Pplace-MS: Methodologically Faster Poisson's Equation-Based Mixed-Size Global Placement

Keyu Peng and Wenxing Zhu

***Abstract*—With the advancement of semiconductor technologies, the acceleration of advanced EDA algorithms is receiving much attention. However, developing a faster mixed-size placer without hardware acceleration and loss of solution quality is of great challenge. In this article, we propose a novel definition of potential energy for each block for global placement based on an analytical solution of Poisson's equation. A fast approximate computation scheme for partial derivatives of the potential energy is given with considerably less computational loads than existing electrostatics-based placers. Moreover, we propose an effective and efficient occupy-aware macro legalization algorithm. Then, a mixed-size placer named Pplace-MS is developed. Compared to the existing leading mixed-size placer, Pplace-MS on average achieves 2.054× speedup in single-threaded mode on the same machine and 2.3% reduction of scaled half-perimeter wirelength on the modern mixed-size placement benchmarks. The proposed approach can also be considered accelerated on GPU, as previous works.**

***Index Terms*—Global placement, macro legalization (mLG), mixed-size design, Poisson's equation.**

I. Introduction

PLACEMENT is a most critical but time-consuming step in VLSI physical design. Modern mixed-size (MMS) designs may contain millions of standard cells and thousands of large macro blocks to meet various design requirements. However, the big topological differences between macros and standard cells pose a great challenge to the mixed-size placement problem. There have been advancements in a faster solution to the problem, such as GPU acceleration [17], [18]. However, more efficient and effective mixed-size placement algorithms in methodology are still of great interest, since a new approach can also be considered accelerated on GPU.

According to [12], existing mixed-size global placement (mGP) algorithms can be classified into three main types: 1) constructive methods; 2) two-stage methods; and 3) one-stage methods. Constructive methods constructively determine the positions of macros, such as [2] and [27]. Capo [2] iteratively invokes fixed-outline floorplanning to guide macros placement during the min-cut placement process. FLOP [27] clusters standard cells into soft macros, and a min-cut-based fixed-outline floorplanner is applied to derive the initial positions of macros. Then, an incremental placer is used to obtain the final placement result. Since partitioning and clustering methods are used in constructive methods, the solution quality is limited due to the reduction of the solution space.

Two-stage methods first determine the positions of macros and proceed with the placement of standard cells without overlapping with other blocks, such as [5] and [7]. CG [5] determines the locations of macros by linear programming based on the transitive closure graph floorplan representation. MP-tree [7] presents a multipacking-tree (MP-tree) representation for macro placement and places macros along the chip boundary. In two-stage methods, when the positions of macros have been determined, the optimization space of standard cells may be significantly reduced, which may lead to a suboptimal final placement solution.

One-stage methods determine the positions of standard cells and macros simultaneously [3], [6], [12], [13], [15], [20], [25]. ComPLx [15] uses the *macro shredding* technique to divide macros into equal-sized shred cells without net connections, and each macro will be rebuilt by the shred cells to which it belongs, at the end of the placement. APlace [13] and NTUplace3 [6] use different bell-shaped potential functions for macros and standard cells to distribute blocks evenly. FastPlace 2.0 [25] constructs unequal bin structure according to the sizes of macros and standard cells, and then moves each block by the *Cell Shifting* technique. mPL6 [3] is a multilevel-placement engine that approximates the placement density by solving the Helmholtz equation and thus smoothing the density constraint. ePlace-MS [20] treats macros and standard cells in almost the same way. It models all blocks as charged particles in the placement region, where the charge of each particle is its area. Subsequently, ePlace-MS distributes all blocks uniformly through the electric force of the electrostatic system. One-stage methods take into account the information about net connections and sizes of both standard cells and macros, which may allow standard cells connected to a macro to be closer to each other and provide better placement results. In particular, this approach has been proven to be the most effective for mixed-size placement [6], [8], [20].

For one-stage placers, the analytical approach has shown excellent performance for large-scale mGP problem. The analytical approach minimizes an analytical cost function through numerical analysis techniques, which can be classified into two types by the difference of cost function: 1) nonlinear and 2) quadratic. Quadratic placers, such as Kraftwerk2 [24], FastPlace3 [26], and ComPLx [15], express the wirelength in a convex quadratic cost function. Nonlinear placers, such as APlace3 [14], mPL6 [3], NTUplace3 [6], ePlace-MS [20], and

Manuscript received 27 February 2023; revised 1 July 2023 and 12 September 2023; accepted 24 September 2023. Date of publication 28 September 2023; date of current version 22 January 2024. This work was supported by the National Natural Science Foundation of China under Grant 62174033. This article was recommended by Associate Editor I. H.-R. Jiang. *(Corresponding author: Wenxing Zhu.)*

The authors are with the Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, Fuzhou 350108, China (e-mail: wxzhu@fzu.edu.cn).

Digital Object Identifier 10.1109/TCAD.2023.3320628

RePlAce [8], smooth the half-perimeter wirelength (HPWL) through continuously differentiable nonlinear cost functions, e.g., log–sum–exp (LSE) function [23] and weighted-average (WA) function [11]. Since the quadratic cost function can only handle two-pin connections, multipin nets are often modeled by the clique net model or the star net model. Nonlinear cost functions are able to better model both two-pin and multipin nets wirelength, hence nonlinear mixed-size placers have demonstrated better solution quality.

ePlace [19] is a state-of-the-art analytical placer, which develops an electrostatics-based density function *eDensity* to model the density distribution of blocks. It uses Poisson's equation to model the electric potential and field distribution through *eDensity*, thus spreading blocks by the electric force from the electrostatic system. Based on ePlace, ePlace-MS [20] is a high performance one-stage mixed-size placer, in which the *eDensities* of standard cells and macros are modeled and handled in exactly the same way. Since a larger macro is applied with a larger density force, which will cause the placement oscillation, ePlace-MS proposes a preconditioner to avoid this issue. In contrast, RePlAce [8] adds the *local density function* to the objective function that comprehends density overflow per bin and obtains the best wirelength as known on the MMS placement benchmarks.

After optimizing the positions of both standard cells and macros, one-stage methods produce a solution with a small amount of overlaps between macros. Eliminating these overlaps is crucial for the final solution quality, especially for the circuits containing millions of standard cells and thousands of macros. ComPLx [15] uses extra multiple global placement iterations to gradually decrease these overlaps. FastPlace 2.0 [25] adopts simulated annealing (SA) to perturb sequence pairs constructed from macros position relationships, to eliminate overlaps between macros.

In mPL6 [3], similar to floorplanning that two directed acyclic graphs are constructed based on the horizontal and vertical position relationships of macros, overlaps between macros are eliminated by removing the edges of these two graphs to change the position relationships of macros. ePlace-MS [20] and RePlAce [8] use SA-based methods to determine the positions of macros, which randomly perturb solutions and may accept bad solutions. To reduce overlaps, the placers will continuously increase the random movements of macros, placing a greater burden on the following optimization process and leading to the loss of solution quality.

Electrostatic-based placement algorithms have been extended to placement problems considering various complex constraints, such as routability [8] and fence region constraints [10], and have shown excellent performance. Recently, it has been of great interest to accelerate state-of-the-art EDA algorithms, such as DREAMPlace [17] and Xplace [18]. However, developing a much faster mixed-size placer without hardware acceleration and without loss of solution quality is still an intrinsic challenge.

To address this challenge, motivated by Poisson's equation-based floorplanning work [16], this article proposes a novel definition of potential energy for global placement based on the analytical solution of Poisson's equation in [28] and develops an effective and efficient mixed-size placement algorithm, which is named Pplace-MS. The main contributions are summarized as follows.

1) Based on *Riemann's integral theorem*, a novel formulation of the potential energy function for each block is presented. Moreover, a fast computation scheme for the partial derivatives of potential energy is developed. The proposed approach can significantly improve the speed of electrostatics-based placers.
2) A novel method for calculating the density force based on Poisson's equation is given. The density force improves the overlap elimination efficiency by 20.15% and improves sHPWL compared to the density force of the previous electrostatics-based placers.
3) A weight value is presented to characterize the topological differences between macros and guide the addition of filler cells and the calculation of bin density.
4) An occupy-aware macro legalization (mLG) algorithm is proposed to achieve mLG after mGP, which can eliminate overlaps between macros accurately and effectively.
5) Experimental results on the MMS benchmarks demonstrate that our mixed-size placer achieves 105.4% runtime improvement and 2.3% sHPWL reduction on average over the latest state-of-the-art work [8] in single-threaded mode.

The remainder of this article is organized as follows. Section II gives the problem statement and some preliminaries. Section III presents our core placement algorithm. Experimental results and comparisons are provided in Section IV. Finally, Section V concludes this work.

II. Preliminaries

In this section, we introduce the mixed-size placement problem and the background of Poisson's equation for VLSI global placement.

A VLSI circuit can be modeled as a hypergraph $H(V, E)$, where V is the set of blocks and E is the set of hyperedges. Let V_s and V_m be the sets of standard cells and movable macros, respectively. Moreover, let $n = |V_s \cup V_m|$, let n_m be the number of macros, and let (w_i, h_i) be the width and height of block v_i. Given a placement region $\mathbf{R} = [0, W] \times [0, H]$, it can be split into the set B of uniform bins, $|B| = K \times K$, where the specific setting of K can be found in Section III-C. Then, the width and height of bins are $w_b = (W/K)$ and $h_b = (H/K)$, respectively. Furthermore, let (x_i, y_i) represent the coordinate of the center of block v_i, and $\mathbf{v} = (\mathbf{x}, \mathbf{y}) = (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n)$ denote the coordinates of all blocks.

The VLSI placement problem is to determine the positions of movable blocks, including standard cells and movable macros, so that 1) there is no overlap between blocks; 2) each block is placed at the placement site on row; and 3) each block is placed using enough free sites while minimizing the total wirelength

$$\begin{aligned} \min_{\mathbf{v}} \quad & WL(\mathbf{v}) \\ \text{s.t.} \quad & \text{no overlap between blocks} \\ & \text{each block is placed at placement site on row} \\ & \text{each block using enough free sites} \end{aligned} \tag{1}$$

where the wirelength $WL(\mathbf{v})$ can be calculated by the HPWL [4].

To address the placement problem, analytical placers consist of three stages: 1) global placement; 2) legalization; and 3) detailed placement (DP). Global placement evenly distributes movable blocks. Then, legalization removes all overlaps between blocks and places standard cells row by row. Further, DP refines the solution. In global placement, analytical placers transform the nonoverlapping constraint in model (1) into density constraint and consider the following global placement problem:

$$\begin{aligned} \min_{\mathbf{v}} \quad & WL(\mathbf{v}) \\ \text{s.t.} \quad & \rho_b(\mathbf{v}) \le \rho_m \quad \forall b \in B \end{aligned} \tag{2}$$

where $\rho_b(\mathbf{v})$ is the density of bin b, and ρ_m is the target distribution density of bin b.

Poisson's equation is adopted in leading academic placers *ePlace-series* [8], [19], [20] for block spreading in VLSI global placement. ePlace [19] models the placement of a circuit as an electrostatic system, in which each block v_i is regarded as a positive charge and its electric quantity q_i is set to its area A_i. The density of blocks $\rho(x, y)$ on $\mathbf{R}$ is considered as the electric density. According to the electric potential of the system, the electric potential $\psi(x, y)$ is obtained by the solution of Poisson's equation

$$\begin{cases} \nabla \cdot \nabla \psi(x, y) = -\rho(x, y), (x, y) \in \mathbf{R} & \text{(3a)} \\ \hat{n} \cdot \nabla \psi(x, y) = 0, (x, y) \in \partial \mathbf{R} & \text{(3b)} \\ \iint_{\mathbf{R}} \psi(x, y)\, \mathrm{d}x\mathrm{d}y = \iint_{\mathbf{R}} \rho(x, y)\mathrm{d}x\mathrm{d}y = 0 & \text{(3c)} \end{cases}$$

where $\partial \boldsymbol{R}$ and $\hat{\boldsymbol{n}}$ represent the boundary of $\boldsymbol{R}$ and the outer normal vector of the boundary, respectively. Equation (3a) is Poisson's equation, (3b) is the Newmann boundary condition, and (3c) is the compatibility condition to ensure the uniqueness of the solution of Poisson's equation.

By defining the density function for each block $v_i \in V$ as

$$\rho_i(x, y) = \begin{cases} 1, \text{ if } (x, y) \in R_i \\ 0, \text{ else} \end{cases}$$

where R_i is the region of block v_i, the density function of the system is defined in Pplace [28] as

$$\rho(x, y) = \sum_{v_i \in V} \rho_i(x, y), \quad (x, y) \in \boldsymbol{R}. \tag{4}$$

Then essentially, by changing the compatibility condition (3c) to

$$\iint_{\mathbf{R}} \psi(x, y)\mathrm{d}x\mathrm{d}y = \iint_{\mathbf{R}} \rho(x, y)\mathrm{d}x\mathrm{d}y$$

Poisson's equation is solved analytically in [28] to obtain the corresponding electric density $\psi(x, y)$ for VLSI global placement.

Furthermore, Pplace [28] has the following discrete form for the electric potential on bin $b_{l,j}$:

$$\hat{\psi}(l, j) \approx \sum_{u=0}^{K} \sum_{p=0}^{K} a_{u,p} \cos \frac{u\left(l + \frac{1}{2}\right)\pi}{m} \cos \frac{p\left(j + \frac{1}{2}\right)\pi}{m} \tag{5}$$

where $a_{u,p}$ is the coefficients given in [28, eq. (27)].

Poisson's equation approach in [28] has been adopted for large scale fixed-outline floorplanning [16], in which a new block nonoverlapping condition is given by the following theorem.

Theorem 1 [16]: There is no overlap between all blocks $v_i \in V$ if and only if

$$\iint_{R_i} \sum_{v_j \in V} \rho_j(x, y)\mathrm{d}x\mathrm{d}y = \mathrm{Area}(R_i). \tag{6}$$

Then based on Theorem 1, a novel method was developed for fixed-outline floorplanning, and it was experimentally shown to have excellent performance. In this article, we extend Theorem 1 for mGP.

III. Proposed Algorithm

By Theorem 1, the nonoverlapping constraint in model (1) is equivalent to (6). Hence, similar to PeF [16], by the regularization technique, the mGP problem can be transformed to

$$\min_{\mathbf{v}} WL(\mathbf{v}) + \lambda \sum_{v_i \in V} \iint_{R_i} \sum_{v_j \in V} \rho_j(x, y)\mathrm{d}x\mathrm{d}y. \tag{7}$$

For relaxing the density function to facilitate optimization, the density function $\sum_{v_j \in V} \rho_j(u, v)$ in (7) can be smoothed using Poisson's equation.

Then similar to PeF [16], the potential energy of block v_i can be calculated by integrating the potential over its region R_i

$$\mathcal{D}_i = \iint_{R_i} \psi(x, y)\mathrm{d}x\mathrm{d}y$$

and the potential energy of the electrostatic system is defined as

$$\mathcal{D}(\mathbf{v}) = \sum_{v_i \in V} \mathcal{D}_i(x_i, y_i).$$

The electrostatic system reaches an electrostatic equilibrium state when the total potential energy is reduced to zero. Further, a necessary condition for eliminating overlaps can be obtained, i.e., the system potential energy $\mathcal{D}(\mathbf{v})$ is the smallest. This allows us to use the regularization function $\mathcal{D}(\mathbf{v})$ to transform the model (7) into

$$\min_{\mathbf{v}} f(\mathbf{v}) = \hat{WL}(\mathbf{v}) + \lambda \mathcal{D}(\mathbf{v}) \tag{8}$$

where $\hat{WL}(\mathbf{v})$ is a smoothed function that approximate the HPWL wirelength and λ is a regularization factor. In this article, the LSE wirelength function [23] is adopted for $\hat{WL}(\mathbf{v})$.

For problem (2), the main objective of the previous mixed-size placers, such as ePlace-MS [20] and RePlAce [8], is to make the density of bins evenly. Thus, ePlace-MS [20] and RePlAce [8] move blocks through the electric field of bins to reduce the density of bins. In this article, problem (7) aims to eliminate overlaps by reducing the density within the region R_i of each block v_i.

In this section, first, we derive the formula of the potential energy for each block, and the approximation of the potential-driven density force. Then, we analyze the density forces in terms of standard cells and macros. After that, a mGP algorithm is developed. Moreover, after the mGP stage, an occupy-aware mLG algorithm is designed to eliminate the overlaps between macros.

A. Potential Energy and Potential-Driven Density Force

Assume that for each block $v_i \in V$, $x_i \in [(w_i/2), W-(w_i/2)]$ and $y_i \in [(h_i/2), H-(h_i/2)]$, we set the following parameters:

$$P_i^L = \left\lfloor \frac{x_i - \frac{w_i}{2}}{w_b} \right\rfloor, \quad P_i^R = \left\lceil \frac{x_i + \frac{w_i}{2}}{w_b} \right\rceil$$
$$Q_i^B = \left\lfloor \frac{y_i - \frac{h_i}{2}}{h_b} \right\rfloor, \quad Q_i^U = \left\lceil \frac{y_i + \frac{h_i}{2}}{h_b} \right\rceil. \tag{9}$$

Based on *Riemann's integral theorem*, the potential energy for each block $v_i \in V$ can be approximated on bin grid by

$$\mathcal{D}_i(x_i, y_i) = \int_{y_i-\frac{h_i}{2}}^{y_i+\frac{h_i}{2}} \mathrm{d}y \int_{x_i-\frac{w_i}{2}}^{x_i+\frac{w_i}{2}} \psi(x, y)\mathrm{d}x \approx \sum_{p=P_i^L}^{P_i^R} \sum_{q=Q_i^B}^{Q_i^U} d_x\big(b_{p,q}, v_i\big) \cdot d_y\big(b_{p,q}, v_i\big) \cdot \hat{\psi}(p, q) \tag{10}$$

where $d_x(b_{p,q}, v_i)$ and $d_y(b_{p,q}, v_i)$ are the overlap functions between block v_i and bin $b_{p,q}$ in the horizontal and vertical directions, respectively. If $w_i > w_b$, we define $d_x(b, v_i)$ as

$$d_x(b, v_i) = \begin{cases} x_i - x_b + \frac{w_b+w_i}{2} : x_i \in \left[x_b - \frac{w_i+w_b}{2}, x_b - \frac{w_i-w_b}{2}\right] \\ w_b : x_i \in \left(x_b - \frac{w_i-w_b}{2}, x_b + \frac{w_i-w_b}{2}\right) \\ -x_i + x_b + \frac{w_b+w_i}{2} : x_i \in \left[x_b + \frac{w_i-w_b}{2}, x_b + \frac{w_i+w_b}{2}\right] \\ 0 : \text{else} \end{cases} \tag{11}$$

where x_b is the x-coordinate of the center of bin b. Moreover, if $h_i > h_b$, the overlap function $d_y(b, v_i)$ in the vertical direction can be defined similarly.

Then, the density force vector for each movable block v_i is set aligning with the steepest descent direction of the potential energy of v_i

$$F_i = -\left(\frac{\partial \mathcal{D}}{\partial x_i}, \frac{\partial \mathcal{D}}{\partial y_i}\right) = -\left(\frac{\partial \mathcal{D}_i}{\partial x_i}, \frac{\partial \mathcal{D}_i}{\partial y_i}\right). \tag{12}$$

The density force moves block v_i from a position with high electric potential to a position with low electric potential, thereby reducing its potential energy, and eventually, movable blocks can be evenly distributed over the placement region.

Specifically, the partial derivatives of potential energy $\mathcal{D}_i$ to x_i of block $v_i \in V$ can be approximated on bin grid by

$$\begin{aligned} \frac{\partial \mathcal{D}_i}{\partial x_i} &\approx \sum_{p=P_i^L}^{P_i^R} \sum_{q=Q_i^B}^{Q_i^U} \frac{\partial d_x\big(b_{p,q}, x_i\big)}{\partial x_i} \cdot d_y\big(b_{p,q}, v_i\big) \cdot \hat{\psi}(p, q) \\ &= \sum_{q=Q_i^B}^{Q_i^U} \frac{\partial d_x\Big(b_{P_i^L,q}, v_i\Big)}{\partial x_i} \cdot d_y\Big(b_{P_i^L,q}, v_i\Big) \cdot \hat{\psi}\big(P_i^L, q\big) \\ &\quad + \sum_{q=Q_i^B}^{Q_i^U} \frac{\partial d_x\Big(b_{P_i^R,q}, v_i\Big)}{\partial x_i} \cdot d_y\Big(b_{P_i^R,q}, v_i\Big) \cdot \hat{\psi}\big(P_i^R, q\big) \\ &\triangleq \sum_{q=Q_i^B}^{Q_i^U} d_y\Big(b_{P_i^R,q}, v_i\Big) \cdot \hat{d}_x^i \cdot \hat{\psi}\big(P_i^R, q\big) \\ &\quad - \sum_{q=Q_i^B}^{Q_i^U} d_y\Big(b_{P_i^L,q}, v_i\Big) \cdot \hat{d}_x^i \cdot \hat{\psi}\big(P_i^L, q\big) \\ &= \sum_{q=Q_i^B}^{Q_i^U} d_y\Big(b_{P_i^L,q}, v_i\Big) \cdot \hat{d}_x^i \Big(\hat{\psi}\big(P_i^R, q\big) - \hat{\psi}\big(P_i^L, q\big)\Big) \end{aligned} \tag{13}$$

where if $w_i > w_b$, then we have $\hat{d}_x^i = 1$.

In (13), for bin $b_{P_i^L,q}$, since $x_i \in [x_{b_{P_i^L,q}} + (w_i - w_b/2), x_{b_{P_i^L,q}} + (w_i + w_b/2)]$, we have

$$d_x\Big(b_{P_i^L,q}, v_i\Big) = -x_i + x_{b_{P_i^L,q}} + \frac{w_b + w_i}{2}.$$

Hence, we get $([\partial d_x(b_{P_i^L,q}, v_i)]/\partial x_i) = -1$. Similarly, for bin $b_{P_i^R,q}$, since $x_i \in [x_{b_{P_i^R,q}} - ([w_i + w_b]/2), x_{b_{P_i^R,q}} - (w_i - w_b/2]$, we have

$$d_x\Big(b_{P_i^R,q}, v_i\Big) = x_i - x_{b_{P_i^R,q}} + \frac{w_b + w_i}{2}$$

and $([\partial d_x(b_{P_i^R,q}, v_i)]/\partial x_i) = 1$. And for any bin $b_{p,q}$, $p \in [P_i^L + 1, P_i^R - 1]$, we can get $d_x(b_{p,q}, v_i) = w_b$ and $([\partial d_x(b_{p,q}, v_i)]/\partial x_i) = 0$.

The last equality in (13) is due to the fact that, each block $v_i \in V$ is a rectangular module, and it holds obviously that $d_y(b_{P_i^R,q}, v_i) = d_y(b_{P_i^L,q}, v_i)$.

Intuitively, (13) means that $(\partial \mathcal{D}_i/\partial x_i)$ for each block is determined by the potential difference of boundary bins covered by the block. Similarly, we can get the partial derivatives $(\partial \mathcal{D}_i/\partial y_i)$ of potential energy.

The approximated density force in (13) can move macro and standard cells well in most cases. However, if a standard cell v_i is located only in one bin in the horizontal and vertical directions, then according to (13), it will result in $(\partial \mathcal{D}_i/\partial x_i) \approx 0$ and $(\partial \mathcal{D}_i/\partial y_i) \approx 0$, respectively, due to the fact that $P_i^L = P_i^R$ and $Q_i^B = Q_i^U$.

To handle this issue, we propose the following technique for small blocks. Suppose the width w_i of block v_i satisfy $w_i \le w_b$, we give an inflation factor $\tau > 1.0$ and define the virtual width $\hat{w}_i$ of block v_i by $\hat{w}_i = \tau \cdot w_b$. In this article, we set $\tau = 1.30$ to obtain a better tradeoff between the quality of the solution and the speed of convergence of our mixed-size placement algorithm. Thus, for the calculation of density $\rho_b(\mathbf{v})$ and potential energy $\mathcal{D}(\mathbf{v})$, we consider $\hat{w}_i$ as the width of block v_i. According to (9), replacing w_i in it with $\hat{w}_i$, we get new parameter values P_i^L, P_i^R. Then, if $w_i \le w_b$, we define the horizontal overlap function of block v_i and bin b as

$$d_x(b, v_i) = \begin{cases} \frac{w_i}{\tau w_b}\left(x_i - x_b + \frac{w_b(1+\tau)}{2}\right) : x_i \in \left[x_b - \frac{w_b(1+\tau)}{2}, x_b + \frac{w_b(1-\tau)}{2}\right] \\ \frac{w_i}{\tau} : x_i \in \left(x_b + \frac{w_b(1-\tau)}{2}, x_b - \frac{w_b(1-\tau)}{2}\right) \\ \frac{w_i}{\tau w_b}\left(-x_i + x_b + \frac{w_b(1+\tau)}{2}\right) : x_i \in \left[x_b - \frac{w_b(1-\tau)}{2}, x_b + \frac{w_b(1+\tau)}{2}\right] \\ 0 = \ : \text{else} \end{cases} \tag{14}$$

and have $\hat{d}_x^i = (w_i/\tau w_b)$ in (13) to ensure that the total overlap of v_i with all bins is equal to its area. Similarly, $d_y(b, v_i)$ and $\hat{d}_y^i$ for the y-direction can be obtained trivially.

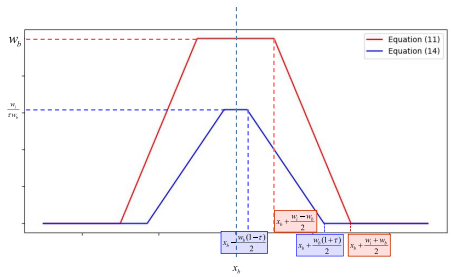

Fig. 1. Curves of (11) and (14).

Fig. 1 gives the overlap function of block v_i with bin b, where the horizontal axis represents the coordinate of block v_i in the x-direction and the vertical axis represents the overlap value. x_b is the x-coordinate of the center of bin b. If $w_i > w_b$, then the overlap function [i.e., (11)] is shown in the orange curve. If $w_i < w_b$, then the overlap function [i.e., (14)] is shown in the blue curve.

Thus, based on (11) and (14), the approximated partial derivative of the potential energy can be calculated according to (13) and they are not zeros.

B. Analysis of Density Force

At this point, we have proposed a novel method for calculating density force in our Pplace-MS, which is different from other Poisson's equation-based placement algorithms, such as ePlace [19], ePlace-MS [20], Pplace [28], and RePlAce [8]. However, when the bins are infinitely small, we find that both the spectral method in ePlace and the analytical approach in [28] converge to the solution $\psi(x, y)$ of Poisson's equation. In this case, for our Pplace-MS, the density force is in the form of the integral of ψ over R_i followed by a derivative

$$F_i = -\left(\frac{\partial \mathcal{D}_i}{\partial x_i}, \frac{\partial \mathcal{D}_i}{\partial y_i}\right) = -\left(\frac{\partial\left(\iint_{R_i} \psi(x,y)\mathrm{d}x\mathrm{d}y\right)}{\partial x_i}, \frac{\partial\left(\iint_{R_i} \psi(x,y)\mathrm{d}x\mathrm{d}y\right)}{\partial y_i}\right).$$

While, for ePlace, the density force is in the form of the derivative of ψ followed by an integral over R_i: $F_i = \iint_{R_i} \xi(x, y)\mathrm{d}x\mathrm{d}y$, where $\xi(x, y) = -([\partial\psi(x, y)]/[\partial x], [\partial\psi(x, y)]/[\partial y])$. Then, since $\psi(x, y)$ is continuously differentiable, we can have

$$-\left(\frac{\partial\left(\iint_{R_i} \psi(x,y)\mathrm{d}x\mathrm{d}y\right)}{\partial x_i}, \frac{\partial\left(\iint_{R_i} \psi(x,y)\mathrm{d}x\mathrm{d}y\right)}{\partial y_i}\right) = \iint_{R_i} \xi(x, y)\mathrm{d}x\mathrm{d}y.$$

Therefore, when the bins are infinitely small, Pplace-MS and ePlace will obtain the same density force. However, in practice, the bins cannot be infinitely small, and thus our method of calculating the density force has characteristics different from ePlace [19], ePlace-MS [20], and Pplace [28]. The two different density forces have different properties on cells and they are analyzed in detail below.

Table I compares two different density forces in Poisson's equation-based placement algorithms. In ePlace-MS [20], the density force on block v_i is determined by the distribution of electric field on the region of v_i, which is called here the electric field-driven density force and denoted by the symbol $\mathcal{F}_i$. The electric field-driven density force on block v_i is calculated by

$$\mathcal{F}_i = q_i\xi_i = \sum_{b_{p,q}\in B_{v_i}} A\left(v_i, b_{p,q}\right)\cdot \xi_{b_{p,q}} \tag{15}$$

where B_{v_i} is the set of bins overlapped by v_i, and $A(v_i, b_{p,q})$ is the overlap area between v_i and bin $b_{p,q}$. The electric field-driven density force moves each cell v_i through the direction of the electric field of each bin in R_i, thus reducing the density of bins. According to (15), in ePlace-MS, the local electric field directs each block to underfilled regions until an even density distribution of bins is achieved.

TABLE I
COMPARISON OF DENSITY FORCE BETWEEN EPLACE-MS AND OUR PPLACE-MS

	ePlace-MS [20]	Pplace-MS
Density force	$\mathcal{F}_i = q_i\xi_i$	$F_i = -(\frac{\partial \mathcal{D}_i}{\partial x_i}, \frac{\partial \mathcal{D}_i}{\partial y_i})$
Generation of density force for block v_i	Distribution of electric field on the region of each movable block v_i	Negative gradient of potential energy at each movable block v_i
Feature of density force	Local electric field directs each block to underfilled regions	Comprehends the local electric potential at each block v_i

While, the model in our Pplace-MS (8) aims to generate the density force F_i that moves each cell v_i to a lower potential region by understanding the local potential of v_i, resulting in a lower density in the region R_i. In this article, our density force on each movable block v_i is the negative gradient potential energy of v_i in (12), which is named here the potential-driven density force. The density force comprehends the local electric potential at each block v_i and moves the block in the steepest descent direction of potential energy at v_i, which efficiently eliminates the overlap between blocks.

1) Characteristics of the Density Forces for Standard Cells: The two density forces have different characteristics for the reduction of overlap area. Assume that the overlap area between two blocks accounts for the majority of their areas, which is very common in the mGP stage.

In Fig. 2, blue rectangles and gray grids represent standard cells and bins, respectively. In Fig. 2(a), red arrows depict the electric field ξ_{b_i} at bin b_i, and in Fig. 2(b), orange arrows depict the potential-driven density force F_i at each standard cell v_i. As shown in Fig. 2, standard cells v_1 and v_2 both occupy the regions of bins b_1, b_2, b_3, and b_4. Hence, in Fig. 2(a), v_1 and v_2 are affected by the electric field ξ_{b_i} of bin $b_i, i = 1, 2, 3, 4$, at the same time. The electric field-driven density force is generated by the electric fields of the regions where the cell is located, hence we can divide $\mathcal{F}_1$ and $\mathcal{F}_2$ into two distinct components: 1) the force generated by the electric field of the overlapping region ($\mathcal{F}_1^O$ and $\mathcal{F}_2^O$) and 2) the force generated by the electric field of the nonoverlapping region ($\mathcal{F}_1^N$ and $\mathcal{F}_2^N$), i.e., $\mathcal{F}_1 = \mathcal{F}_1^O + \mathcal{F}_1^N$ and $\mathcal{F}_2 = \mathcal{F}_2^O + \mathcal{F}_2^N$. According to (15), we have $\mathcal{F}_1^O = \mathcal{F}_2^O$. However, in order to eliminate the overlap between v_1 and v_2, they must be moved in different directions. Therefore, $\mathcal{F}_1^O$ and $\mathcal{F}_2^O$ do not contribute to the elimination of the overlap between v_1 and v_2. Moreover, the nonoverlapping area of v_1 and v_2, respectively,

TABLE II
NUMBER OF ITERATIONS OF THE MGP STAGE ON THE MMS BENCHMARKS [27], A = ADAPTEC, B = BIGBLUE, AND N = NEWBLUE

#Iteration	a1	a2	a3	a4	b1	b2	b3	b4	a5	n1	n2	n3	n4	n5	n6	n7	Normalized
"EFF"	587	536	558	637	656	446	859	1403	840	660	555	205	819	1177	743	749	1.2015
Pplace-MS	508	468	505	584	577	423	700	1102	698	505	457	201	759	693	581	604	1.0000

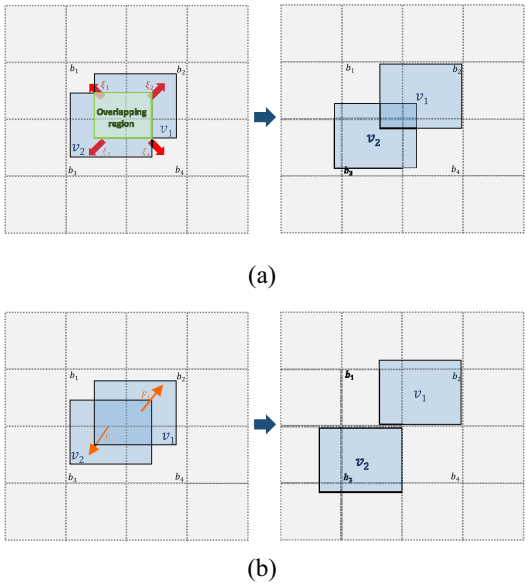

Fig. 2. Density forces for standard cells v_1 and v_2. (a) Electric field-driven density force. (b) Potential-driven density force.

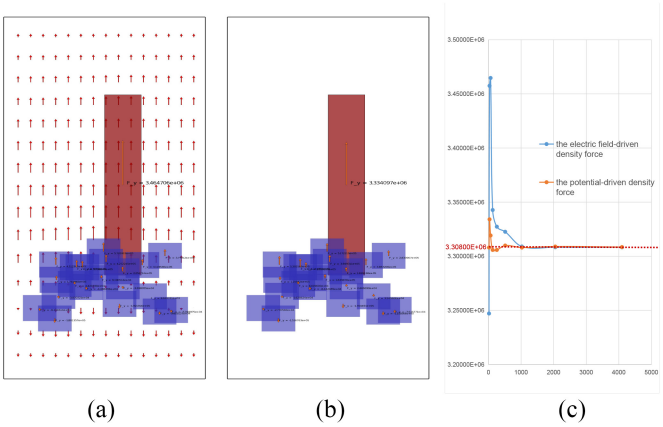

Fig. 3. Two different density forces on a macro: (a) electric field-driven density force and (b) potential-driven density force. Blue and red rectangles represent the standard cells and the macro, respectively, and red and orange arrows indicate the electric field and the density force in the y-direction, respectively. (c) Density forces on the macro with respect to the number of bins.

is small and consequently the values of density forces $\mathcal{F}_1^N$ and $\mathcal{F}_2^N$ are small due to the weight $A(v_i, b_{p,q})$ in (15). Hence, in this case, as shown in Fig. 2(a) the electric field-driven density force will require several iterations to eliminate the overlap. This phenomenon is common for standard cells in the mGP stage (see Section III-D for details), which affects the efficiency of the density forces and increases the number of iterations, henceforth waste computational resources.

In Fig. 2(b), the potential-driven density force F_i moves standard cell v_i in the steepest descent direction of its potential energy $\mathcal{D}_i$. Since the potential at the lower left region of v_1 is greater than the potential at the upper right region of v_1, the system will exert a potential-driven density force F_1 in the upper right direction on v_1 to reduce the potential energy $\mathcal{D}_i$. Conversely, a density force F_2 is exerted to v_2 in the lower left direction. As shown in Fig. 2(b), the potential-driven density force eliminates the overlap efficiently.

To compare the efficiency of the two density forces, we computed the solution by our program with the potential-driven density force replaced by the electric field-driven density force in ePlace-MS and name the mixed-size placer as "EFF." Specifically, we set "EFF" and Pplace-MS with the same initial placement (IP) and target density overflow as the termination condition for the mGP stage. Based on the number of iterations required to reduce the same density rate for both, we can compare the efficiency of the two density forces in eliminating overlap. Table II shows that our potential-driven density force is 20.15% more efficient in reducing the density overflow than the electric field-driven density force, and the experimental results (in Table V) also show that the quality of our solution is better in sHPWL.

2) Characteristics of the Density Force for Macros: For mixed-size placement, the electric field-driven density forces and the potential-driven density forces on movable macros will have a significant impact on the placement solution.

Note that, the calculations of $\psi_{b_{p,q}}$ and $\xi_{b_{p,q}}$ are subject to numerical errors due to the discretizations and truncations. Hence, the $\psi_{b_{p,q}}$ and $\xi_{b_{p,q}}$ corresponding to each bin $b_{p,q}$ are calculated with errors. However, the calculation of the electric field-driven density force of block v_i is based on the idea of integration: $F_i = \iint_{R_i} \xi(x, y)\mathrm{d}x\mathrm{d}y$. So for the density force of a macro, the huge size of the macro may amplify these errors by a factor of $w_i \times h_i$, and these errors may affect the movement of the macro. While, our density force calculation method only uses boundary bins covered by the macro for the calculation [see (13)]. Therefore, our calculation is able to reduce the amplification of the numerical errors in the x- and y-directions and may reduce the effect of the numerical errors on the movement of the macro.

Fig. 3 illustrates the two density forces in the y-direction on a macro. In Fig. 3, the 20 standard cells and one macro belong to the same net. The width and height of the macro are set to 550 and 3500, respectively. According to (15), a macro in ePlace-MS will be superimposed all the electric fields where it is located as the density force. As shown in Fig. 3(a), due to the large density overflow in the region where the standard cells are located, the electric fields are directed outwards from this region. The electric fields in the region of the macro are mostly in the upward direction, so the macro will be subject to an upward electric field-driven density force. In fact, the electric field-driven density force on the macro in Fig. 3(a) is $\mathcal{F}^y = 3.46e + 6$.

Further, as shown in Fig. 3(b), since the lower region of the macro has a higher potential than the upper region of the macro, the potential-driven density force will exert an

upward force on the macro, reducing the potential energy of the macro and eliminating the overlap. It is worth pointing out that, if there is a greater density overflow in the lower part of the macro, i.e., if there exists more standard cells overlap in this region, then this region will have a higher potential. As a result, more upward force will be applied to the macro to free up more space for other standard cells. In fact, in Fig. 3(b), the potential-driven density force on the macro is $F^y = 3.334e + 6$.

Fig. 3(c) gives the curves of the electric field-driven density force and the potential-driven density force on the macro. The horizontal axis denotes m (the number of bins is $m \times m$), and the vertical axis denotes the value of the density force on the macro in the y-direction. In addition, the blue and orange curves represent the curves of the electric field-driven density force and the potential-driven density force, respectively. Since the two different density forces will converge to the exact density force when the number of bins is large enough, we set the average of the two density forces when the number of bins is 8196×8196 to be the converged density force F^y_{conv}.

According to the curves in Fig. 3(c), $F^y_{\text{conv}} = 3.308e+6$. In addition, the value of potential-driven density force is closer to F^y_{conv} when the number of bins is small. Further, the error between the electric field-driven density force on the macro in Fig. 3(a) and F^y_{conv} is $1.567e+5$, while the error is $2.609e+4$ between the potential-driven density force on the macro in Fig. 3(b) and F^y_{conv}, which is smaller.

In order to illustrate the effect of the two density forces on global placement, we analyze it by iteration of "EFF" and Pplace-MS on the global placement of benchmark newblue1. It is worth noting that "EFF" and Pplace-MS differ only in the way density forces are calculated. Fig. 4(a)–(h) compares the performance of the electric field-driven density force [left-side figures, i.e., Fig. 4(a), (c), (e), and (g)] and the potential-driven density force [right-side figures, i.e., Fig. 4(b), (d), (f), and (h)] at the mGP stage for newblue1 [27], where Fig. 4(a) and (b) shows the placements of "EFF" and Pplace-MS, respectively, at an overflow rate of 30%.

As shown in Fig. 4(a), at the 500th iteration of the mGP stage in "EFF," all the electric fields in the region where the largest macro is located are in the upper right direction. Then, the macro will be excessively moved to the upper right by the electric field-driven density force on it. After that, as shown in Fig. 4(c), attributed to the macro having been excessively moved to the upper right, at the 505th iteration all electric fields in the region where the macro is located are therefore in the lower left direction, and subsequently this macro will be excessively moved to the lower left. Then as shown in Fig. 4(e), the oscillation of the solution will keep recurring, which will lead to an increase in wirelength. Contrarily, the right-side subfigures in Fig. 4 show that the potential-driven density forces smoothly move macros during the iterations of the mGP stage in the process of eliminating overlaps. Moreover, no oscillation of the solution occurs, and the other blocks move closer to macros, thus shortening the wirelength.

Since the only difference between "EFF" and Pplace-MS is that "EFF" employs the electric field-driven density force while Pplace-MS uses the potential-driven density force, the above oscillation of "EFF" should be due to the higher numerical error in the electric field-driven density force.

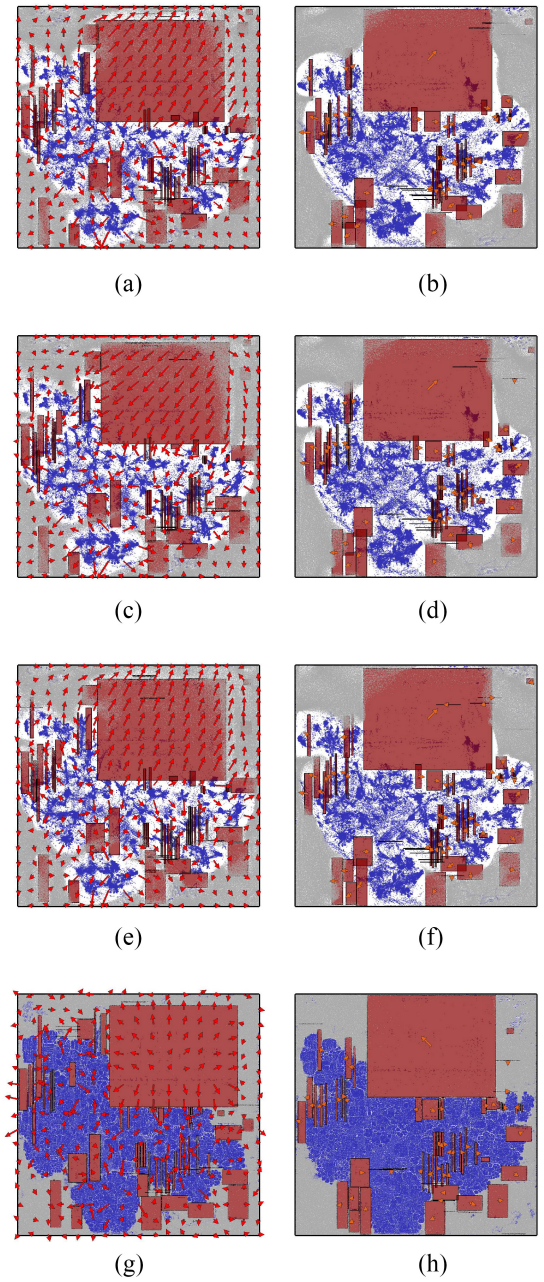

Fig. 4. Distribution of blocks in the mGP stage of the newblue1 benchmark. Macros, standard cells, and filler cells are denoted by red, blue, and gray rectangles, respectively. (a) Iteration: 500 ("EFF"). (b) Iteration: 380 (Pplace-MS). (c) Iteration: 505 ("EFF"). (d) Iteration: 385 (Pplace-MS). (e) Iteration: 510 ("EFF"). (f) Iteration: 390 (Pplace-MS). (g) Iteration: 660 ("EFF"). (h) Iteration: 505 (Pplace-MS).

C. Filler Cells and Bin Density Calculation

Motivated by ePlace [19], we use target density to guide the addition of fillers and scale the density weights of macros. For benchmarks in which macros have large differences in sizes, smaller macros may be placed between larger ones, thus creating more whitespace. So, the density weight of macros should be reduced to attract standard cells to fill these whitespaces.

Hence, we relate this weight to the size difference of macros. And we design a parameter θ_m to evaluate the size difference of macros as follows:

$$\theta_m = \frac{\sum_{v_i \in V_m} \left(A_i - \hat{A}_m\right)^2}{n_m \cdot \left(\max_{v_i \in V_m}\{A_i\} - \hat{A}_m\right) \cdot \left(\hat{A}_m - \min_{v_i \in V_m}\{A_i\}\right)}$$

where $\hat{A}_m$ is the average area of macros. When the size difference of macros is large, then we will get a larger θ_m. Based on the Gompertz function [9], we develop the virtual density weight $\widetilde{\rho}_t$ as follows:

$$\widetilde{\rho}_t = \rho_t + (1.0 - \rho_t)\left[\exp\left(\frac{1}{\theta_m - \theta_{\text{ref}}} \cdot \exp\left(-\max\left\{2, \frac{A_m}{A_s}\right\}\right)\right)\right] \tag{16}$$

where if the target density $\rho_t = 100\%$, then $\widetilde{\rho}_t = 1.0$. Here, A_m and A_s are the total areas of macros and standard cells, respectively. And ρ_t, a value between utilization and 1.0, is the target density given by the MMS benchmarks [27]. In the experiments, θ_{ref} is set to 1.8.

According to (16), the density weight of macros will be reduced when θ_m is large, thus attracting standard cells to fill whitespace between blocks. Moreover, for some cases in which the area of macros is much larger than the area of standard cells (e.g., the area of macros is more than twice the area of standard cells), we tend to give the macros a larger density weight to make the macro spread out faster. As analyzed in RePlAce [8], macros need to be moved to the boundary of the placement region to help improve the solution quality, by providing more space in the center of the placement region for the placement of standard cells. Therefore, for these cases, to facilitate the movement of macros, we give the macros a greater density weight and move these macros quickly to the boundary by using a larger density force. Hence, we add the term $\max\{2, (A_m/A_s)\}$ to the equation of $\widetilde{\rho}_t$. Then, the virtual density weight is used to guide the addition of fillers and scaling macros' density weights as follows.

1) Addition of Filler Cells: The same as ePlace [19], we add movable filler cells to the electrostatic system, which have no net connection to all blocks. The addition of filler cells not only effectively prevents overspreading of blocks but also controls the tightening of nets.

The total area A_f of filler cells is calculated by $A_f = \widetilde{\rho}_t \cdot A_{ws} - \widetilde{\rho}_t \cdot A_m - A_s$, where A_{ws} is the whitespace area of the placement region. The same as ePlace [19], we set the area of each filler cell as the average area of the middle 80% of the blocks sorted by area size. And we use V_f to denote the set of added filler cells. Then, we set the number of bins $|B| = K \times K$, where $K = \lceil \log_2 \sqrt{n + |V_f|} \rceil$.

2) Calculation of Bin Density: In this article, we use different weights for standard cells and macros when calculating the density of bins. Specifically, we set the weight of standard and filler cells to 1 and macros to the virtual density weight $\widetilde{\rho}_t$. For each bin b, the density is calculated by

$$\rho_b = \frac{1}{w_b h_b} \sum_{v_i \in V_s \cup V_f} d_x(b, v_i) \cdot d_y(b, v_i) + \frac{\widetilde{\rho}_t}{w_b h_b} \sum_{v_i \in V_m} d_x(b, v_i) \cdot d_y(b, v_i) \tag{17}$$

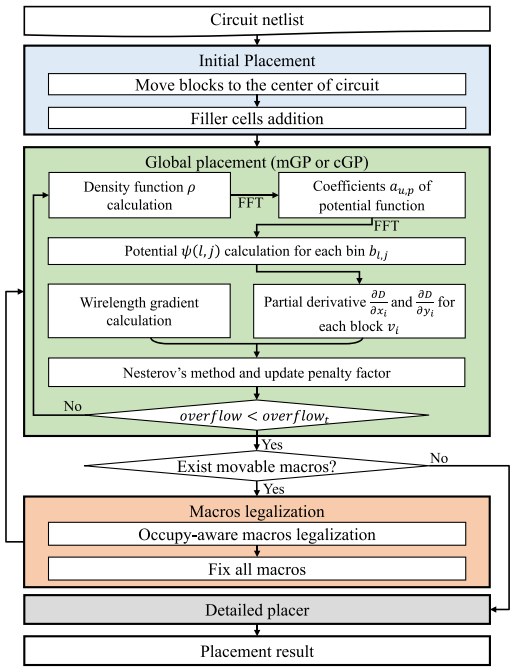

Fig. 5. Overview of our mixed-size placement algorithm Pplace-MS.

where $d_x(b, v_i)$ and $d_y(b, v_i)$ are the overlap functions in the horizontal and vertical directions, respectively, calculated as in (11) and (14).

As shown in (17), we give a smaller density weight $\widetilde{\rho}_t$ to macros. Therefore, during the density calculation, the density value of the bin in which a macro is located will be reduced, and thus its potential will also be reduced, which helps to keep the standard cells close to the macro.

D. Mixed-Size Global Placement Algorithm and Runtime Analysis

Fig. 5 gives an overview of Pplace-MS. It is decomposed into five stages: IP, mGP, mLG, standard cell global placement (cGP), and DP.

In the IP stage, we simply put all movable blocks to the center of the placement region. Subsequently, we add filler cells according to Section III-C at random positions in the placement region. In the DP stage, the same as [8] and [20], the detailed placer of NTUplace3 [6] is called to legalize standard cells and subsequently execute the DP to obtain a final placement solution. In this section, we mainly present the details of global placement and mLG stages.

1) Details of Global Placement: As shown in Fig. 5, in the mGP stage, we first calculate the density of each bin $b \in B$ by (17), which takes $O(K^2)$ time. Then, we invoke FFT once to calculate the coefficients $a_{u,p}$ in (5). Here, the time complexity of invoking FFT is $O(K^2 \log K)$, which takes up the main runtime in the electrostatics-based placer. Subsequently, we invoke FFT again to calculate (5). The partial derivatives $(\partial \mathcal{D}_i / \partial x_i)$ and $(\partial \mathcal{D}_i / \partial y_i)$ can be calculated by (13) for all blocks, which takes $O(n)$. Next, we calculate wirelength gradient by the gradient of LSE wirelength function. Then, the preconditioned Nesterov's method in ePlace [19] is used to optimize the positions of movable blocks by using

TABLE III
COMPARISON OF THE NUMBER OF TIMES THE FFT INVOKED IN EACH GP ITERATION AND THE AVERAGE RUNTIME OF GLOBAL PLACEMENT ON THE MMS BENCHMARKS

Mixed-Size placer	FFT times	Normalized average runtime
RePlAce	4	2.324×
ePlace-MS	3	
"EFF"	3	1.406×
Pplace-MS	2	1.000×

$(\partial f/\partial x_i)$ and $(\partial f/\partial y_i)$, in which the preconditioner in ePlace-MS [20] is adopted. Moreover, the penalty factor λ is updated by [19, eqs. (35) and (36)].

The electric potential of bins is determined by the global density distribution of blocks. The potential-driven density force will move blocks from a large potential to a small potential positions, thus reducing the overlap. Then, if the total density overflow is smaller than the target density overflow, the mGP stage will be terminated.

In the mLG stage, an occupy-aware mLG algorithm is called to eliminate the overlaps between macros, and then we will fix the locations of all macros. In the standard cGP stage, we use the global placement algorithm that is exactly the same as the mGP stage to perform standard cGP, which will optimize the overlaps caused by mLG.

2) Runtime Analysis: In each global placement iteration, Poisson's equation-based placers spend most computational resources on invoking the FFT in order to calculate the density force. To sum up, our approach needs to invoke the FFT twice for each iteration of global placement. While ePlace-MS [20] needs to call the FFT three times per iteration of global placement (see [20, eqs. (12) and (15)]). Furthermore, RePlAce [8] achieves the best wirelength as known by adding the *local density function* to the objective function, which makes it necessary to call the FFT four times per iteration of global placement (see [19, eqs. (21) and (24)] and [8, eqs. (7) and (8)]).

Table III compares the number of times the FFT is invoked in each global placement iteration, and the average runtime of global placement of respective placers on the MMS benchmarks.

Recently, GPU acceleration has been applied to ePlace. As a leading GPU-accelerated placer, DREAMPlace [17] spends 31.1% of the global placement runtime to calculate the gradient of potential function by the *density backward computation*. It is worth pointing out that our fast computation scheme is able to effectively reduce the runtime of GPU-accelerated placer by reducing the number of times the FFT is invoked. This situation can also be applied similarly to Xplace [18], an extremely fast global placement framework with GPU acceleration.

E. Occupy-Aware Macro Legalization

After the mGP stage, we get a solution $\mathbf{v}^0_{\text{mLG}} = (\mathbf{x}^0_{\text{mLG}}, \mathbf{y}^0_{\text{mLG}})$ with a small overlapping area between macros, as the solution shown in Fig. 4(h). This places a burden on the DP, which may significantly increase the wirelength. Hence, in this section, we propose an occupy-aware mLG algorithm to eliminate the overlaps between macros.

Some mixed-size placers [3], [25] legalize macros via floorplanning-based legalization approaches, which eliminate overlaps between macros by operating on floorplan representations of macros. However, due to the large size of macros, the placement will be significantly changed after each operation on the floorplan representation. In NTUplace3 [6], a macros shifting technique is presented, in which each macro is placed in its closest legal position by diamond search until there is no overlap between them. ePlace-MS [20] and RePlAce [8] legalize macros via an SA approach, which randomly chooses a macro, moves it to a new position randomly, then accepts this move with some probabilities.

In $\mathbf{v}^0_{\text{mLG}}$, a macro tends to be surrounded by its connected standard cells due to the wirelength force and the addition of filler cells in the mGP stage, which is well illustrated in Fig. 4(h). Therefore, each movement of the macros may increase the overlap area between macros and standard cells. Hence, we expect to eliminate the overlap between macros with minimal macro movement. The motivations for our mLG algorithm are as follows.

1) We expect the algorithm to legalize macros in linear time.
2) All overlapped macros are moved with tiny displacement in each iteration, so that connected standard cells remain close to it after the mLG. This guarantees the effectiveness of the reoptimization stage cGP after mLG.
3) Macros are encouraged to move toward the placement boundary, so that the moved macros take up more of the area of filler cells other than standard cells.

To this end, we first detect the overlaps between macros by defining an occupying function of macro v_i as

$$\theta_i(p,q) = \begin{cases} 0, & A(v_i, b_{p,q}) = 0 \\ 1, & A(v_i, b_{p,q}) \neq 0. \end{cases} \tag{18}$$

We present a sufficient condition for nonoverlapping of macros. Based on (18), we can prove the following results.

Lemma 1: There is no overlap between any two macros v_i and $v_j \in V_m$ if

$$\sum_{p=P_i^L}^{P_i^R} \sum_{q=Q_i^B}^{Q_i^U} \theta_j(p,q) = 0 \tag{19}$$

where P_i^L, P_i^R, Q_i^B, and Q_i^U are defined in (9) for block v_i.

Proof: According to (18), we have

$$\theta_j(p,q) \geq 0.$$

If (19) holds, for any $p \in [P_i^L, P_i^R]$ and any $q \in [Q_i^B, Q_i^U]$ we get $\theta_j(p,q) = 0$. That means the sets of bins occupied by macros v_i and v_j do not intersect, and consequently we have that v_i and v_j do not overlap. ■

Lemma 2: For all macros $v_i \in V_m$

$$\sum_{p=P_i^L}^{P_i^R} \sum_{q=Q_i^B}^{Q_i^U} (\theta_i(p,q) - 1) = 0.$$

Proof: Based on (18), we have

$$\sum_{p=P_i^L}^{P_i^R} \sum_{q=Q_i^B}^{Q_i^U} \theta_i(p,q) = \sum_{p=P_i^L}^{P_i^R} \sum_{q=Q_i^B}^{Q_i^U} 1 = (P_i^R - P_i^L)(Q_i^U - Q_i^B).$$

Hence

$$\sum_{p=P_i^L}^{P_i^R}\sum_{q=Q_i^B}^{Q_i^U}(\theta_i(p,q)-1)=0.$$

■

For bin $b_{p,q}\in B$, if more than one macro occupies $b_{p,q}$, we consider there may exist overlaps of macros in the region of bin $b_{p,q}$. We define the occupied potential of bin $b_{p,q}$ as follows:

$$\tilde{\mathcal{O}}(p,q)=\sum_{v_i\in V_m}\theta_i(p,q)-1. \tag{20}$$

So for each macro $v_i\in V_m$, the occupied potential energy of v_i can be defined as

$$\mathcal{O}_i(x_i,y_i)\triangleq\sum_{p=P_i^L}^{P_i^R}\sum_{q=Q_i^B}^{Q_i^U}d_x\big(b_{p,q},v_i\big)\cdot d_y\big(b_{p,q},v_i\big)\cdot\tilde{\mathcal{O}}(p,q).$$

Thus, a sufficient condition for macros nonoverlapping can be given by:

Theorem 2: There is no overlap between each macro $v_i\in V_m$ and any other macros if

$$\mathcal{O}_i(x_i,y_i)=0. \tag{21}$$

Proof: We prove this by contradiction. Assume that there exists a macro v_k overlaps with v_i, then according to Lemma 1, we get

$$\sum_{p=P_i^L}^{P_i^R}\sum_{q=Q_i^B}^{Q_i^U}\theta_k(p,q)\geq 1.$$

Then, based on Lemma 2 we have

$$\begin{aligned}&\sum_{p=P_i^L}^{P_i^R}\sum_{q=Q_i^B}^{Q_i^U}\left[\sum_{v_j\in V_m}\theta_j(p,q)-1\right]\\&\geq\sum_{p=P_i^L}^{P_i^R}\sum_{q=Q_i^B}^{Q_i^U}\left[\theta_i(p,q)+\theta_k(p,q)-1\right]\\&=\sum_{p=P_i^L}^{P_i^R}\sum_{q=Q_i^B}^{Q_i^U}\theta_k(p,q)\geq 1.\end{aligned}$$

According to (14), we have $d_x(b_{p,q},v_i)>0$ and $d_y(b_{p,q},v_i)>0$ for any $p\in[P_i^L,P_i^R]$ and any $q\in[Q_i^B,Q_i^U]$. Therefore, $\mathcal{O}_i(x_i,y_i)>0$, which contradicts to (21). ■

Hence, by Theorem 2, we can define the occupied potential energy of the mLG system as

$$\mathcal{O}(V_m)=\sum_{v_i\in V_m}\mathcal{O}_i(x_i,y_i).$$

Moreover, based on Theorem 2, we can establish the occupy-aware mLG problem as minimizing the occupied potential energy of the mLG system

$$\min_{V_m}\ \mathcal{O}(V_m).$$

Algorithm 1: Occupy-Aware mLG

Input : Solution after the mGP stage $\mathbf{v}_{\text{mLG}}^0=(\mathbf{x}_{\text{mLG}}^0,\mathbf{y}_{\text{mLG}}^0)$.
Output: Solution $\mathbf{v}_{\text{mLG}}=(\mathbf{x},\mathbf{y})$.

1 move $v_i\in V_m$ to the nearest row;
2 **while** $\mathcal{O}(V_m)\neq 0$ **do**
3 update $\tilde{\mathcal{O}}(p,q)$ by Equation (20) for each bin $b_{p,q}\in B$;
4 **for** *each macro* $v_i\in V_m$ *and* $\mathcal{O}_i\neq 0$ **do**
5 calculate $\left(\text{move}_i^x,\text{move}_i^y\right)$ by Equation (22);
6 calculate psb_i^A by Equation (23);
7 **if** *rand* $\gamma<psb_i^A$ **then**
8 continue;
9 $psb_i^x\leftarrow\text{calBoundFactorX}(x_i,\text{move}_i^x)$;
10 **if** *rand* $\gamma<psb_i^x$ **then**
11 $\text{move}_i^x\leftarrow 0$;
12 $psb_i^y\leftarrow\text{calBoundFactorY}(y_i,\text{move}_i^y)$;
13 **if** *rand* $\gamma<psb_i^y$ **then**
14 $\text{move}_i^y\leftarrow 0$;
15 $(x_i,y_i)\leftarrow(x_i,y_i)+\alpha\cdot(\text{sign}(\text{move}_i^x),\text{sign}(\text{move}_i^y))$;
16 update the position for each macro $v_i\in V_m$.

And the partial derivatives of the occupied potential energy to x_i and y_i of block $v_i\in V_m$ can be obtained by

$$\begin{aligned}\frac{\partial\mathcal{O}}{\partial x_i}=&\sum_{q=Q_i^B}^{Q_i^U}d_y\Big(b_{P_i^R,q},v_i\Big)\cdot\hat{d}_x^i\cdot\tilde{\mathcal{O}}\big(P_i^R,q\big)\\&-\sum_{q=Q_i^B}^{Q_i^U}d_y\Big(b_{P_i^L,q},v_i\Big)\cdot\hat{d}_x^i\cdot\tilde{\mathcal{O}}\big(P_i^L,q\big).\end{aligned}$$

Similarly, $([\partial\mathcal{O}]/[\partial y_i])$ can be obtained. The occupied force $\hat{F}_i=-([\partial\mathcal{O}]/[\partial x_i],[\partial\mathcal{O}]/[\partial y_i])$ moves each macro $v_i\in V_m$ out of the bin with high occupied potential.

The overview of our occupy-aware mLG algorithm is presented in Algorithm 1. Based on the solution $\mathbf{v}_{\text{mLG}}^0=(\mathbf{x}_{\text{mLG}}^0,\mathbf{y}_{\text{mLG}}^0)$ obtained by mGP, line 1 moves all macros to its nearest rows of the placement region. In lines 2–16, we iteratively optimize the positions of macros until the occupied potential energy of the mLG system is equal to zero, i.e., there is no overlap between macros. Line 3 updates the occupied potential of each bin $b_{p,q}\in B$ according to (20).

In lines 4–16 of Algorithm 1, we target macros with overlapping and optimize their positions. Line 5 calculates the occupied force of macro v_i by

$$\hat{F}_i=\left(\text{move}_i^x,\text{move}_i^y\right)=-\left(\frac{\partial\mathcal{O}}{\partial x_i},\frac{\partial\mathcal{O}}{\partial y_i}\right). \tag{22}$$

Line 6 calculates the area probability factor psb_i^A of v_i through

$$psb_i^A=\begin{cases}0, & \frac{A_i}{\hat{A}_m}\leq 1\\ 0.4\cdot\left(\frac{A_i}{\hat{A}_m}-1.0\right), & 1<\frac{A_i}{\hat{A}_m}<3\\ 0.8, & \frac{A_i}{\hat{A}_m}\geq 3\end{cases} \tag{23}$$

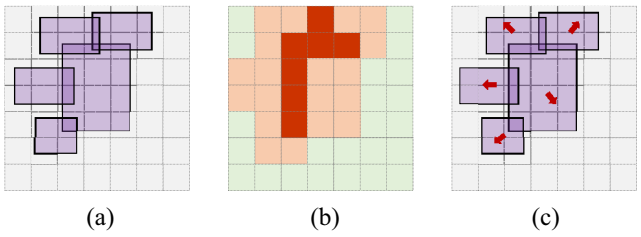

Fig. 6. (a) Macro layout with a small overlap rate; (b) occupied potential for bins; and (c) occupied force.

where $\hat{A}_m$ is the average area of macros. Here, if the area of macro v_i is more than three times of $\hat{A}_m$, then we set 80% as the probability of not moving the macro. If the area of macro v_i is less than or equal to $\hat{A}_m$, then this probability is 0%. The probabilities of macros of other sizes are obtained linearly based on these two probabilities. In line 7, a random number $\gamma \in (0, 1)$ is generated to determine whether to move the macro v_i in this iteration. In this case, line 8 will not make any movement to macro v_i, and directly go to the next loop to optimize the next macro. Lines 9–11 calculate the probability psb_i^x of moving the macro v_i in the x-direction, where psb_i^x is calculated by

$$psb_i^x = \begin{cases} 0.5, \text{ if } \operatorname{sign}\left(-\text{move}_i^x\right) \neq \operatorname{sign}\left(x_i - \frac{W}{2}\right) \\ 0, \quad \text{otherwise.} \end{cases}$$

Then, we generate a random number $\gamma \in (0, 1)$ to compare with psb_i^x. If $\gamma < psb_i^x$, we will not move macro v_i in the x-direction. Here, psb_i^x can guide macros to move more toward the placement boundary. Further, lines 12–14 execute similar procedures for macro v_i in the y-direction. Line 15 updates solution (x_i, y_i) of v_i according to move_i obtained through the above flow and the steplength $\alpha = (w_b, h_r)$, where h_r is the row height of the circuit. Finally, the positions of macros are updated in line 16 based on $(\mathbf{x}, \mathbf{y})$.

To demonstrate the effectiveness of our occupy-aware mLG algorithm, we give an example as shown in Fig. 6(a), where the purple blocks are macros and grids represent bins (in practice a macro will be much larger than the bin size). As shown in Fig. 6(b), the distribution of macros in Fig. 6(a) is mapped to the occupation potential, where a green bin means its occupied potential is equal to -1, an orange bin means its occupied potential is equal to 0, and a red bin means its occupied potential is greater than 0. The occupation potential comprehends the overlap of macros, and the occupied force $\hat{F}_i$ for each $v_i \in V_m$, as shown in Fig. 6(c), moves macros out of red bins, thus reducing the occupation potential of macros and eliminating the overlap between macros.

Since the movement of macros in the mLG stage causes overlaps between macros and standard cells, we need to reoptimize the positions of standard cells to eliminate these overlaps. This reoptimization process is renamed standard cGP. First, we inherit the penalty factor λ of the last iteration of the mGP stage using the same method as ePlace-MS [20]: $\lambda_{\text{cGP}}^{\text{init}} = \lambda_{\text{mGP}}^{\text{last}} \times 1.1^m$, where $\lambda_{\text{cGP}}^{\text{init}}$ is the initial penalty factor in the cGP stage, and $\lambda_{\text{mGP}}^{\text{last}}$ is the last penalty factor in the mGP stage, m is the number of mGP stage iterations divided by 10. In order to move filler cells overlapped with macros first, we execute the same algorithm as mGP to optimize only filler cells for 15 iterations. Then, we use the same algorithm of the mGP stage to optimize standard cells and filler cells together until the density overflow is smaller than the target density overflow.

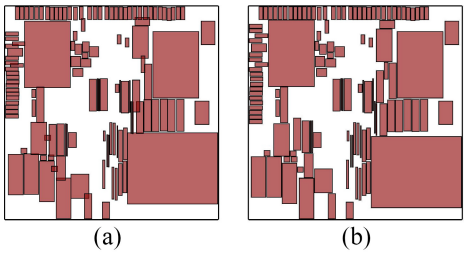

Fig. 7. Macros layout on adaptec2. (a) After the mGP stage with the occupied potential energy $\mathcal{O}(V_m) = 9426$. (b) At the 176th iteration of the mLG stage with the occupied potential energy $\mathcal{O}(V_m) = 0$.

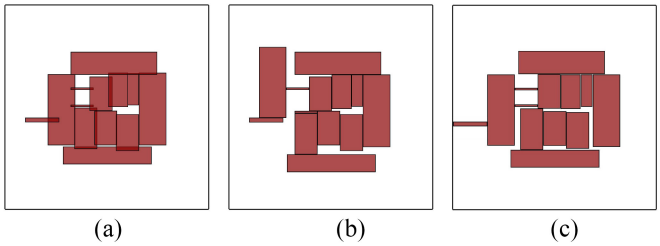

Fig. 8. (a) Macro layout with a small overlap rate. (b) Macro layout with $\sum_{v_i \in V_m} A_i \cdot \|\text{Move}_i\| = 1352656.496$ after the legalization of NTUplace3 [6] is invoked. (c) Macro layout with $\sum_{v_i \in V_m} A_i \cdot \|\text{Move}_i\| = 924825.443$ after our occupy-aware mLG is invoked.

F. Performance of Macro Legalization

In the occupy-aware mLG algorithm, the gradient of occupied potential energy is used to guide the movement of macros, which makes our mLG more robust and avoids invalid movement of macros. In addition, our algorithm moves all target macros in each iteration, which prevents some macros from moving excessively. Moreover, this facilitates the cGP stage to move standard cells out of the macros region more efficiently.

As shown in Fig. 7, after the execution of our occupy-aware mLG algorithm, there is no overlap between macros in macro layout. Also, only a small amount of displacement is applied to macros and thus the distribution of macros is not significantly changed, which will preserve the "good" solution from the mGP.

Fig. 8(a) gives an example of macros layout to verify the effectiveness of our occupy-aware mLG algorithm. The impact of the movement of a macro v_i on the final placement solution is related to the area A_i of the macro. Therefore, we take the total penalized movement $\sum_{v_i \in V_m} A_i \cdot \|\text{Move}_i\|$ to assess the quality of our algorithm, where $\|\text{Move}_i\|$ is the amount of movement of macro v_i. As shown in Fig. 8(b), the legalization of NTUplace3 [6] produces large movements of macros, which tends to have a great impact on solution quality. In Fig. 8(c), only a small amount of movement is generated by our algorithm.

IV. Experimental Results

The proposed mixed-size placement algorithm is implemented in C++ programming language and executed on a

TABLE IV
STATISTICS OF THE MMS BENCHMARKS [27]

Benchmark	#Objects	#Movable Objects	#Standard Cells	#Macros	#Fixed I/Os	#Nets	#Pins	Target Density
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 V
COMPARISONS OF SHPWL ($\times 10^6$) AND RUNTIME (MINUTES) AMONG NTUPLACE3 [6], EPLACE-MS [20], REPLACE [8], "EFF," AND OUR PPLACE-MS BASED ON THE MMS BENCHMARKS [27]. CITED RESULTS ARE MARKED WITH †

Benchmark	NTUplace3 [6]		ePlace-MS† [20]				RePlAce [8]						"EFF"		Pplace-MS	
			LSE		WA		RePlAce-ld		RePlAce-ds		RePlAce-ldds					
	sHPWL	CPU	sHPWL	CPU	sHPWL	CPU	sHPWL	CPU	sHPWL	CPU	sHPWL	CPU	sHPWL	CPU	sHPWL	CPU
adaptec1	75.92	1.33	66.99	5.25	66.82	5.47	65.14	2.44	65.48	7.03	64.69	8.47	67.00	1.48	66.33	1.12
adaptec2	84.89	2.17	76.74	7.58	76.76	7.43	73.51	3.56	72.79	12.46	N/A	N/A	79.18	2.16	72.25	1.63
adaptec3	170.88	3.68	161.63	26.22	161.55	27.23	151.82	9.36	149.82	22.17	150.09	29.54	153.80	5.85	152.05	4.56
adaptec4	167.13	2.48	145.89	56.4	147.04	29.35	140.22	14.31	138.94	36.39	138.71	55.32	141.88	9.68	139.55	8.03
bigblue1	96.42	2.65	87.27	7.85	86.29	7.82	85.74	3.71	87.18	10.03	85.02	12.32	85.59	2.29	85.58	1.74
bigblue2	148.12	4.33	132.72	13.97	130.06	13.70	124.13	6.77	124.07	17.70	123.66	22.59	123.61	4.19	125.13	3.33
bigblue3	324.39	11.73	287.34	82.20	284.39	72.98	282.51	30.83	297.53	62.11	276.24	75.76	272.97	13.75	265.92	10.30
bigblue4	797.17	27.42	660.17	141.37	656.68	204.15	647.75	59.71	640.17	124.74	641.61	159.18	645.56	44.92	646.45	30.65
adaptec5	295.24	13.27	304.68	50.27	312.86	48.35	304.44	14.93	303.88	32.61	303.29	36.92	299.25	12.07	263.07	8.91
newblue1	61.13	2.82	60.43	11.70	61.87	10.87	58.69	4.12	58.32	10.23	57.31	12.37	59.00	2.57	57.67	1.95
newblue2	164.25	3.10	159.11	51.12	162.98	62.40	152.29	5.51	N/A	N/A	N/A	N/A	153.61	3.57	150.85	2.60
newblue3	N/A	N/A	287.69	30.57	304.16	17.53	260.71	7.83	N/A	N/A	N/A	N/A	268.28	13.17	255.12	5.89
newblue4	231.59	6.45	226.29	28.27	229.20	29.73	224.81	10.58	224.9	23.64	224.94	27.47	223.75	6.65	209.88	5.09
newblue5	414.81	19.07	392.77	55.47	392.93	63.40	391.71	23.97	388.16	60.56	388.84	71.72	385.33	19.64	369.70	12.98
newblue6	471.51	13.88	414.56	112.62	409.28	69.65	407.43	26.72	406.87	63.39	405.75	81.61	408.01	17.07	409.10	12.67
newblue7	N/A	N/A	889.18	392.02	895.11	191.47	872.36	59.67	878.75	117.26	880.49	145.44	871.64	36.12	868.20	26.03
Normalized	1.147	1.169	1.058	7.105	1.065	6.436	1.023	2.054	1.028	5.158	1.019	6.214	1.030	1.397	1.000	1.000

Linux machine with Intel Core i9-12900 3.20-GHz CPU and 32-GB memory in single-threaded mode.

We perform a comparison study using the MMS benchmarks [27], which are based on the ISPD-2005 [22] and ISPD-2006 [21] benchmarks. The MMS benchmarks inherit the same netlists and target density ρ_t from them, and some macros are movable. The statistics of the MMS benchmarks [27] are listed in Table IV, where "#Objects," "#Movable Objects," "#Standard Cells," "#Macros," "#Fixed I/Os," "#Nets" and "#Pins" give the total numbers of blocks, movable blocks, standard cells, movable macros, fixed I/O pads, nets, and pins, respectively. In the table, "Target Density" is the target density given by ISPD-2005 [22] and ISPD-2006 [21].

In Section IV-A, we test Pplace-MS on the MMS benchmarks [27] and compare Pplace-MS's performance with other advanced mixed-size placers. In Section IV-B, we conduct ablation studies to show the efficiency of our virtual density weight technique (VD) proposed in Section III-C, and the occupy-aware mLG presented in Section III-E.

A. Validation on the MMS Benchmarks

On the MMS benchmarks, we compare our Pplace-MS with NTUplace3 [6], ePlace-MS [20], RePlAce [8], and "EFF." RePlAce has achieved the best published wirelength-driven mixed-size placement results by disabling the routability function. And RePlAce has three different schemes for the MMS benchmarks: 1) "RePlAce-ld": scheme with the *local density function* in the objective function; 2) "RePlAce-ds": scheme using *dynamic step size adaptation* technique by adding a trial placement stage before the mGP stage; and 3) "RePlAce-ldds": scheme combining the above two schemes. The binary codes of the three schemes can be obtained from [1]. "EFF" is our Pplace-MS with the potential-driven density force replaced by the electric field-driven density force in ePlace-MS (see Section III-B for details).

We executed the binary codes on the same machine, and the test results are summarized in Table V. In addition, the same as [8] and [20], the binary code of the detailed placer provided by NTUplace3 [6] was called to obtain the final placement result. The binary code of ePlace-MS is not available, hence we cite their results directly from [20]. If the density of a local area exceeds the target density ρ_t, then HPWL is penalized to sHPWL, calculated using the official evaluation script in [21].

Note that, on different execution environments the binary codes sometimes may produce placement solutions different from those reported in the corresponding paper. Therefore,

TABLE VI
COMPARISON OF THE POST-GLOBAL PLACEMENT AND POST-LEGALIZATION HPWL ($\times 10^6$) RESULTS OF REPLACE [8] AND PPLACE-MS

Benchmark	RePlAce									Pplace-MS		
	RePlAce-ld			RePlAce-ds			RePlAce-ldds					
	Post-GP	Post-Leg	Diff(%)	Post-GP	Post-Leg	Diff(%)	Post-GP	Post-Leg	Diff(%)	Post-GP	Post-Leg	Diff(%)
adaptec1	62.70	65.97	5.22	62.98	66.24	5.18	62.46	65.37	4.66	62.54	67.57	7.58
adaptec2	70.89	74.33	4.85	68.78	74.01	7.60	N/A	N/A	N/A	68.87	73.38	6.24
adaptec3	146.96	153.38	4.37	144.43	151.69	5.03	144.46	152.03	5.24	146.48	153.94	4.91
adaptec4	136.03	142.23	4.56	134.17	141.31	5.32	134.14	140.76	4.94	134.92	141.81	4.94
bigblue1	83.41	86.39	3.57	85.14	87.96	3.31	82.49	85.77	3.97	83.57	86.36	3.26
bigblue2	120.29	125.65	4.46	119.31	125.84	5.47	119.05	125.31	5.25	121.92	126.7	3.82
bigblue3	264.95	277.01	4.55	287.03	301.11	4.91	264.56	280.01	5.84	254.47	270.6	6.07
bigblue4	628.89	655.03	4.16	617.98	648.60	4.95	618.26	649.87	5.11	626.14	654.33	4.36
adaptec5	299.79	305.23	1.81	299.42	304.42	1.67	298.74	304.07	1.78	242.89	251.27	3.39
newblue1	57.44	59.87	4.23	56.93	59.44	4.41	55.92	58.39	4.42	55.2	57.98	4.90
newblue2	148.28	154.34	4.09	N/A	N/A	N/A	N/A	N/A	N/A	146.42	152.32	3.92
newblue3	256.99	263.49	2.53	N/A	N/A	N/A	N/A	N/A	N/A	250.4	258.94	3.35
newblue4	228.71	230.82	0.92	225.41	227.74	1.03	225.21	227.58	1.05	195.3	199.11	1.96
newblue5	385.83	393.16	1.90	382.00	389.40	1.94	382.57	390.07	1.96	343	352.07	2.63
newblue6	402.07	411.79	2.42	401.24	411.12	2.46	400.04	410.10	2.52	397.72	407.63	2.46
newblue7	857.28	880.28	2.68	862.2	886.66	2.84	865.29	888.21	2.65	840.86	865.52	2.88
Normalized	1.045	1.037	0.818	1.047	1.043	0.932	1.040	1.035	0.883	1.000	1.000	1.000

we consider placement solutions that are unavailable or whose wirelengths are 20% worse than the results reported in respective publications as not applicable (N/A) for fair comparisons. Except the N/As, in Table V, the average HPWLs of "RePlAce-ld," "RePlAce-ds," and "RePlAce-ldds" are 0.085% shorter, 0.431% longer, and 0.419% shorter than those reported in RePlAce [8], so we think these results are valid.

Compared to "RePlAce-ld," our Pplace-MS achieves 2.3% sHPWL reduction and 2.054× faster runtime on average and outperforms "RePlAce-ld" in 12 out of 16 benchmarks on sHPWL. Compared to "RePlAce-ds," our Pplace-MS achieves 2.8% sHPWL reduction and 5.158× faster runtime on average, and outperforms "RePlAce-ds" in 8 out of 14 benchmarks on sHPWL. Compared to "RePlAce-ldds," our Pplace-MS achieves 1.9% sHPWL reduction and 6.214× faster runtime on average, and outperforms "RePlAce-ldds" in 5 out of 13 benchmarks on sHPWL. Compared to "EFF," our Pplace-MS achieves 3.0% sHPWL reduction and 1.397× faster runtime on average, and outperforms "EFF" in 14 out of 16 benchmarks on sHPWL.

Table VI presents the HPWL results of the post-global placement and the post-legalization of RePlAce and Pplace-MS. Compared with "RePlAce-ld," "RePlAce-ds," and "RePlAce-ldds," the post-global placement of Pplace-MS achieves 4.5%, 4.7%, and 4.0% HPWL reduction, respectively, on average. Moreover, the HPWL of the post-legalization of Pplace-MS is 3.7%, 4.3%, and 3.5% less than respective algorithms on average. However, the difference between HPWL before and after legalization by Pplace-MS is higher than that of "RePlAce-ld," "RePlAce-ds," and "RePlAce-ldds" by 18.2%, 6.8%, and 11.7%, respectively. Although Pplace-MS increases the HPWL more in legalization, it still achieves the best HPWL results in the post-legalization.

B. Ablation Studies

In this section, we perform ablation studies on our proposed VD and occupy-aware mLG to demonstrate their effectiveness. The results are presented in Table VII.

In Table VII, the "w/o mLG" column presents the experimental results of our Pplace-MS by skipping our mLG to directly perform the detailed placer of NTUplace3 [6] after the mGP stage. Experimental results show that our Pplace-MS with the occupy-aware mLG achieves 5.4% sHPWL reduction, compared to without the mLG.

TABLE VII
ABLATION STUDIES OF OCCUPY-AWARE MLG AND VD ON THE MMS BENCHMARKS [27]

Benchmark	Pplace-MS		w/o mLG		w/o VD	
	sHPWL	CPU	sHPWL	CPU	sHPWL	CPU
adaptec1	66.33	1.12	71.48	1.08	—	—
adaptec2	72.25	1.63	86.19	1.59	—	—
adaptec3	152.05	4.56	164.31	4.47	—	—
adaptec4	139.55	8.03	145.63	7.84	—	—
bigblue1	85.58	1.74	85.93	1.72	—	—
bigblue2	125.13	3.33	126.04	3.35	—	—
bigblue3	265.92	10.30	311.01	9.86	—	—
bigblue4	646.45	30.65	686.82	30.81	—	—
adaptec5	263.07	8.91	264.26	8.41	273.26	10.41
newblue1	57.67	1.95	58.03	1.84	59.78	2.02
newblue2	150.85	2.60	156.42	2.48	154.04	2.59
newblue3	255.12	5.89	290.54	9.16	253.86	6.3
newblue4	209.88	5.09	211.08	4.65	236.67	4.91
newblue5	369.70	12.98	371.95	12.68	384.38	13.61
newblue6	409.10	12.67	410.05	12.32	411.09	12.54
newblue7	868.20	26.03	888.53	230.88	870.42	28.37
Normalized	1.000	1.000	1.054	1.002	1.033	1.045

The "w/o VD" column in Table VII gives the experimental results of our Pplace-MS without using the VD, mentioned in Section III-C. It is worth noting that, for benchmarks with 100% target density (see Table IV), the macro virtual density obtained from (16) is still 1.0, hence we do not use these benchmarks to test the VD. Experimental results show that the addition of the virtual density weight achieves 3.3% sHPWL reduction for benchmarks with the specified target density on the MMS benchmarks.

V. CONCLUSION

This article has proposed a novel mixed-size placement algorithm with Poisson's equation-based global placement and occupy-aware mLG. The mGP is based on a new definition of potential energy for each block based on an analytical solution of Poisson's equation, and a fast approximate computation

scheme for partial derivatives of the potential energy. The occupy-aware mLG can eliminate overlaps between macros accurately and effectively. Experimental results on the MMS benchmarks have shown that the proposed algorithm achieves higher performance over leading electrostatics-based mixed-size placer in single-threaded mode. We believe the proposed scheme can also be accelerated using GPU architecture, as in [17] and [18].

REFERENCES

[1] "RePlAce," Accessed: Sep. 8, 2022. [Online]. Available: https://github.com/mgwoo/RePlAce

[2] S. Adya, S. Chaturvedi, J. Roy, D. Papa, and I. Markov, "Unification of partitioning, placement and floorplanning," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design*, 2004, pp. 550–557.

[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*, 2006, pp. 212–214.

[4] Y.-W. Chang, Z.-W. Jiang, and T.-C. Chen, "Essential issues in analytical placement algorithms," *IPSJ Trans. Syst. LSI Design Methodol.*, vol. 2, pp. 145–166, Aug. 2009.

[5] H.-C. Chen, Y.-L. Chuang, 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*, 2008, pp. 218–223.

[6] 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 constraints," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 7, pp. 1228–1240, Jul. 2008.

[7] T.-C. Chen, P.-H. Yuh, Y.-W. Chang, F.-J. Huang, and T.-Y. 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.

[8] C.-K. Cheng, A. B. Kahng, I. Kang, and L. Wang, "RePlAce: Advancing solution quality and routability validation in global placement," *IEEE Trans. Comput.-Aided Design Integr. Circuits Systems*, vol. 38, no. 9, pp. 1717–1730, Sep. 2019.

[9] B. Gompertz, "On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies," *Philos. Trans. Royal Soc. London B, Biol. Sci.*, vol. 27, pp. 513–524, Jan. 1825.

[10] J. Gu, Z. Jiang, Y. Lin, and D. Z. Pan, "DREAMPlace 3.0: Multi-electrostatics based robust vlsi placement with region constraints," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design*, 2020, pp. 1–9.

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

[12] 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.

[13] A. Kahng and Q. Wang, "An analytic placer for mixed-size placement and timing-driven placement," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, 2004, pp. 565–572.

[14] A. B. Kahng and Q. Wang, "A faster implementation of APlace," in *Proc. ACM Int. Symp. Phys. Design*, New York, NY, USA, 2006, pp. 218–220.

[15] M.-C. Kim and I. L. Markov, "ComPLx: A competitive primal-dual Lagrange optimization for global placement," in *Proc. ACM/IEEE Design Autom. Conf.*, 2012, pp. 747–755.

[16] X. Li, K. Peng, F. Huang, and W. Zhu, "PeF: Poisson's equation based large-scale fixed-outline floorplanning," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 6, pp. 2002–2015, Jun. 2023.

[17] Y. Lin, S. Dhar, W. Li, H. Ren, B. Khailany, and D. Z. Pan, "DREAMPlace: Deep learning toolkit-enabled GPU acceleration for modern VLSI placement," in *Proc. ACM/IEEE Design Autom. Conf.*, 2019, pp. 1–6.

[18] L. Liu, B. Fu, M. D. Wong, and E. F. Young, "Xplace: An extremely fast and extensible global placement framework," in *Proc. ACM/IEEE Design Autom. Conf.*, 2022, pp. 1309–1314.

[19] J. Lu, P. Chen, C. C. Chang, L. Sha, and C. K. Cheng, "ePlace: Electrostatics-based placement using fast fourier transform and Nesterov's method," *ACM Trans. Design Autom. Electron. Syst.*, vol. 20, no. 2, pp. 1–34, 2015.

[20] J. Lu et al., "ePlace-MS: Electrostatics-based placement for mixed-size circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 5, pp. 685–698, May 2015.

[21] G.-J. Nam, "ISPD 2006 placement contest: Benchmark suite and results," in *Proc. ACM Int. Symp. Phys. Design*, 2006, p. 167.

[22] G. J. Nam, C. J. Alpert, P. Villarrubia, B. Winter, and M. C. Yildiz, "The ISPD2005 placement contest and benchmark suite," in *Proc. ACM Int. Symp. Phys. Design*, 2005, pp. 216–220.

[23] W. C. Naylor, D. Ross, and S. Lu, "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.

[24] 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.

[25] N. Viswanathan, M. Pan, and C. Chu, "FastPlace 2.0: An efficient analytical placer for mixed-mode designs," in *Proc. Asia–South Pacific Conf. Design Autom.*, 2006, pp. 1–6.

[26] N. Viswanathan, M. Pan, and C. Chu, "FastPlace 3.0: A fast multilevel quadratic placement algorithm with placement congestion control," in *Proc. Asia–South Pacific Design Autom. Conf.*, 2007, pp. 135–140.

[27] J. Z. Yan, N. Viswanathan, and C. Chu, "Handling complexities in modern large-scale mixed-size placement," in *Proc. ACM/IEEE Design Autom. Conf.*, 2009, pp. 436–441.

[28] W. Zhu, Z. Huang, J. Chen, and Y.-W. Chang, "Analytical solution of Poisson's equation and its application to VLSI global placement," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, 2018, pp. 1–8.

Keyu Peng received the B.Sc. degree in information and computing sciences from Northeast Forestry University, Harbin, China, in 2019, and the M.Sc. degree in operations research and cybernetics from Fuzhou University, Fuzhou, China, in 2023. He is currently pursuing the Ph.D. degree with Southeast University, Nanjing, China.

His research interest is VLSI physical design automation.

Wenxing Zhu received the B.Sc. degree in applied mathematics and the M.Sc. and Ph.D. degrees in operations research from Shanghai University, Shanghai, China, in 1989, 1992, and 1996, respectively.

He joined Fuzhou University, Fuzhou, China, in 1996 and was promoted to a Professor in 2004. His research interests include algorithms for VLSI physical design automation, and optimization theory and applications.