However, by directly differentiating $N(\mathbf{v})$ with respect to x_i, we obtain the following formula:

$$\begin{aligned}\frac{\partial N(\mathbf{v})}{\partial x_i} &= \frac{1}{2}\left(\frac{\partial N_i(\mathbf{v})}{\partial x_i} + \frac{\partial\left(\sum_{j\neq i} N_j(\mathbf{v})\right)}{\partial x_i}\right) \\ &= \frac{1}{2} q_i \frac{\partial \psi_i}{\partial x_i} + \frac{1}{2}\sum_{j\neq i} q_j \frac{\partial \psi_j}{\partial x_i} = \frac{1}{2} q_i \boldsymbol{\xi}_{i_x} + \frac{1}{2}\sum_{j\neq i} q_j \frac{\partial \psi_j}{\partial x_i}\end{aligned} \tag{8}$$

which is different from $q_i\boldsymbol{\xi}_i$ with one extra term. By the nature of electrostatics, the potential at each charge i is the superposition of the potential contributed by all the remaining charges in the system. Let N_{i_j} denote the potential energy of charge i contributed by j, vice versa. For an electrostatic system defined on the 2-D plane, we have

$$N_{i_j}(\mathbf{v}) = -\frac{q_i q_j}{2\pi\epsilon_0}\ln\left(\frac{r_{i,j}(\mathbf{v})}{r_{\text{ref}}}\right) = N_{j_i}(\mathbf{v}) \tag{9}$$

where $r_{i,j}(\mathbf{v})$ is the physical distance between the two charges i and j based on the placement solution $\mathbf{v}$. r_{ref} is the reference distance where the potential by charge i (j) diminishes to zero, in this paper, we see it as the dimension of placement domain R. As a result, we have $N_{i_j}(\mathbf{v}) = N_{j_i}(\mathbf{v})$. Thus the mutual potential energy of each pair of charges i and j are equivalent. By the principle of potential superposition, we have

$$N_i(\mathbf{v}) = \sum_{j\neq i} N_{i_j}(\mathbf{v}) = \sum_{j\neq i} N_{j_i}(\mathbf{v}). \tag{10}$$

Therefore

$$\begin{aligned}\frac{\partial N(\mathbf{v})}{\partial x_i} &= \frac{1}{2}\left(\frac{\partial N_i(\mathbf{v})}{\partial x_i} + \frac{\partial\left(\sum_{j\neq i} N_j(\mathbf{v})\right)}{\partial x_i}\right) \\ &= \frac{\partial N_i(\mathbf{v})}{\partial x_i} = q_i \frac{\partial \psi_i(\mathbf{v})}{\partial x_i} = q_i \xi_{i_x}(\mathbf{v})\end{aligned} \tag{11}$$

so, $q_i\xi_{i_x}(\mathbf{v})$ is the actual gradient of $N(\mathbf{v})$ with respect to the horizontal movement Δx_i of the object i. Similarly, the density gradient of $N(\mathbf{v})$ with respect to the vertical movement of i is $q_i\xi_{i_y}(\mathbf{v})$. As a result, $q_i\boldsymbol{\xi}_i(\mathbf{v})$ is consistent with the gradient descent of the density cost (potential energy) function.

D. Correctness of Numerical Solution

Poisson's equation in (7) is solved via spectral methods [43] using a 2-D fast Fourier transform (FFT) applied to the spatial domain. Sinusoidal waveform approaches zero at the end of each function period, such behavior well matches the Neumann condition $\hat{\mathbf{n}} \cdot \nabla\psi(x, y) = \mathbf{0}$, $\forall (x, y) \in \partial R$ in (7), which requires zero gradient along the boundaries. As a result, we apply discrete sinusoidal transformation (DST) to the spatial field distribution $\boldsymbol{\xi}(x, y)$. As the electric potential and density distribution are the integral and derivative of the field, i.e., $\nabla\psi(x, y) = -\boldsymbol{\xi}(x, y)$ and $\rho(x, y) = \nabla \cdot \boldsymbol{\xi}(x, y)$, we reconstruct them via discrete cosine transformation (DCT). Based on an even mirroring and periodic extension, we have the DCT coefficients $a_{j,k}$ of the spatial density distribution $\rho(x, y)$ as

$$a_{j,k} = \frac{1}{m^2}\sum_{x=0}^{m-1}\sum_{y=0}^{m-1}\rho(x, y)\cos\left(w_j x\right)\cos(w_k y) \tag{12}$$

where w_j and w_k are frequency components. The density $\rho(x, y)$ can then be spatially expressed as

$$\rho(x, y) = \sum_{j=0}^{m-1}\sum_{k=0}^{m-1} a_{j,k}\cos\left(w_j x\right)\cos(w_k y). \tag{13}$$

As $\nabla \cdot \nabla\psi(x, y) = -\rho(x, y)$, we have the spatial potential distribution expressed as

$$\psi(x, y) = \sum_{j=0}^{m-1}\sum_{k=0}^{m-1}\frac{a_{j,k}}{w_j^2 + w_k^2}\cos\left(w_j x\right)\cos(w_k y). \tag{14}$$

Notice that for every pair of horizontal and vertical frequency components $\cos(w_j x)$ and $\cos(w_k y)$ from density $\rho(x, y)$ and potential $\psi(x, y)$, we have the Poisson's equation

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on June 12,2025 at 02:50:26 UTC from IEEE Xplore. Restrictions apply.

in (7) well satisfied in the numerical perspective as shown below

$$\begin{aligned}\nabla \cdot \nabla \psi(x, y) &= \frac{\partial^2 \psi(x, y)}{\partial x^2} + \frac{\partial^2 \psi(x, y)}{\partial y^2} \\ &= \left(-\sum_j \sum_k \frac{a_{j,k} w_j^2}{w_j^2 + w_k^2} \cos\big(w_j x\big) \cos(w_k y) \right) \\ &\quad + \left(-\sum_j \sum_k \frac{a_{j,k} w_k^2}{w_j^2 + w_k^2} \cos\big(w_j x\big) \cos(w_k y) \right) \\ &= -\sum_j \sum_k a_{j,k} \cos\big(w_j x\big) \cos(w_k y) = -\rho(x, y).\end{aligned}$$

We remove the dc component ρ_{avg} from $\rho(x, y)$ by setting $a_{0,0} = 0$. The spatial field distribution is expressed as

$$\begin{cases} \xi_x(x, y) = \sum_j \sum_k \frac{a_{j,k} w_j}{w_j^2 + w_k^2} \sin(w_j x) \cos(w_k y) \\ \xi_y(x, y) = \sum_j \sum_k \frac{a_{j,k} w_k}{w_j^2 + w_k^2} \cos(w_j x) \sin(w_k y) \end{cases} \tag{15}$$

which also satisfies (7) in the numerical perspective since

$$\begin{aligned}\nabla \psi(x, y) &= \left(\frac{\partial \psi(x, y)}{\partial x}, \frac{\partial \psi(x, y)}{\partial y} \right) \\ &= \left(-\sum_j \sum_k \frac{a_{j,k} w_j}{w_j^2 + w_k^2} \sin\big(w_j x\big) \cos(w_k y) \right. \\ &\qquad \left. -\sum_j \sum_k \frac{a_{j,k} w_k}{w_j^2 + w_k^2} \sin\big(w_j x\big) \cos(w_k y) \right) \\ &= \big(-\xi_x(x, y), -\xi_y(x, y) \big) = -\boldsymbol{\xi}(x, y).\end{aligned}$$

Given $|V| = n'$ movable objects in the netlist, ePlace-MS decomposes the placement region R into $m \times m$ grids, where $m = \sqrt{n'}$, to have one object per grid on average. The above 2-D-FFT computation thus costs exactly $O(n' \log n')$ runtime per iteration. As the number of fillers is at essentially the same order of the number of all the standard cells and movable macros, the complexity is essentially $O(n \log n)$. The well-formulated density gradient, global density smoothness and low computational complexity enables ePlace-MS to conduct placement on the flat netlist and the flat density grid with constantly high resolution. Compared to all the prior mixed-size nonlinear placers [3], [13], [15] with multilevel netlist clustering and grid coarsening, ePlace-MS avoids quality loss due to the suboptimal clustering and low density resolution especially at early iterations.

E. Advantage Analysis

Density force modeling remains quite a controversial problem [31] in quadratic placement, where the best location of anchor point for each object is usually unclear. relaxed quadratic spreading and linearization [45] nullifies the top 10% density force vectors to suppress over-spreading of standard cells, while the empirical tuning lacks theoretical support and may not guarantee convergence. Reference [7] uses Green's function to determine appropriate positions of cell anchors. The 2-D convolution makes the complexity to be $O(n^2)$ thus is computationally expensive. Kraftwerk2 [44] determines the anchor position via solution to the Poisson's equation. Due to the function order restriction in the quadratic placement infrastructure, the density cost is degraded from exponential to linear, which helps achieve convexity and efficiency but loses quality. SimPL [16] and ComPLx [17] determine the anchor position via recursive bi-partitioning, while convergence is theoretically promised via the primal-dual framework. Nevertheless, the solution quality is sensitive toward the initial solution. Moreover, it is hard to tell how much the optimum solution would follow the initial layout with minimum wirelength yet high overlap.

Nonlinear placement has no restriction on function orders thus ensures more flexibility in density modeling. However, the nonconvexity of the density function remains a headache to the nonlinear solvers. Bell-shape method [35] covers only adjacent grids in the local scale. Iterative grid uncoarsening is usually conducted in prior nonlinear placers [13], [15] to keep consistent with the scale of clustered netlist. However, the quality degradation due to low density resolution is not negligible. Besides, such local density smoothness would force objects to detour around obstacles thus inevitably lower the convergence rate. Notice that bell-shape method could realize fully global density smoothness by parameter adjustment, nevertheless, the regarding complexity scales up to $O(n^2)$, which is numerically expensive. Helmholtz equation in [3] smooths the density in global scale with only $O(n \log n)$ runtime complexity. However, sub-optimality in the choice of the linear factor ϵ in the Helmholtz equation [2, eq. (7)] introduces noises. Moreover, there is no formulation of density gradient functions in [3], where up to millions of constraints are simultaneously applied to all the grid density, which complicates the problem, degrades the placement quality and efficiency.

eDensity concisely formulates the placement density problem using the closed-form equation in (6). By differentiating it we derive the gradient vector to direct density cost reduction, where by (5) only one penalty factor is needed for force balancing with wirelength. eDensity numerically solves the partial differential equation via the spectral methods [43] in (14) and (15). Based on the nice properties of FFT, it consumes exactly only $O(n \log n)$ runtime per iteration. At each grid, the local electric potential and field are impacted by the global density distribution, while objects driven by density forces are able to freely move over blockages or macros, as Fig. 5 shows. Moreover, the global smoothness enables all the movable objects in over-filled regions to detect whitespace at remote area, as illustrated in Fig. 3, which helps quickly converge to the objective of even density. To this end, unlike all the prior methodologies in literature, eDensity approaches density equalization via directly simulating the behavior of a real electrostatic system, which in reality will always transfer toward the states of lower potential energy (until the energy decreases to zero), therefore theoretically guarantees the global convergence of eDensity. The nature of simulation enables us to use constantly high density resolution throughout the whole global placement, without any potential misguidance to the nonlinear solver.

V. Nonlinear Optimization for mGP

Our prior work shows high performance of Nesterov's method on placing standard cell-based circuits. In this section, we extend it to handle mixed-size placement, where we observe consistently good performance as shown by the experimental results in Section IX. In the framework of ePlace-MS (Fig. 1), mGP uses Nesterov's method to smoothly conduct simultaneous optimization on both macros and standard cells, as Fig. 5 shows. As a generalized approach, mGP

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on June 12,2025 at 02:50:26 UTC from IEEE Xplore. Restrictions apply.

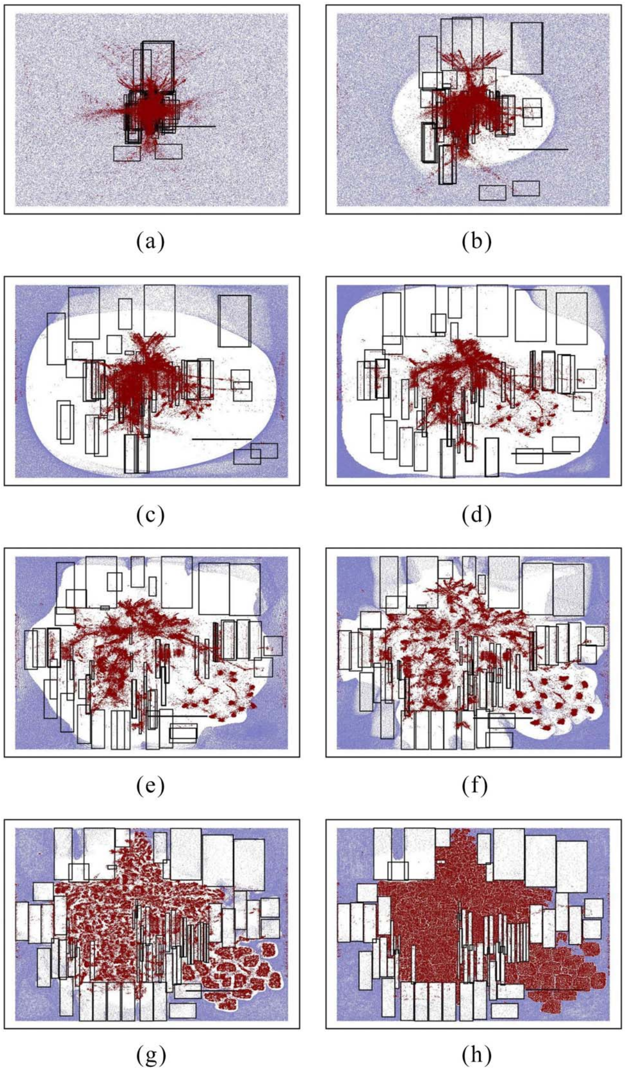

Fig. 5. Snapshots of mGP progression in ePlace-MS-WA on the MMS ADAPTEC1 benchmark with standard cells, macros, and fillers shown by red points, black rectangles, and blue points. Total wirelength, total potential energy, total density overflow, total OVLP, and total mOVLP are denoted by W, N, τ, O, and Om, respectively. (a) Iter = 50, $W = 34.76e6$, $N = 26.55e11$, $\tau = 84.31\%$, $O = 73.30e6$, and Om = 37.85e6. (b) Iter = 100, $W = 41.46e6$, $N = 91.4e10$, $\tau = 67.59\%$, $O = 59.11e6$, and Om = 24.40e6. (c) Iter = 125, $W = 44.44e6$, $N = 35.2e10$, $\tau = 55.31\%$, $O = 49.04e6$, and Om = 16.68e6. (d) Iter = 150, $W = 48.21e6$, $N = 98.78e9$, $\tau = 44.60\%$, $O = 40.10e6$, and Om = 98.40e5. (e) Iter = 175, $W = 52.74e6$, $N = 22.57e9$, $\tau = 39.72\%$, $O = 36.81e6$, and Om = 80.98e5. (f) Iter = 200, $W = 56.99e6$, $N = 46.31e8$, $\tau = 33.15\%$, $O = 31.56e6$, and Om = 43.09e5. (g) Iter = 235, $W = 61.05e6$, $N = 34.78e7$, $\tau = 20.34\%$, $O = 22.76e6$, and Om = 11.05e5. (h) Iter = 265, $W = 63.37e6$, $N = 25.51e6$, $\tau = 9.67\%$, $O = 16.48e6$, and Om = 58.11e4.

handles macros and standard cells in exactly the same way (see macro shifting at each netlist declustering level [13], formation of soft blocks by standard cells [46], [47], special density smoothing of macros [15], [18], macro shredding [17], etc.). In each iteration, we compute the gradient and preconditioner, predict the Lipschitz constant, and adjust steplength via backtracking. Nesterov's method solves the nonlinear problem iteratively till convergence is reached.

A. Existing Problems

Line search remains the major runtime bottleneck in conjugate gradient method,[2] which is widely used in prior nonlinear placers [15]. In practice, it is not guaranteed that the steplength output by line search could satisfy the conjugacy requirement [42]. Specifically, the vector of current search direction may not be orthogonal (with respect to the Hessian matrix of the cost function) to all the previous vectors. Therefore, local convergence rate of conjugate gradient method [11], $2(\sqrt{\kappa} - 1/\sqrt{\kappa} + 1)^k$, cannot be guaranteed. Instead of line search, Chen *et al.* [5] determined the steplength via upper-bounding the Euclidean distance of objects movement per iteration by a constant number. Such prediction assumes underestimation of steplength, which in general slows down the placement convergence rate. Moreover, steplength overestimation could still occur at some special area of the search space where the gradient changes sharply, therefore degrades the solution quality. As a result, a systematic solution with dynamic steplength adjustment and theoretical support becomes quite necessary.

Algorithm 1 Nesterov's Method in ePlace-MS

```
Input: $a_k$, $\mathbf{u}_k$, $\mathbf{v}_k$, $\mathbf{v}_{k-1}$, $\nabla f_{\text{pre}}(\mathbf{v}_k)$, $\nabla f_{\text{pre}}(\mathbf{v}_{k-1})$
Output: $\mathbf{u}_{k+1}$, $\mathbf{v}_{k+1}$, $a_{k+1}$
$\alpha_k = BkTrk\left(\mathbf{v}_k, \mathbf{v}_{k-1}, \nabla f_{\text{pre}}(\mathbf{v}_k), \nabla f_{\text{pre}}(\mathbf{v}_{k-1})\right)$
$\mathbf{u}_{k+1} = \mathbf{v}_k - \alpha_k \nabla f_{\text{pre}}(\mathbf{v}_k)$
$a_{k+1} = \left(1 + \sqrt{4a_k^2 + 1}\right)/2$
$\mathbf{v}_{k+1} = \mathbf{u}_{k+1} + (a_k - 1)(\mathbf{u}_{k+1} - \mathbf{u}_k)/a_{k+1}$
return
```

B. Nesterov's Method

The flow of Nesterov's method used in ePlace-MS is illustrated in Algorithm 1. We use Lipschitz constant prediction together with steplength backtracking to control the speed of optimization. a_k is an optimization parameter which is iteratively updated. There are two concurrently updated solutions, $\mathbf{u_k}$ and $\mathbf{v_k}$, where only $\mathbf{u}$ is output as the final solution (at the end of mGP and cGP), while $\mathbf{v}$ is used for steplength prediction. ∇f_{pre} denotes the preconditioned gradient vector, which will be discussed in Section VI. Initially, we set $a_0 = 1$ and have both $\mathbf{u}_0$ and $\mathbf{v}_0$ set as $\mathbf{v}_{\text{mIP}}$. *BkTrk* denotes steplength backtracking as shown in Section V-D. The convergence rate of Nesterov's method is proven to be $O(1/k^2)$ in [37], which achieves the upper-bound of global convergence rate of first-order optimization methods [36], on condition that the steplength α_k satisfies (16) at every single iteration k

$$f(\mathbf{v}_k) - f(\mathbf{v}_k - \alpha_k \nabla f(\mathbf{v}_k)) \geq 0.5\alpha_k \|\nabla f(\mathbf{v}_k)\|^2. \tag{16}$$

Bisection search is suggested by [37] to generate the maximal α_k without violating the inequality in (16). Similar to line search, it usually introduces significant runtime overhead. As [37] claims, the function $f(\mathbf{v}_k - \alpha\nabla f(\mathbf{v}_k))$ would be evaluated by $O(\log L)$ times along the search direction for a single iteration, increasing the complexity to $O(n \log n \log L)$. Here L is the Lipschitz constant as defined in Definition 1. As a result, step length prediction becomes necessary to accelerate the optimization process.

C. Lipschitz Constant Prediction

Instead of line search, we compute the steplength through a closed-form formula of the Lipschitz constant of the gradient.

[2] Our empirical studies on FFTPL [24] show that line search takes more than 60% of the total runtime on placing ADAPTEC1 of ISPD 2005.

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on June 12,2025 at 02:50:26 UTC from IEEE Xplore. Restrictions apply.

Definition 1: Given a multivariate convex function $f(\mathbf{v}) \in C^{1,1}(H)$, $\exists L > 0$ s.t. $\forall \mathbf{u}, \mathbf{v} \in H$

$$\|\nabla f(\mathbf{u}) - \nabla f(\mathbf{v})\| \leq L\|\mathbf{u} - \mathbf{v}\|. \tag{17}$$

H as Hilbert space is a generalized notion of Euclidean space, $C^{1,1}(H)$ requires $f(\mathbf{v})$ with Lipschitz continuous gradient. As our objective is nonconvex, we leverage Nesterov's method in an approximate way. Reference [37] states that $\alpha_k = L^{-1}$ satisfies the steplength requirement specified in (16) but lacks a formal proof. The rationale behind is that smaller Lipschitz constant indicates higher smoothness of the gradient thus faster convergence can be achieved via larger steplength, vice versa. Here we provide a proof to the statement that $\alpha_k = L^{-1}$ always satisfies (16) as Theorem 1.

Theorem 1: Given convex $f \in C^{1,1}(H)$ and L defined in Definition 1, $\alpha \leq L^{-1}$ satisfies (16).

Proof: $\forall \mathbf{u}, \mathbf{v} \in H$, we have

$$\begin{aligned} & f(\mathbf{v}) - f(\mathbf{u}) - \langle \nabla f(\mathbf{u}), \mathbf{v} - \mathbf{u} \rangle \\ &= \int_{\mathbf{u}}^{\mathbf{v}} \nabla f\left(\mathbf{v}'\right) d\mathbf{v}' - \langle \nabla f(\mathbf{u}), \mathbf{v} - \mathbf{u} \rangle \\ &= \int_{0}^{1} \nabla f(\mathbf{u} + \tau(\mathbf{v} - \mathbf{u})) d\left(\tau(\mathbf{v} - \mathbf{u})\right) - \langle \nabla f(\mathbf{u}), \mathbf{v} - \mathbf{u} \rangle \\ &= \int_{0}^{1} \langle \nabla f(\mathbf{u} + \tau(\mathbf{v} - \mathbf{u})) - \nabla f(\mathbf{u}),\ \mathbf{v} - \mathbf{u} \rangle d\tau \\ &\leq \int_{0}^{1} \|\nabla f(\mathbf{u} + \tau(\mathbf{v} - \mathbf{u})) - \nabla f(\mathbf{u})\| \cdot \|\mathbf{v} - \mathbf{u}\| d\tau \\ &\leq \int_{0}^{1} L \cdot \|\tau(\mathbf{v} - \mathbf{u})\| \cdot \|\mathbf{v} - \mathbf{u}\| d\tau \\ &= 0.5L\|\mathbf{v} - \mathbf{u}\|^2 \end{aligned} \tag{18}$$

where the first and second inequalities hold based on the Cauchy–Schwartz inequality [1] and the definition of Lipschitz constant in (17), respectively. Equation (18) indicates that

$$f(\mathbf{v}) \leq f(\mathbf{u}) + \langle \nabla f(\mathbf{u}), \mathbf{v} - \mathbf{u} \rangle + 0.5L\|\mathbf{v} - \mathbf{u}\|^2. \tag{19}$$

Let $\mathbf{u} = \mathbf{v}_k$ and $\mathbf{v} = \mathbf{v}_k - \alpha_k \nabla f(\mathbf{v}_k)$, based on (19)

$$\begin{aligned} f(\mathbf{u}) - f(\mathbf{v}) &= f(\mathbf{v}_k) - f(\mathbf{v}_k - \alpha_k \nabla f(\mathbf{v}_k)) \\ &\geq \langle \nabla f(\mathbf{u}), \mathbf{u} - \mathbf{v} \rangle - 0.5L\|\mathbf{v} - \mathbf{u}\|^2 \\ &= \langle \nabla f(\mathbf{v}_k), \alpha_k \nabla f(\mathbf{v}_k) \rangle - 0.5\alpha_k^2 L\|\nabla f(\mathbf{v}_k)\|^2 \\ &= \alpha_k \|\nabla f(\mathbf{v}_k)\|^2 - 0.5\alpha_k^2 L\|\nabla f(\mathbf{v}_k)\|^2 \\ &\geq (\alpha_k - 0.5\alpha_k^2 \alpha_k^{-1})\|\nabla f(\mathbf{v}_k)\|^2 \\ &= 0.5\alpha_k \|\nabla f(\mathbf{v}_k)\|^2 \end{aligned} \tag{20}$$

where the second inequality holds if we have $L \leq \alpha_k^{-1}$. ■

As a result, L^{-1} can be used as the steplength to accelerate the algorithm without convergence penalty. Exact Lipschitz constant is very expensive to compute (even more time consuming than line search). Moreover, static estimation will be invalidated through iterative change of the cost function, as both the wirelength coefficient γ in (3) and penalty factor λ in (5) are being iteratively adjusted in ePlace-MS (more details can be found in [24]). As a result, we approximate the Lipschitz constant and steplength as follows:

$$\widetilde{L}_k = \frac{\|\nabla f(\mathbf{v}_k) - \nabla f(\mathbf{v}_{k-1})\|}{\|\mathbf{v}_k - \mathbf{v}_{k-1}\|},\ \alpha_k = \widetilde{L}_k^{-1} \tag{21}$$

where only $\mathbf{v}$ is used for Lipschitz constant prediction. The computation overhead is negligible since both $\nabla f(\mathbf{v}_{k-1})$ and $\nabla f(\mathbf{v}_k)$ are known thus there is no extra computation.

Algorithm 2 *BkTrk*

```
Input: $a_k, a_{k+1}, \mathbf{u}_k, \mathbf{v}_k, \mathbf{v}_{k-1}, \nabla f_{\text{pre}}(\mathbf{v}_k), \nabla f_{\text{pre}}(\mathbf{v}_{k-1})$
Output: $\alpha_k$
1: $\hat{\alpha}_k = \frac{\|\mathbf{v}_k-\mathbf{v}_{k-1}\|}{\|\nabla f_{\text{pre}}(\mathbf{v}_k)-\nabla f_{\text{pre}}(\mathbf{v}_{k-1})\|}$
2: $\hat{\mathbf{u}}_{k+1} = \mathbf{v}_k - \hat{\alpha}_k \nabla f_{\text{pre}}(\mathbf{v}_k)$
3: $\hat{\mathbf{v}}_{k+1} = \hat{\mathbf{u}}_{k+1} + (a_k - 1)\left(\hat{\mathbf{u}}_{k+1} - \mathbf{u}_k\right)/a_{k+1}$
4: while $\hat{\alpha}_k > \epsilon \left(\frac{\|\hat{\mathbf{v}}_{k+1}-\mathbf{v}_k\|}{\|\nabla f_{\text{pre}}(\hat{\mathbf{v}}_{k+1})-\nabla f_{\text{pre}}(\mathbf{v}_k)\|}\right)$ do
5:     $\hat{\alpha}_k = \frac{\|\hat{\mathbf{v}}_{k+1}-\mathbf{v}_k\|}{\|\nabla f_{\text{pre}}(\hat{\mathbf{v}}_{k+1})-\nabla f_{\text{pre}}(\mathbf{v}_k)\|}$
6:     $\hat{\mathbf{u}}_{k+1} = \mathbf{v}_k - \hat{\alpha}_k \nabla f_{\text{pre}}(\mathbf{v}_k)$
7:     $\hat{\mathbf{v}}_{k+1} = \hat{\mathbf{u}}_{k+1} + (a_k - 1)\left(\hat{\mathbf{u}}_{k+1} - \mathbf{u}_k\right)/a_{k+1}$
8: end while
9: $\alpha_k = \hat{\alpha}_k$
10: return
```

D. Steplength Backtracking

We develop a backtracking method to enhance the prediction accuracy via preventing potential steplength overestimation by (21), which would unexpectedly misguide the nonlinear solver. Being used to generate $\mathbf{v}_{k+1}$, however, α_k by (21) is predicted using $\mathbf{v}_k$ and $\mathbf{v}_{k-1}$. Instead, our backtracking method predicts α_k using $\mathbf{v}_k$ and $\mathbf{v}_{k+1}$. At line 1 of Algorithm 2, we set the steplength computed by (21) as a temporary variable $\hat{\alpha}_k$. The respective temporary solution $\hat{\mathbf{v}}_{k+1}$ (line 3) is used to produce a reference steplength. If it is exceeded by $\hat{\alpha}_k$ (line 4), we update $\hat{\alpha}_k$ and $\hat{\mathbf{v}}_{k+1}$ at lines 5 and 7 and do the backtracking circularly until the inequality at line 4 is satisfied. $\mathbf{v}_k$ and $\mathbf{v}_{k-1}$ are the placement solutions for the current iteration k and the past iteration $k - 1$. $\mathbf{u}_k$ is the other solution (at iteration k) simultaneously updated with $\mathbf{v}_k$, as shown in Algorithm 1. $\epsilon = 0.95$ is the scaling factor to encourage earlier return of function *BkTrk* thus prevent over-backtracking, which could consume too much runtime with limited accuracy improvement. The runtime overhead is zero if the first check at line 4 is passed, since the newly computed gradient $\nabla f(\hat{\mathbf{v}}_{k+1})$ can be reused at the following iteration. Experiments show that the average number of backtracks per iteration over all the 16 MMS benchmarks [47] is only 1.037, indicating less than 4% runtime overhead on mGP. Disabling backtracking causes ePlace-MS-WA (using the weighted-average wirelength model) to fail on MMS BIGBLUE4 and increase wirelength by 43.12% on average of the remaining 15 MMS benchmarks.

VI. Preconditioning

This section introduces our development of the nonlinear preconditioner, which is used by Nesterov's method in Algorithm 1 and steplength backtracking in Algorithm 2. Preconditioning reduces the condition number of a problem, which is transformed to be more suitable for numerical solution. Traditional preconditioning techniques compute the

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on June 12,2025 at 02:50:26 UTC from IEEE Xplore. Restrictions apply.

inverse of the Hessian matrix $\mathbf{H_f}$ of the objective function f. Preconditioning has broad application in quadratic placers [17], [21], [45], [46] but none attempts in nonlinear placers [3], [13], [15], mainly due to the nonconvexity of the density function. In this paper, we approximate the original Hessian $\mathbf{H_f}$ with a positive definite diagonal matrix $\widetilde{\mathbf{H}}_\mathbf{f}$ as the preconditioner. We multiply it to the gradient vector and use $\nabla f_{\text{pre}} = \widetilde{\mathbf{H}}_\mathbf{f}^{-1} \nabla f$ to direct the nonlinear placement optimization. A preconditioned gradient vector ∇f_{pre} is used to stretch the function space to be more spherical in order to smooth and accelerate the numerical optimization. However, as the objective function of global placement is of large scale (usually millions of objects to place) and highly nonlinear, to compute the Hessian matrix becomes very expensive and indeed computationally impractical. As a result, we choose the Jacobi preconditioner using only the diagonal terms of the Hessian matrix $\mathbf{H_f}$

$$\mathbf{H}_{\mathbf{f}_{\mathbf{x},\mathbf{x}}} \approx \widetilde{\mathbf{H}}_{\mathbf{f}_{\mathbf{x},\mathbf{x}}} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & 0 & \cdots & 0 \\ 0 & \frac{\partial^2 f}{\partial x_2^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}. \tag{22}$$

By (5), we have

$$\widetilde{\mathbf{H}}_\mathbf{f} = \begin{pmatrix} \widetilde{\mathbf{H}}_{\mathbf{f}_{\mathbf{x},\mathbf{x}}} & 0 \\ 0 & \widetilde{\mathbf{H}}_{\mathbf{f}_{\mathbf{y},\mathbf{y}}} \end{pmatrix} = \widetilde{\mathbf{H}}_\mathbf{W} + \lambda \widetilde{\mathbf{H}}_\mathbf{N}. \tag{23}$$

As $(\partial^2 f(\mathbf{v})/\partial x_i^2) = (\partial^2 W(\mathbf{v})/\partial x_i^2) + \lambda(\partial^2 N(\mathbf{v})/\partial x_i^2)$, we need to separately compute or estimate $(\partial^2 W/\partial x_i^2)$ and $(\partial^2 N/\partial x_i^2)$ at every iteration.

A. Wirelength

Based on the LSE wirelength modeling equation shown in (3), we differentiate it to derive the gradient function of the wirelength of net e with respect to x_i as shown below

$$\begin{aligned} \frac{\partial W_e^{\text{LSE}}(\mathbf{v})}{\partial x_i} &= \frac{\gamma}{\sum_{j\in e} \exp\left(x_j/\gamma\right)} \times \frac{\partial \sum_{j\in e} \exp\left(x_j/\gamma\right)}{\partial x_i} \\ &\quad + \frac{\gamma}{\sum_{j\in e} \exp\left(-x_j/\gamma\right)} \times \frac{\partial \sum_{j\in e} \exp\left(-x_j/\gamma\right)}{\partial x_i} \\ &= \frac{\exp\left(x_i/\gamma\right)}{\sum_{j\in e} \exp\left(x_j/\gamma\right)} - \frac{\exp\left(-x_i/\gamma\right)}{\sum_{j\in e} \exp\left(-x_j/\gamma\right)}. \end{aligned} \tag{24}$$

Via further differentiating (24) with respect to x_i, we are able to derive the second-order gradient of the LSE function as below

$$\begin{aligned} \frac{\partial^2 W_e^{\text{LSE}}(\mathbf{v})}{\partial x_i^2} &= \frac{\exp\left(x_i/\gamma\right)\left\{\sum_{j\in e} \exp\left(x_j/\gamma\right) - \exp\left(x_i/\gamma\right)\right\}}{\gamma\left\{\sum_{j\in e} \exp\left(x_j/\gamma\right)\right\}^2} \\ &\quad + \frac{\exp\left(-x_i/\gamma\right)\left\{\sum_{j\in e} \exp\left(-x_j/\gamma\right) - \exp\left(-x_i/\gamma\right)\right\}}{\gamma\left\{\sum_{j\in e} \exp\left(-x/\gamma\right)\right\}^2}. \end{aligned} \tag{25}$$

Similarly, we can derive the gradient function of the WA wirelength model by differentiating (4). However, the second-order differentiation of (4) is complicated, moreover, quite computationally expensive. As a result, we use the vertex degree of object i instead

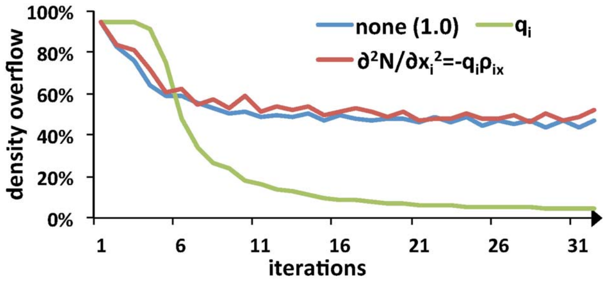

Fig. 6. Performance comparison of the three candidate density preconditioners via a density-only placement on the MMS ADAPTEC1 benchmark.

$$\frac{\partial^2 W_e^{\text{WA}}(\mathbf{v})}{\partial x_i^2} = \sum_{e\in E_i} \frac{\partial^2 W_e(\mathbf{v})}{\partial x_i^2} \approx |E_i| \tag{26}$$

where E_i denote the set of all the nets incident to the object i. We have the second-order derivative of the wirelength function $W(\mathbf{v})$ with respect to the horizontal movement of object i (i.e., x_i) expressed as below

$$\frac{\partial^2 W(\mathbf{v})}{\partial x_i^2} = \frac{\partial^2 \sum_{e\in E_i} W_e(\mathbf{v})}{\partial x_i^2} = \sum_{e\in E_i} \frac{\partial^2 W_e(\mathbf{v})}{\partial x_i^2}. \tag{27}$$

Since $W(\mathbf{v})$ in both LSE and WA are strongly convex [12], [35] and globally differentiable, the Hessian matrices are also positive definite with straightly positive eigenvalues. As a result, we can use the closed-form formula $(\partial^2 W/\partial x_i^2)$ in (27) as the nonlinear wirelength preconditioner.

B. Density

By differentiating the density gradient function in (11), we could obtain the second-order derivative as below

$$\frac{\partial^2 N(\mathbf{v})}{\partial x_i^2} = -q_i \frac{\partial \xi_{i_x}}{\partial x_i} = -q_i \rho_{i_x} \tag{28}$$

where $\rho_i = \rho_{i_x} + \rho_{i_y}$. However, the density function $N(\mathbf{v})$ by (6) is based on a repulsive force dominant system, thus it is nonconvex. As a result, we could have $(\partial^2 N(\mathbf{v})/\partial x_i^2) < 0$ for some object i. Negative preconditioner will invert the direction of gradient, causing the cost to increase and the placement solution to diverge. To avoid this, we concisely approximate the density preconditioner as below

$$\frac{\partial^2 N(\mathbf{v})}{\partial x_i^2} = q_i \frac{\partial^2 \psi_i(\mathbf{v})}{\partial x_i^2} \approx q_i. \tag{29}$$

Such operation actually helps decompose charges of different electric quantities all into unit charges, the electric force applied onto each charge is uniquely determined by the local electric field, while placement oscillation due to imbalance of density forces is avoided. The rationale behind is similar to the mechanical movement, where the motion velocity of each object depends on its acceleration, which is uniquely determined by the respective field (electrostatic, gravitational, etc.) but not the mass of the object. As a result, our density equalization method is indeed a simulation of the behavior of a real electrostatic system. Such system in the real world will always progress toward states of lower energy, which guarantees the convergence in the end achieving the even density

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on June 12,2025 at 02:50:26 UTC from IEEE Xplore. Restrictions apply.

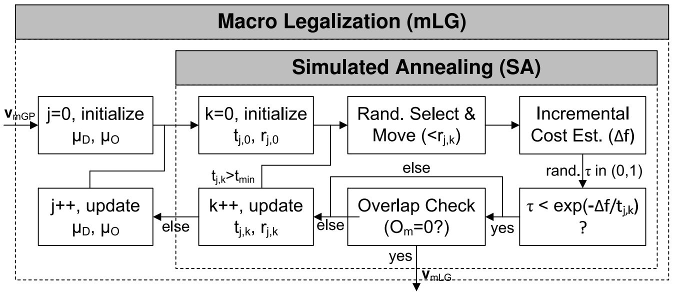
Macro Legalization (mLG)
Simulated Annealing (SA)
$\mathbf{v}_{mGP}$
j=0, initialize μ_D, μ_O
k=0, initialize $t_{j,0}$, $r_{j,0}$
Rand. Select & Move (<$r_{j,k}$)
Incremental Cost Est. (Δf)
rand. τ in (0,1)
else
$t_{j,k}>t_{min}$
j++, update μ_D, μ_O
k++, update $t_{j,k}$, $r_{j,k}$
Overlap Check (O_m=0?)
τ < exp(-Δf/$t_{j,k}$) ?
else
else
yes
yes
$\mathbf{v}_{mLG}$

Fig. 7. Our two-level annealing-based macro legalizer.

distribution. The performance comparison of the three density preconditioners [no preconditioner, (29) and (28)] is shown in Fig. 6. Compared to the other two options, our proposed preconditioner using charge quantity q_i (object area) achieves the highest effectiveness and efficiency in the convergence of the density cost minimization.

C. Summary

As a result, we use $(\partial^2 f(\mathbf{v})/\partial x_i^2) \approx (\partial^2 W(\mathbf{v})/\partial x_i^2) + \lambda q_i$ to approximate the ith diagonal term of the placement preconditioner $\widetilde{\mathbf{H}}_{\mathbf{f_{x,x}}}$ with respect to horizontal charge movement, while the preconditioner for the vertical charge movement can be derived in a similar way. Disabling the preconditioner causes ePlace-MS to fail on nine out of the totally 16 MMS benchmarks, since macros significantly differ from standard cells with much higher magnitude of gradients. As a result, unpreconditioned gradient makes macros with large area and high incidence degree to bounce between opposite placement boundaries, causing the solution to oscillate and hard to converge within limited number of iterations.[3] On average of the remaining seven MMS benchmarks, the wirelength is increased by 24.63%, indicating high effectiveness of our preconditioner.

VII. mLG

Based on the mGP solution $\mathbf{v}_{\text{mGP}}$, mLG legalizes the macro layout via a simulated annealing (SA) [19] based approach, as Fig. 8 shows. Unlike traditional SA-based floorplanners and macro placers [4], [6], [32], [46] which perturb floorplan expression then physically realize it, mLG uses SA to directly control macro motion.

1) We expect a high-quality solution from mGP. Only local macro shifts are expected in mLG, the shrunk design space can be well explored by SA.
2) Our SA engine is more efficient with only minor position change to each single macro.
3) After each random perturbation of floorplan expression, the respective floorplan realization may cause significant layout change, which is time consuming and could induce unexpected quality degradation.

Similar to Timberwolf [40], however, mLG legalizes macros rather than detailedly place cells. As Fig. 7 shows, mLG can be decomposed into two levels. At each iteration of the outer loop (mLG iteration), we update the cost $f_{\text{mLG}}(\mathbf{v})$ by

$$f_{\text{mLG}}(\mathbf{v}) = \text{HPWL}(\mathbf{v}) + \mu_D D(\mathbf{v}) + \mu_O O_m(\mathbf{v}) \tag{30}$$

where HPWL($\mathbf{v}$), $D(\mathbf{v})$, and $O_m(\mathbf{v})$ denote the total wirelength, total standard-cell area covered by macros and total mOVLP. mLG is set as constrained optimization.

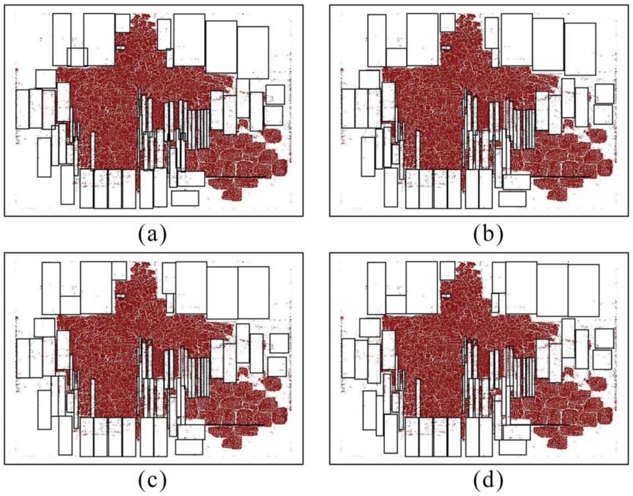
(a)
(b)
(c)
(d)

Fig. 8. Distribution of macros (a) before mLG (b) 1st mLG iteration (c) 2nd mLG iteration (d) after mLG by ePlace-MS-WA on the MMS ADAPTEC1 benchmark with fixed standard-cell layout and all the fillers removed. Total wirelength, total standard-cell area covered by macros, total OVLP, and total mOVLP are denoted by W, D, O, and Om, respectively. (a) $j = 0, k = 0, W = 63.37e6, D = 12.29e5, O = 16.48e6, Om = 60.94e4$. (b) $j = 0, k = 1, W = 63.61e6, D = 13.57e5, O = 16.17e6, Om = 17.14e4$. (c) $j = 0, k = 9, W = 64.19e6, D = 14.55e5, O = 16.03e6, Om = 2.16e4$. (d) $j = 3, k = 1, W = 64.36e6, D = 14.83e5, O = 16.08e6, Om = 0$.

1) Objective is to minimize HPWL($\mathbf{v}$) + $\mu_D D(\mathbf{v})$. Since penalty on $D(\mathbf{v})$ will be transformed to wirelength during cGP and cDP, we treat them equally in mLG thus statically set $\mu_D = (\text{HPWL}(\mathbf{v})/D(\mathbf{v}))$.
2) Constraint is zero mOVLP ($O_m(\mathbf{v}) = 0$). We set μ_O as the penalty factor and initialize it as $(\text{HPWL}(\mathbf{v}) + \mu_D D(\mathbf{v}))/O_m(\mathbf{v})$. μ_O is multiplied by β at each mLG iteration to make the legalizer more aggressive on mOVLP reduction.

At each iteration of the inner loop shown in Fig. 7 (SA iteration), the annealer randomly picks a macro and randomly determine its motion vector within the search range. The cost difference Δf is then incrementally evaluated and we generate a random number $\tau \in (0, 1)$ to determine whether the new layout will be accepted or not by checking if $\tau < \exp(-\Delta f/t_{j,k})$. Here j and k denote the mLG and SA iteration indices. The temperature $t_{j,k}$ at each iteration (j, k) is determined based on the maximum cost increase $\Delta f_{\max}(j, k)$ that will be accepted by more than 50% probability, thus we set $t_{j,k} = (\Delta f_{\max}(j, k)/\ln 2)$. We set $\Delta f_{\max}(j, 0)$ $(\Delta f_{\max}(j, k_{\max}))$ as $0.03 \times \beta^j$ $(0.0001 \times \beta^j)$, denoting that cost increase by less than 3% (0.01%) at the first (last) SA iteration will be accepted by more than 50% probability. These parameters appear small but fit well into our framework, since only minor layout change is expected in mLG. Meanwhile, they are scaled up per mLG iteration to adapt to the enhancement of the penalty factor μ_O. We initialize $\Delta f_{\max}(j, k)$ by $\Delta f_{\max}(j, 0)$ and linearly decrease it toward $\Delta f_{\max}(j, k_{\max})$. The radius $r_{j,k}$ of macro motion range is dependent on both the penalty factor and the amount of macros. Given m macros to legalize, we set $r_{j,0} = (R_x/\sqrt{m}) \times 0.05 \times \beta^j$, which means the entire placement region R can be decomposed into m sub-regions, every macro can be moved within 5% of its assigned region at each time. Similar to the temperature, the radius is scaled by β at each mLG iteration. In practice, we set $\beta = 1.5$ to achieve good tradeoff between quality and efficiency.

[3] We set 3000 as the upper limit of iterations in ePlace-MS.

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on June 12,2025 at 02:50:26 UTC from IEEE Xplore. Restrictions apply.

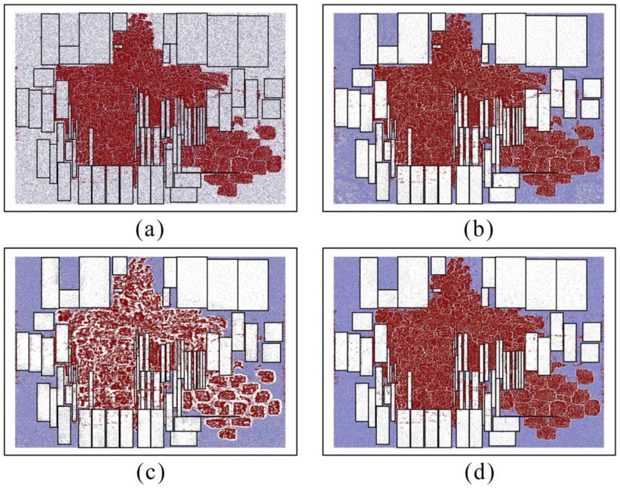

Fig. 9. Distribution of standard cells and fillers (a) before cGP (b) after filler redistribution (c) standard cell and filler co-optimization (d) after cGP by ePlace-MS-WA on the MMS ADAPTEC1 benchmark. The total wirelength, potential energy, density overflow, and OVLP are denoted by W, N, τ, and O, respectively. Total mOVLP remains zero (by mLG). (a) $Iter = 0, W = 64.36e6, N = 42.05e9, \tau = 9.84\%, O = 16.09e6$. (b) $Iter = 20, W = 64.36e6, N = 78.27e6, \tau = 9.84\%, O = 16.09e6$. (c) $Iter = 28, W = 61.30e6, N = 21.70e7, \tau = 19.96\%, O = 22.83e6$. (d) $Iter = 51, W = 63.04e6, N = 15.29e6, \tau = 9.81\%, O = 16.29e6$.

VIII. cGP

cGP mitigates the quality overhead due to mLG via a second-phase global placement of standard cells. With macros fixed, cGP uses the same nonlinear algorithm of mGP. As Fig. 9 shows, cGP causes small changes to the standard-cell layout and converges faster than mGP. mLG is unaware of filler layout and may induce huge macro-filler overlap. As a result, fillers are retrieved and randomly distributed first. With standard cells fixed, a filler-only placement runs for 20 iterations to relocate fillers to their best sites. The resulting solution with low density cost ensures placement of standard cells not to compensate extra density violation by fillers. On average of all the MMS benchmarks, the wirelength will be increased by 6.53% if we skip filler-only placement.

cGP then co-optimizes standard cells with fillers. The initial penalty factor $\lambda_{\text{cGP}}^{\text{init}}$ is determined based on that by the last mGP iteration $\lambda_{\text{mGP}}^{\text{last}}$. As λ will be multiplied by up to 1.1, we set $\lambda_{\text{cGP}}^{\text{init}} = \lambda_{\text{mGP}}^{\text{last}} \times 1.1^m$, using m buffering iterations for cGP to recover the aggressiveness of mGP. As cGP section of Fig. 2 shows, the wirelength (overlap) reduces (increases) sharply for an optimal initial solution (similar to mIP). By increasing λ_{cGP} iteratively, cGP reduces the overlap with mild wirelength overhead. In practice, we set m as the number of mGP iterations divided by 10.

IX. Experiments and Results

We implement ePlace-MS using C programming language and execute the program in a Linux machine with Intel i7 920 2.67 GHz CPU and 12 GB memory. The FFT package from [38] is used in ePlace-MS to perform DCT and DST operations. To validate the performance of ePlace-MS, we conduct experiments on the MMS benchmarks [47], as shown in Table I. MMS benchmarks inherit the same netlists and density constraints ρ_t from ISPD 2005 [34] and ISPD 2006 [33] benchmarks but have all the macros freed to place. There are also fixed IO blocks inserted within the placement domain in order to maintain the uniqueness of the analytic solution. Following the contest policy in ISPD 2006 [33], there is a benchmark-specific density upper-bound ρ_t for eight out of the totally 16 circuits. This target density ρ_t helps produce whitespace among circuit objects to accommodate interconnect and buffers, therefore facilitate the following design stages of routing, timing correction, etc. By the benchmark protocol [33], exceeding ρ_t will penalize the wirelength by $s\text{HPWL} = \text{HPWL} \times \left(1 + 0.01 \times \tau_{\text{avg}}\right)$, where τ_{avg} denotes the scaled density overflow per bin and scaled HPWL (sHPWL) is the scaled wirelength. More detailed circuit statistics of MMS benchmarks can be found in [47]. After cGP is completed, ePlace-MS invokes the detailed placer in [13] for the legalization and cDP. There is no benchmark specific parameter tuning in this paper, and we use the official scripts from [47] to evaluate the performance of all the placers in our experiments.

Seven state-of-the-art mixed-size placers covering two categories of algorithms (as discussed in Section I) are included for performance comparison, namely, Capo10.5 [39], FastPlace3.0 [46], ComPLx (v13.07.30) [17], POLAR [21], mPL6 [3], FLOP [47], and NTUplace3-unified [13]. We have obtained the binaries of four placers and executed them on our machine. FLOP is not available due to IP and other issues, thus we cite their performance from [47]. Capo10.5 and mPL6 fail to work with MMS benchmarks in our machine, so we cite the respective results also from [47] instead. Also, APlace3 [15] crashes on every MMS circuit as reported in [47] thus is not included in the results. MP-tree [6] and CG [4] are not available due to the industrial copyrights, while their results on MMS benchmarks are also not available. However, as both of them have been outperformed by NTUplace3-unified [13] with on average 21% and 9% shorter wirelength (reported in [13, Table V]), we do not include them in our experiments.

The experimental results of HPWL and sHPWL on the MMS circuits are shown in Table II. As shown in Table I, there are no target density constraints for the first eight circuits (i.e., 100%) thus no density penalty on the wirelength. In other words, HPWL equals sHPWL for the first eight MMS testcases in Table II as marked with †. NTUplace3-unified-NR (with macro rotation and flipping disabled) fails on two MMS benchmarks (NEWBLUE3 and NEWBLUE7) with the average wirelength, density overflow, and runtime computed based on the other fourteen benchmarks. Compared to all the placers in the experiments, ePlace-MS produces the best solutions with the shortest wirelength for fourteen out of the totally 16 testcases. Besides, it outperforms the leading-edge mixed-size placer NTUplace3 [13] by up to 22.98% shorter wirelength[4] and on average 8.22% shorter wirelength over all the MMS circuits. Notice that unlike NTUplace3, ePlace-MS does not allow macro rotation or flipping, which indicates further improvement space thus potentially better solution quality. The statistics of density overflow (i.e., the amount of violations to the testcase dependent target density ρ_t as specified in Table I) is shown in Table III. The respective results of Capo10.5, FLOP-NR, and mPL6 are not available from respective publications [47]. ePlace-MS obtains consistently the lowest density overflow at all the eight testcases (with predefined target density), showing the best performance of our density modeling method eDensity. The runtime statistics is shown in Table IV. On average of

[4] ePlace-MS produces 22.98% shorter wirelength than NTUplace3 on NEWBLUE7, which is the largest design in the MMS benchmark suite with roughly 2.5 million components.

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on June 12,2025 at 02:50:26 UTC from IEEE Xplore. Restrictions apply.

TABLE I
Statistics of the MMS Benchmark Suite [47]

Circuits	# Objects	# Mov. Objects	# Std. Cells	# Macros	# Fixed I/Os	# Nets	# Pins	Target Den. (ρ_t)
ADAPTEC1	211447	210967	210904	63	480	221142	944053	100 %
ADAPTEC2	255023	254584	254457	127	439	266009	1069482	100 %
ADAPTEC3	451650	450985	450927	58	665	466758	1875039	100 %
ADAPTEC4	496054	494785	494716	69	1260	515951	1912420	100 %
BIGBLUE1	278164	277636	277604	32	528	284479	1144691	100 %
BIGBLUE2	557866	535741	534782	959	22125	577235	2122282	100 %
BIGBLUE3	1096812	1095583	1093034	2549	1229	1123170	3833218	100 %
BIGBLUE4	2177353	2169382	2169183	199	7970	2229886	8900078	100 %
ADAPTEC5	843128	842558	842482	76	570	867798	3493147	50 %
NEWBLUE1	330474	330137	330073	64	337	338901	1244342	80 %
NEWBLUE2	441516	440264	436516	3748	1252	465219	1773855	90 %
NEWBLUE3	494011	482884	482833	51	11127	552199	1929892	80 %
NEWBLUE4	646139	642798	642717	81	3341	637051	2499178	50 %
NEWBLUE5	1233058	1228268	1228177	91	4790	1284251	4957843	50 %
NEWBLUE6	1255039	1248224	1248150	74	6815	1288443	5307594	80 %
NEWBLUE7	2507954	2481533	2481372	161	26421	2636820	10104920	80 %

TABLE II
HPWL (Marked With †) and sHPWL ($\times 10^6$) on the MMS Benchmark Suite [47]. Mac = Macros, CP = Capo, FP = FastPlace, CPx = ComPLx, NP3U = NTUplace3-Unified, NR = No Rotation or Flipping of Macros. Cited Results Are Marked With *. All the Results Are Evaluated by the Official Scripts [47]

Categories	Constructive			One-Stage						ePlace-MS	
Benchmarks	CP10.5*	FLOP-NR*	FLOP*	FP3.0	CPx	POLAR	mPL6*	NP3U-NR	NP3U	LSE	WA
ADAPTEC1†	84.77	77.18	76.83	82.39	79.05	92.17	77.84	75.92	75.55	66.99	**66.82**
ADAPTEC2†	92.61	87.17	84.14	88.53	99.11	149.43	88.40	84.89	78.50	**76.74**	76.76
ADAPTEC3†	202.37	182.21	175.99	187.98	175.78	197.48	180.64	170.88	169.74	161.63	**161.55**
ADAPTEC4†	202.38	166.55	161.68	187.50	156.75	175.19	162.02	167.13	166.68	**145.89**	147.04
BIGBLUE1†	112.58	95.45	94.92	104.91	96.18	99.12	99.36	96.42	96.57	87.27	**86.29**
BIGBLUE2†	149.54	150.66	153.02	145.89	147.19	157.72	144.37	148.12	147.17	132.72	**130.06**
BIGBLUE3†	583.37	372.79	346.24	400.40	344.63	420.28	319.63	324.39	338.47	287.34	**284.39**
BIGBLUE4†	915.37	807.53	777.84	775.43	772.53	814.07	804.00	797.17	799.66	660.17	**656.68**
ADAPTEC5	565.88	381.83	357.83	338.77	338.67	380.45	376.30	295.24	**294.24**	304.68	312.86
NEWBLUE1	110.54	73.36	67.97	73.91	65.26	70.68	66.93	61.13	61.25	**60.43**	61.87
NEWBLUE2	303.25	231.94	187.40	197.15	187.87	197.65	179.18	164.27	163.76	**159.11**	162.98
NEWBLUE3	1282.19	344.71	345.99	325.72	**269.47**	601.17	415.86	N/A	280.92	287.69	304.16
NEWBLUE4	300.69	256.91	256.54	270.70	256.97	277.60	277.69	231.59	229.36	**226.29**	229.20
NEWBLUE5	570.32	516.71	510.83	500.09	453.05	450.69	515.49	414.81	420.46	**392.77**	392.93
NEWBLUE6	609.16	502.24	493.64	512.19	452.83	475.78	482.44	471.51	474.86	414.56	**409.28**
NEWBLUE7	1481.45	1113.07	1078.18	1016.10	1010.00	1107.59	1038.66	N/A	1100.84	**889.18**	895.11
Avg. (s)HPWL	66.14%	20.16%	15.46%	19.47%	12.04%	32.03%	17.25%	8.61%	8.22%	−0.57%	0.00%

TABLE III
Scaled Average Density Overflow per Bin on the MMS Benchmark Suite [47]. Mac = Macros, CP = Capo, FP = FastPlace, CPx = ComPLx, NP3U = NTUplace3-Unified, NR = No Rotation or Flipping of Macros. Cited Results Are Marked With *

Categories	Constructive			One-Stage						ePlace-MS	
Benchmarks	CP10.5*	FLOP-NR*	FLOP*	FP3.0	CPx	POLAR	mPL6*	NP3U-NR	NP3U	LSE	WA
ADAPTEC5	N/A	N/A	4.19	2.41	1.00	5.48	N/A	4.59	5.34	0.09	0.75
NEWBLUE1	N/A	N/A	1.14	1.03	1.05	2.39	N/A	0.53	1.35	0.06	0.04
NEWBLUE2	N/A	N/A	0.87	0.07	0.19	0.02	N/A	0.12	0.05	0.05	0.03
NEWBLUE3	N/A	N/A	1.02	0.01	0.01	0.00	N/A	N/A	0.00	0.00	0.00
NEWBLUE4	N/A	N/A	4.94	2.62	1.35	10.43	N/A	8.69	10.1	0.35	0.29
NEWBLUE5	N/A	N/A	2.85	1.21	1.08	7.68	N/A	7.80	9.14	0.17	0.17
NEWBLUE6	N/A	N/A	1.34	1.11	1.06	5.10	N/A	1.53	2.09	0.27	0.23
NEWBLUE7	N/A	N/A	1.48	0.60	0.99	1.88	N/A	28.51	0.33	0.14	0.09
Avg. Den. Ovf.	N/A	N/A	27.64×	7.49×	7.69×	23.99×	N/A	17.65×	17.99×	1.15×	1.00

all the 16 MMS benchmarks, ePlace-MS runs faster than Capo10.5, FLOP, ComPLx, and mPL6, and shows essentially the same efficiency with NTUplace3. Despite longer runtime than FastPlace3.0 and POLAR, ePlace-MS produces on average 19.47% and 32.03% shorter wirelength. In general, ePlace-MS outperforms all the mixed-size placement algorithms in literature and achieves good results on both LSE and WA wirelength models, showing that our density function and nonlinear optimization algorithm have high and stable performance, which are not dependent on specific wirelength models.

Fig. 10 shows the CPU breakdown of ePlace-MS-WA on average of all the MMS benchmarks. mGP is the most effective placement stage (as Fig. 2 shows) and consumes the longest runtime. A further breakdown of mGP by Fig. 11 illustrates that computation of density and wirelength gradients and other operations (Lipschitz constant prediction, parameter update, etc.) consume 57%, 29%, and 14% runtime of mGP.

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on June 12,2025 at 02:50:26 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
RUNTIME (MINUTES) ON THE MMS BENCHMARK SUITE [47]. MAC = MACROS, CP = CAPO, FP = FASTPLACE, CPX = COMPLX, NP3U = NTUPLACE3-UNIFIED, NR = NO ROTATION OR FLIPPING OF MACROS. CITED RESULTS ARE MARKED WITH *

Categories	Constructive			One-Stage						ePlace-MS	
Benchmarks	CP10.5*	FLOP-NR*	FLOP*	FP3.0	CPx	POLAR	mPL6*	NP3U-NR	NP3U	LSE	WA
ADAPTEC1	92.78	12.03	13.73	3.32	5.43	4.85	40.07	5.63	6.50	5.25	5.47
ADAPTEC2	122.88	20.32	19.87	3.28	21.55	9.73	47.83	8.45	6.30	7.58	7.43
ADAPTEC3	282.88	32.38	35.27	6.73	14.07	10.72	99.72	15.10	10.57	26.22	27.23
ADAPTEC4	291.15	39.97	40.45	6.80	16.87	12.52	99.52	9.40	8.97	56.40	29.35
BIGBLUE1	140.97	29.60	33.45	5.05	4.25	4.95	47.15	12.07	10.90	7.85	7.82
BIGBLUE2	294.12	39.55	51.83	5.32	49.80	12.28	203.37	17.08	17.68	13.97	13.70
BIGBLUE3	91165.35	117.28	990.62	20.27	176.38	46.07	159.13	47.30	58.13	82.20	72.98
BIGBLUE4	1829.75	230.27	327.85	36.73	126.02	56.63	397.80	115.95	92.17	141.37	204.15
ADAPTEC5	399.28	84.13	54.88	15.22	13.15	21.37	377.27	53.78	46.07	50.27	48.35
NEWBLUE1	52.22	22.80	16.87	5.03	4.58	6.20	52.85	11.85	10.75	11.70	10.87
NEWBLUE2	135.93	44.10	40.23	6.52	51.02	21.93	100.73	13.55	15.00	51.12	62.40
NEWBLUE3	1222.32	38.93	45.95	12.08	36.30	39.27	293.72	N/A	58.08	30.57	17.53
NEWBLUE4	109.82	42.92	40.92	9.05	10.57	13.58	162.20	28.05	32.07	28.27	29.73
NEWBLUE5	275.80	146.68	152.72	21.60	35.78	30.62	413.43	82.03	77.50	55.47	63.40
NEWBLUE6	301.27	157.50	159.38	18.37	21.25	28.02	218.53	62.97	65.73	112.62	69.65
NEWBLUE7	723.10	312.75	418.40	50.53	73.75	66.77	528.00	N/A	116.03	392.02	191.47
Avg. CPU	13.71×	1.96×	2.06×	0.36×	1.09×	0.67×	5.92×	0.90×	1.00×	1.18×	1.00×

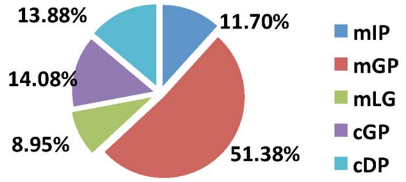

Fig. 10. Runtime breakdown of ePlace-MS-WA on average of all the 16 MMS benchmarks.

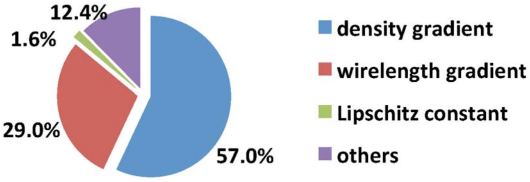

Fig. 11. Runtime breakdown of mGP of ePlace-MS-WA on average of all the 16 MMS benchmarks.

X. CONCLUSION

ePlace-*MS* is a generalized and effective placement algorithm to handle mixed-size circuits of very large scale. Using the density function eDensity based on electrostatics analogy, macros and standard cells are equalized by preconditioning and smoothly co-optimized by Nesterov's method. Steplength is determined via Lipschitz continuity together with a backtracking strategy to prevent overestimation. Unlike all the approaches in literature, ePlace-MS treats standard cells and macros in exactly the same way. The experimental results on MMS benchmarks validate its high and stable performance. The analytic nature of ePlace-MS ensures smooth integration of other design objectives (timing, routability, thermal, etc.). More details of ePlace-MS can be found at its homepage [8].

ACKNOWLEDGMENT

The authors would like to thank Dr. J. Z. Yan and Prof. C. Chu for providing the MMS placement benchmark suite, Dr. M.-K. Hsu and Prof. Y.-W. Chang for providing the binary of NTUplace3-unified, Dr. N. Viswanathan and Prof. C. Chu for providing the binary of FastPlace3.0 and FastPlace-DP, Dr. M.-C. Kim and Prof. I. L. Markov for providing the binary of SimPL and ComPLx, T. Lin and Prof. C. Chu for providing the binary of POLAR, and Y. Wang for the assistance in placement visualization.

REFERENCES

[1] K. M. Abadir and J. R. Magnus, *Matrix Algebra*. New York, NY, USA: Cambridge Univ. Press, 2005.

[2] T. Chan, J. Cong, and K. Sze, "Multilevel generalized force-directed method for circuit placement," in *Proc. Int. Symp. Phys. Design (ISPD)*, Monterey, CA, USA, 2005, pp. 185–192.

[3] T. F. Chan, J. Cong, J. R. Shinnerl, K. Sze, and M. Xie, "mPL6: Enhanced multilevel mixed-size placement," in *Proc. Int. Symp. Phys. Design (ISPD)*, Monterey, CA, USA, 2006, pp. 212–214.

[4] H.-C. Chen, Y.-L. Chunag, Y.-W. Chang, and Y.-C. Chang, "Constraint graph-based macro placement for modern mixed-size circuit designs," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, San Jose, CA, USA, 2008, pp. 218–223.

[5] T.-C. Chen, Z.-W. Jiang, T.-C. Hsu, H.-C. Chen, and Y.-W. Chang, "NTUplace3: An analytical placer for large-scale mixed-size designs with preplaced blocks and density constraint," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 7, pp. 1228–1240, Jul. 2008.

[6] T.-C. Chen, P.-H. Yuh, Y.-W. Chang, F.-J. Huang, and D. Liu, "MP-Trees: A packing-based macro placement algorithm for modern mixed-size designs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 9, pp. 1621–1634, Sep. 2008.

[7] H. Eisenmann and F. M. Johannes, "Generic global placement and floorplanning," in *Proc. Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 1998, pp. 269–274.

[8] (Mar. 30, 2014). *ePlace-MS Homepage*. [Online]. Available: http://vlsi-cuda.ucsd.edu/~ljw/ePlace-MS/

[9] M. R. Garey, D. S. Johnson, and L. Stockmeyer, "Some simplified NP-complete graph problems," *Theor. Comput. Sci.*, vol. 1, no. 3, pp. 237–267, 1976.

[10] S. K. Han, K. Jeong, A. B. Kahng, and J. Lu, "Stability and scalability in global routing," in *Proc. 13th Int. Workshop Syst. Level Interconnect Prediction (SLIP)*, San Diego, CA, USA, 2011, pp. 1–6.

[11] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J. Res. Nat. Bur. Stand.*, vol. 49, no. 6, pp. 409–436, 1952.

[12] M.-K. Hsu, V. Balabanov, and Y.-W. Chang, "TSV-aware analytical placement for 3D IC designs based on a novel weighted-average wirelength model," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 4, pp. 497–509, Apr. 2013.

[13] M.-K. Hsu and Y.-W. Chang, "Unified analytical global placement for large-scale mixed-size circuit designs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 9, pp. 1366–1378, Sep. 2012.

[14] (2012). *ITRS*. [Online]. Available: http://www.itrs.net/Links/2012ITRS/Home2012.htm

[15] A. B. Kahng and Q. Wang, "A faster implementation of APlace," in *Proc. Int. Symp. Phys. Design (ISPD)*, Monterey, CA, USA, 2006, pp. 218–220.

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on June 12,2025 at 02:50:26 UTC from IEEE Xplore. Restrictions apply.

[16] M.-C. Kim, D.-J. Lee, and I. L. Markov, "SimPL: An effective placement algorithm," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 1, pp. 50–60, Jan. 2012.

[17] M.-C. Kim and I. Markov, "ComPLx: A competitive primal-dual Lagrange optimization for global placement," in *Proc. 49th Annu. Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 2012, pp. 747–752.

[18] M.-C. Kim, N. Viswanathan, C. J. Alpert, I. L. Markov, and S. Ramji, "MAPLE: Multilevel adaptive placement for mixed-size designs," in *Proc. Int. Symp. Phys. Design (ISPD)*, Monterey, CA, USA, 2012, pp. 193–200.

[19] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[20] J. Kleinhans, G. Sigl, F. Johannes, and K. Antreich, "GORDIAN: VLSI placement by quadratic programming and slicing optimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 10, no. 3, pp. 356–365, Mar. 1991.

[21] T. Lin, C. Chu, J. R. Shinnerl, I. Bustany, and I. Nedelchev, "POLAR: Placement based on novel rough legalization and refinement," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, San Jose, CA, USA, 2013, pp. 357–362.

[22] J. Lu, "Fundamental research on electronic design automation in VLSI design: Routability," M.Phil. thesis, Dept. Electron. Inf. Eng., Hong Kong Polytechnic Univ., Hong Kong, 2010.

[23] J. Lu *et al.*, "ePlace: Electrostatics based placement using fast Fourier transform and Nesterov's method," *ACM Trans. Design Autom. Electron. Syst.*, vol. 20, no. 2, Feb. 2015, Art. ID 17.

[24] J. Lu *et al.*, "FFTPL: An analytic placement algorithm using fast Fourier transform for density equalization," in *Proc. IEEE Prof. Int. Conf. ASIC (ASICON)*, Shenzhen, China, 2013, pp. 1–4.

[25] J. Lu *et al.*, "ePlace: Electrostatics based placement using Nesterov's method," in *Proc. 51st Annu. Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 2014, pp. 1–6.

[26] J. Lu, W.-K. Chow, and C.-W. Sham, "Clock network synthesis with concurrent gate insertion," in *Proc. 20th Int. Workshop Power Timing Model. Optim. Simulat. (PATMOS)*, Grenoble, France, 2010, pp. 228–237.

[27] J. Lu, W.-K. Chow, and C.-W. Sham, "A new clock network synthesizer for modern VLSI designs," *Integration*, vol. 45, no. 2, pp. 121–131, 2012.

[28] J. Lu, W.-K. Chow, and C.-W. Sham, "Fast power- and slew-aware gated clock tree synthesis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 11, pp. 2094–2103, Nov. 2012.

[29] J. Lu, W.-K. Chow, C.-W. Sham, and E. F.-Y. Young, "A dual-MST approach for clock network synthesis," in *Proc. 15th Asia South Pac. Design Autom. Conf. (ASPDAC)*, Taipei, Taiwan, 2010, pp. 467–473.

[30] J. Lu and C.-W. Sham, "LMgr: A low-memory global router with dynamic topology update and bending-aware optimum path search," in *Proc. 14th Int. Symp. Qual. Electron. Design (ISQED)*, Santa Clara, CA, USA, 2013, pp. 231–238.

[31] I. L. Markov, J. Hu, and M.-C. Kim, "Progress and challenges in VLSI placement research," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (DAC)*, San Jose, CA, USA, 2012, pp. 275–282.

[32] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "VLSI module placement based on rectangle-packing by the sequence pair," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 15, no. 12, pp. 1518–1524, Dec. 1996.

[33] G.-J. Nam, "ISPD 2006 placement contest: Benchmark suite and results," in *Proc. Int. Symp. Phys. Design (ISPD)*, Monterey, CA, USA, 2006, p. 167.

[34] G.-J. Nam, C. J. Alpert, P. Villarrubia, B. Winter, and M. Yildiz, "The ISPD 2005 placement contest and benchmark suite," in *Proc. Int. Symp. Phys. Design*, San Francisco, CA, USA, 2005, pp. 216–220.

[35] W. C. Naylor, R. Donelly, and L. Sha, "Non-linear optimization system and method for wire length and delay optimization for an automatic electric circuit placer," U.S. Patent 6 301 693, 2001.

[36] A. S. Nemirovskii and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*. Chichester, U.K.: Wiley, 1983.

[37] Y. E. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," *Sov. Math.*, vol. 27, no. 2, pp. 372–376, 1983.

[38] T. Ooura. (2001). *General Purpose FFT Package*. [Online]. Available: http://www.kurims.kyoto-u.ac.jp/~ooura/fft.html

[39] J. A. Roy, S. N. Adya, D. A. Papa, and I. L. Markov, "Min-cut floorplacement," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 7, pp. 1313–1326, Jul. 2006.

[40] C. Sechen and A. Sangiovanni-Vincentelli, "TimberWolf3.2: A new standard cell placement and global routing package," in *Proc. Conf. Design Autom. (DAC)*, Las Vegas, NV, USA, 1986, pp. 432–439.

[41] C.-W. Sham, E. F.-Y. Young, and J. Lu, "Congestion prediction in early stages of physical design," *ACM Trans. Design Autom. Electron. Syst.*, vol. 14, no. 1, pp. 1–18, Jan. 2009.

[42] J. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Dept. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. TR-94-125, 1994.

[43] G. Skollermo, "A Fourier method for the numerical solution of Poisson's equation," *Math. Comput.*, vol. 29, no. 131, pp. 697–711, 1975.

[44] P. Spindler, U. Schlichtmann, and F. M. Johannes, "Kraftwerk2—A fast force-directed quadratic placement approach using an accurate net model," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 8, pp. 1398–1411, Aug. 2008.

[45] N. Viswanathan *et al.*, "RQL: Global placement via relaxed quadratic spreading and linearization," in *Proc. 44th ACM/IEEE Design Autom. Conf. (DAC)*, San Diego, CA, USA, 2007, pp. 453–458.

[46] N. Viswanathan, M. Pan, and C. Chu, "FastPlace3.0: A fast multilevel quadratic placement algorithm with placement congestion control," in *Proc. Asia South Pac. Design Autom. Conf. (ASPDAC)*, Yokohama, Japan, 2007, pp. 135–140.

[47] J. Z. Yan, N. Viswanathan, and C. Chu, "Handling complexities in modern large-scale mixed-size placement," in *Proc. 46th Annu. Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 2009, pp. 436–441.

[48] H. Zhuang, J. Lu, K. Samadi, Y. Du, and C.-K. Cheng, "Performance-driven placement for design of rotation and right arithmetic shifters in monolithic 3D ICs," in *Proc. Int. Conf. Commun. Circuits Syst. (ICCCAS)*, Chengdu, China, 2013, pp. 509–513.

Jingwei Lu (S'11–M'14) received the B.S. degree in information engineering from Zhejiang University, Hangzhou, China, the M.Phil. degree in computer engineering from Hong Kong Polytechnic University, Hong Kong, and the Ph.D. degree in computer science (computer engineering) from the University of California, San Diego, La Jolla, CA, USA, in 2014.

He joined Cadence Design Systems, Inc., San Jose, CA, USA, in 2014, as a Lead Software Engineer. His current research interests include analytic optimization of placement and other physical design automation.

Dr. Lu was the recipient of the Jacobs Fellowship from the University of California, San Diego, in 2010–2013, the Best Paper Award from the IEEE International Symposium on Quality Electronic Design in 2013, and the Best Paper Award nominations from ACM/IEEE Design Automation Conference in 2014 and ACM/IEEE Asia and South Pacific Design Automation Conference in 2010. He has been a Technical Reviewer for the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, the IEEE Transactions on Very Large Scale Integration, and the *Integration, the VLSI Journal* since 2014.

Authors' photographs and biographies not available at the time of publication.

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

He was a Senior CAD Engineer at Advanced Micro Devices Inc., Sunnyvale, CA, USA, from 1984 to 1986. He joined the University of California, San Diego (UCSD), La Jolla, CA, USA, in 1986, where he is a Distinguished Professor with the Department of Computer Science and Engineering and an Adjunct Professor with the Department of Electrical and Computer Engineering. He served as a Chief Scientist at Mentor Graphics, Wilsonville, OR, USA, in 1999. He was an Honorary Guest Professor with Tsinghua University, Beijing, China, from 2002 to 2008 and a Visiting Professor with the National Taiwan University, in 2011. His current research interests include interconnect modeling and analysis, network optimization, and design automation on microelectronic circuits.

Dr. Cheng was the recipient of the Best Paper Awards from the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems in 1997 and 2002, the NCR Excellence in Teaching Award, School of Engineering, UCSD, in 1991, the IBM Faculty Awards in 2004, 2006, and 2007, and the Distinguished Faculty Certificate of Achievement, UJIMA Network, UCSD, in 2013. He was an Associate Editor of the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems from 1994 to 2003.

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on June 12,2025 at 02:50:26 UTC from IEEE Xplore. Restrictions apply.