

MOGA-CmpCNN: A Multi-Objective Genetic Algorithm Integrated with CmpCNN Model for CMP-Aware Metal Fill

Anonymous Author(s)

Submission Id: 997

ABSTRACT

Metal fill plays a vital role in VLSI design by ensuring a level surface, enhancing manufacturing robustness, and reducing defects. Present density-driven metal fill algorithms can result in larger post-CMP height variations, posing challenges for smaller tech nodes. Therefore, we propose a MOGA-CmpCNN framework, which employs a multi-objective genetic algorithm that integrates a CmpCNN model to enhance layout planarity by effectively guiding the filling process. A multi-objective fitness score is designed to reduce the fill amount and lateral capacitance while maintaining a small variation in post-CMP height. Experimental results demonstrate that our MOGA-CmpCNN framework achieves a significant improvement in layout planarity, ranging from 61% to 94%, while also reducing the fill amount and lateral capacitance substantially.

KEYWORDS

Design for Manufacturability, Metal Fill, Neural Network, Genetic Algorithm

1 INTRODUCTION

Chemical mechanical polishing (CMP) is a crucial manufacturing process for achieving uniform planarization in the fabrication of integrated circuits (ICs) [1]. Due to the strong correlation between post-CMP height and underlying feature density, foundries highly recommend metal fill to achieve uniform density [2]. Metal fill improves the layout planarization and enhances manufacturing robustness, playing a vital role in IC design and manufacturing.

Metal fill algorithms typically operate in two phases, fill synthesis and fill insertion, where fill synthesis determines the fill amount to be inserted and fill insertion places the fill in the layout [3]. These algorithms are broadly classified into rule-based and model-based approaches [4]. Rule-based algorithms rely on predefined density related rules and guidelines to determine the fill features, which are provided by the foundry based on their empirical knowledge [5]. Refs. [6–9] proposed rule-based metal fill algorithms with given design rules and other constraints, such as density and timing. Though these approaches are relatively simple to implement, the density related rules can result in larger post-CMP height variations [10].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

DAC'24, June 23–27, 2024, Moscone West, San Francisco

© 2024 Association for Computing Machinery.

ACM ISBN 978-1-4503-5700-5/18/06...\$15.00

<https://doi.org/10.1145/1122445.1122456>

Model-based algorithms optimize fill solution to reduce the height variations by utilizing accurate CMP models describing the correlation between layout and post-CMP height [4]. Refs. [4, 10] proposed metal fill algorithms with empirical models, significantly reducing height variations. These empirical models account solely for density/perimeter factors, necessitating improvements to enhance predictive accuracy. To improve model precision and minimize height variations, Ref. [11] proposed a fill insertion framework integrated with a commercial full-chip CMP simulator. The computation time is relatively high even with 64 CPU cores for invoking the CMP simulator repeatedly [12].

Embracing machine learning (ML) techniques has resulted in substantial progress in CMP modeling [12–16], improving both efficiency and precision. Ref. [12] proposed a NeurFill framework with a CMP neural network as the CMP simulator for fill synthesis, achieving a remarkable 8134× speedup compared to Ref. [11]. A multi-modal algorithm for fill insertion has been proposed, yielding a 0.5%–3.4% enhancement in fill quality at the cost of increase in overall runtime. While it achieves significant performance gains in reducing height variations, handling fill synthesis and fill insertion as separate processes can result in a lack of holistic perspective during the synthesis stage. Hence, it is essential to approach the metal fill process holistically to obtain an optimal fill solution.

In this work, we propose a MOGA-CmpCNN framework for metal fill. The key contributions are summarized below.

- (1) **Multi-Objective Genetic Algorithm (MOGA)** is proposed to optimize the fill solution holistically, adhering to density constraints with less height variations and fill amount. Experimental results demonstrate remarkable fill amount reduction of over 60% while maintaining relatively small height variations.
- (2) **CmpCNN Model** is incorporated into MOGA for precise height prediction, guiding MOGA towards a fill solution with minimum height variations. Experimental results reveal that integrating the CmpCNN model yield a substantial reduction in overall variation, ranging from 61% to 94%.
- (3) **Different Fill Patterns** are employed to investigate their effect on height variations, fill amount, and capacitance. Experimental results underscore the importance of selecting a pattern based on specific requirements and trade-offs.

2 PRELIMINARIES

2.1 Fill Patterns

Considering the fill pattern in the metal fill process is of paramount importance to increase the layout uniformity and reliability [17]. Rectangular metal fill is one of the most commonly used fill patterns, which evenly distributes metal density [18]. More advanced patterns (e.g., staggered, alternate, and

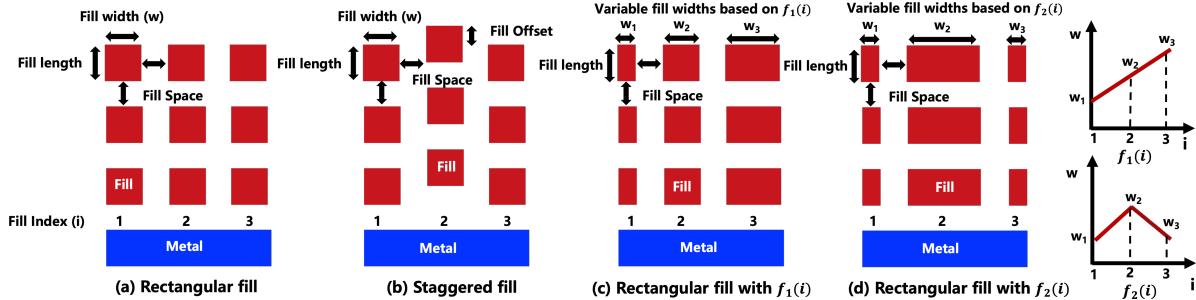


Figure 1: Examples of (a) rectangular fill, (b) staggered fill, (c) rectangular fill with $f_1(i)$, and (d) rectangular fill with $f_2(i)$.

basket-weave) are explored to enhance manufacturability, where staggered is preferred with the ability to result in a more uniform etch rate [19]. Besides, rectangular pattern with different distribution characteristic functions (DCF) have also been explored to lower the capacitance, whose *fill width* or *fill length* are variable following different DCFs [17]. In this work, we explore optimal fill solutions using diverse fill patterns (Fig. 1), including rectangular, staggered, and rectangular fills with different DCFs. Two different DCFs ($f_1(i)$ and $f_2(i)$) of our work are set as follows:

$$f_1(i) = i \times w, \quad f_2(i) = \begin{cases} i \times w, & \text{when } i \leq th \\ (2 \times th - i) \times w, & \text{when } i > th \end{cases}, \quad (1)$$

where w denotes *fill width*. i and th denote the index of the fill and the threshold index, respectively.

2.2 CmpCNN Model

To minimize the post-CMP height variation due to poor fill solution, an accurate CMP model for height prediction to guide the fill process is of great importance [20]. Convolutional neural networks (CNNs) have demonstrated significant success in building CMP models [12–16], which achieve remarkable accuracy while also maintaining efficient runtime performance. Specifically, the CmpCNN model achieves 1.89× error reduction compared to the prior work, and a 57× speedup compared to the commercial CMP simulation tool [16]. The CmpCNN model consists of 5 convolution blocks for feature extraction and 2 fully connected (FC) blocks for regression [16]. It is multi-input neural network model, which takes the layout image with size of 400×400 and its density as input and predict the corresponding height. In this work, we utilize the CmpCNN model for height prediction to minimize the height variation of the fill solution.

2.3 Definitions and Problem Formulation

To evaluate the performance of fill solutions, we have introduced the following metrics.

Definition 1. *Overall Variation (OV)* computes the sum of standard deviations of height per layer σ_k .

$$OV = \sum_{k=1}^K \sigma_k = \sum_{k=1}^K \frac{\sum_{m=1}^M \sum_{n=1}^N (h_{k,m,n} - \bar{h}_k)^2}{M * N}, \quad (2)$$

where K , M , and N denote the total layers, columns, and rows of layout windows. $h_{k,m,n}$ denotes the window height, and \bar{h}_k denotes the average height of the corresponding layer.

Definition 2. *Max Variation (MV)* computes the maximum deviation of accumulated height across layers.

$$MV = \max(H_{m,n}) - \min(H_{m,n}), \quad (3)$$

where $H_{m,n} = \sum_{k=1}^K h_{k,m,n}$, which denotes the cumulative height of each window.

Definition 3. *Overall Density (OD)* computes the overall density of the entire layout across layers.

$$OD = \frac{\sum_{k=1}^K \sum_{m=1}^M \sum_{n=1}^N d_{k,m,n}}{K * M * N}, \quad (4)$$

where $d_{k,m,n}$ denotes the metal density of each window.

Definition 4. *Lateral Capacitance (LC)* computes the capacitance formed by the conductors on the same layer [21].

$$LC = P_l(d) \times l, \quad (5)$$

where l , d , and P_l represent the length of parallel edges of conductors, the distance between parallel edges, and lateral capacitance per unit length, respectively.

Definition 5. *Fill Amount (FA)* computes the total fill amount of each window across layers.

$$FA = \sum_{k=1}^K \sum_{m=1}^M \sum_{n=1}^N \#fill_{k,m,n}, \quad (6)$$

where $\#fill_{k,m,n}$ denotes the fill amount of each window.

Problem 1. Given a layout with a uniformed window size $W_1 \times W_2$. The primary goal is to insert fills with minimum *overall variation*, *max variation*, *overall density*, *fill amount*, and *lateral capacitance* while meeting the design rule constrains (DRC).

3 MOGA-CmpCNN FRAMEWORK

In this work, we propose a MOGA-CmpCNN framework for metal fill, including (1) **layout discretization and fillable region extraction**, (2) **initial height prediction and target height assignment**, and (3) **MOGA-CmpCNN fill insertion**.

3.1 Layout Discretization and Fillable Region Extraction

In this work, we first perform layout discretization to divide the layout into multiple windows with a uniform size $W_1 \times W_2$. Afterwards, we extract fillable regions of each window for subsequent fill insertion.

To ensure DRC-compliant fill insertion, the *fillable region* is contracted inward by half of the minimum space from the window

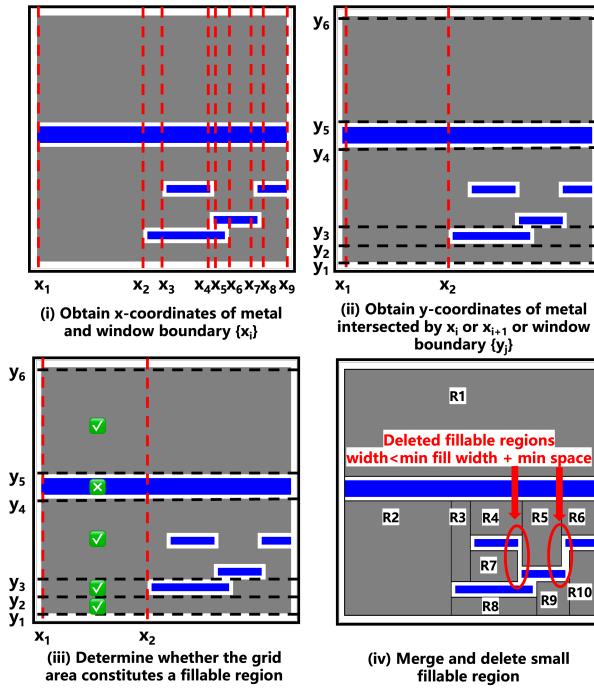


Figure 2: The flow of fillable region extraction.

boundary and expanded outward by half of the minimum space from the metal lines. Furthermore, we allocate half of the minimum space both before and after each dummy fill. In this way, we achieve compliance with minimum space requirements. As illustrated in Fig. 2, where x and y denote the horizontal and vertical directions, the flow of fillable region extraction is as follows: (i) Obtain x -coordinates of metal lines and window boundary, $\{x_i\}$. (ii) Obtain y -coordinates of metal lines intersected by x_i or x_{i+1} or window boundary, $\{y_j\}$. (ii) Determine whether the grid area $(x_i, x_{i+1}, y_j, y_{j+1})$ constitutes a fillable region. (iv) Merge the fillable region and delete the small fillable region whose width or length is less than the sum of the minimum fill width and minimum space.

3.2 Initial Height Prediction and Target Height Assignment

In this work, we first use the ML-based CMP model such as CmpCNN [16] for initial height prediction, thereby obtaining the minimum height for each layer. Subsequently, we assign target heights to individual windows to match the minimum height or the empirical height at a specified density threshold, aligning with density rules and minimizing height variation within each layer.

$$h_{target} = \min(h_e, h_{min}) \quad (7)$$

where the h_{target} , h_e and h_{min} are the target height, empirical height, and minimum height of each layer, respectively.

3.3 MOGA-CmpCNN Fill Insertion

3.3.1 Overview of the Flow of MOGA-CmpCNN. In this work, we view the entire metal fill process as a unified entity, employing multi-objective genetic algorithms (MOGA) integrated with

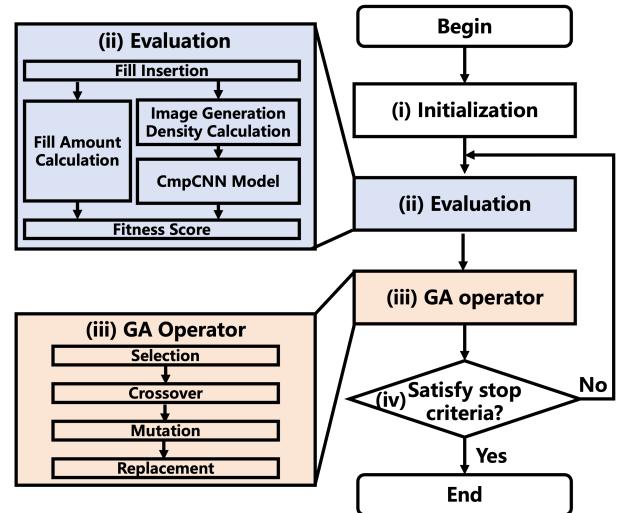


Figure 3: The flow of MOGA-CmpCNN fill insertion algorithm.

CmpCNN model to optimize the fill parameters (*fill width*, *fill length*, and *fill space* in two directions) to obtain an optimal fill solution. Fig. 3 illustrates the detailed flows of our algorithm.

(i) **Initialization:** Generate an initial population of individuals, each characterized by values for fill parameters.

(ii) **Evaluation:** Generate precise fill solutions based on individual fill parameters and evaluate the corresponding fitness score.

(iii) **GA Operator:** Create a new generation of individuals by applying GA operators simultaneously, including selection, crossover, mutation, and replacement.

(iv) **Stop Criteria:** Check if stop conditions are satisfied (e.g., number of generations, achieving satisfactory fitness score, or convergence).

(v) **Stop or Iteration:** If stop conditions are satisfied, the algorithm stops, returning the best fill solution. Otherwise, return to step (ii) and repeat the optimization process.

3.3.2 Multi-Objective Fitness Score. To adhere to density constraints, minimize the post-CMP height variations, and reduce the fill amount, we proposed a multi-objective fitness score. It consists of density score ($Score_d$), height score ($Score_h$) and fill amount score ($Score_f$), which are defined as follows.

$$\begin{aligned}
 Fitness\ Score &= Score_d + Score_h + Score_f, \\
 Score_d &= \begin{cases} c_1 + c_2 \times (d_{target} - d) & \text{when } d < d_{target} \\ d - d_{target} & \text{when } d \geq d_{target} \end{cases}, \\
 Score_h &= c_3 + (h - h_{target})^2, \\
 Score_f &= \begin{cases} 0 & \text{when } \#fill = 0 \\ c_4 \times \log(\#fill) & \text{when } \#fill > 0 \end{cases},
 \end{aligned} \tag{8}$$

where d , d_{target} , h , h_{target} , and $\#fill$ denote the density of the fill solution, the target density, the predicted height of the fill solution, the target height, and the fill amount, respectively. c_1 , c_2 , c_3 , and c_4 are the constant parameters, which will be discussed in Section 4.

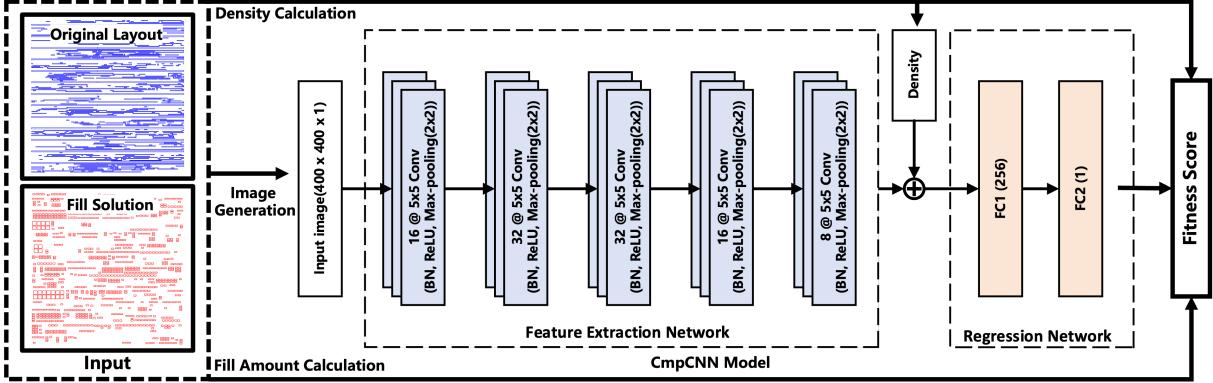


Figure 4: The details of calculating the multi objective fitness score for each fill solution.

Table 1: Layout designs are from ICCAD 2018 contest.

Circuit	# Wires	Size (m^2)	OV (\AA)	MV (\AA)	OD (%)	LC (pF)
1	305667	600×540	40.65	116.61	8.87	9.00
2	750166	1050×1110	140.35	447.99	9.52	21.23
3	64903	270×170	41.23	107.90	8.24	4.04
4	149464	420×210	38.19	94.57	9.38	8.55
5	275425	480×350	39.29	101.07	9.16	16.57

The details of calculating the fitness score for each fill solution are illustrated in Fig. 4. After generating a precise fill solution, image generation and density calculation are applied to generate inputs for height prediction. The image tensor and density are fed into a CmpCNN model to predict the corresponding height. Afterwards, the fitness score is calculated with the density, predicted height, and the fill amount according to Equation (8).

4 EXPERIMENTAL RESULTS

The MOGA-CmpCNN framework is developed in C++ and run on a Linux machine with a 2.50GHz CPU and Nvidia GTX 2080Ti GPU based on the OpenGA library [22] and Pytorch C++ library [23]. The CMP model is developed and evaluated in Python with Pytorch library [23]. Layout designs are from ICCAD 2018 contest [21], consisting of 5 layout designs with 9 layers as reported in Table 1. Additionally, we utilized the official capacitance evaluation tool provided by the organizers of the ICCAD 2018 contest [21] to compute the LC. The original height variations (OV and MV), OD, and LC are also reported for reference. The window size is set as 20 $\mu\text{m} \times 20 \mu\text{m}$. The density of each window after metal fill should be greater than 40%.

4.1 Performance of CmpCNN Model

To encompass a comprehensive dataset range for an accurate model, we first employ a commercial metal fill tool, e.g. Smartfill [24] to generate fill solutions with different fill parameters, such as *fill width* and *fill length*. Then, we perform clip generation [16] to partition layout designs with/without fill solutions into multiple clips with window size of 20 $\mu\text{m} \times 20 \mu\text{m}$. Afterwards, we perform CMP simulation and image binarization to generate data samples to train a model based on CmpCNN [16]. In total, over 10,000 data samples are prepared as training and testing sets. After training

Table 2: Comparisons of different fill patterns on circuit1 of ICCAD 2018 contest dataset.

	OV (\AA)	MV (\AA)	OD (%)	LC (pF)	FA
original	40.65	116.61	8.87	9.00	0
Rectangular	3.19 ($\downarrow 92\%$)	12.49 ($\downarrow 89\%$)	43.52	16.18	1878829
Staggered	4.59 ($\downarrow 89\%$)	16.32 ($\downarrow 86\%$)	43.31	15.99	2021059
$f_1(z)$	4.14 ($\downarrow 90\%$)	12.40 ($\downarrow 89\%$)	43.33	15.96	2082602
$f_2(z)$	4.08 ($\downarrow 90\%$)	13.37 ($\downarrow 89\%$)	43.41	15.82	2104384

process, the normalized root mean square error of our model is 5%. Hence, the trained CmpCNN model is accurate given that the acceptable tolerance margin for CMP model is within $\pm 10\%$ [25].

4.2 Investigation of Different Fill Patterns

To investigate the optimal fill solution, we have conducted comparative experiments to delve deeper into the effectiveness of various fill patterns, including the rectangular fill, staggered fill, and rectangular fill with different DCFs $f_1(i)$ and $f_2(i)$. For a fair comparison, we maintain consistent flow with our MOGA framework, differing only in the fill pattern. Circuit1 is chosen as the representative case within the ICCAD 2018 contest dataset to illustrate the comparisons. As shown in Table 2, rectangular fill showcases the minimum OV of 3.19 \AA and boasts the largest LC of 16.18 pF among the patterns studied. In contrast, staggered fill, rectangular fill with $f_1(z)$ and $f_2(z)$ all achieve reductions in LC, which are 0.19 pF , 0.22 pF , and 0.36 pF , respectively. However, this reduction in LC was accompanied by increases in OV, MV, and FA. The rectangular fill with $f_2(z)$ achieves the smallest LC, with only marginal increments of 0.89 \AA in OV and 0.88 \AA in MV compared to the rectangular fill. As such, the optimal fill pattern should be determined by the specific needs and trade-offs associated with height variations, capacitance, and other relevant factors. The following experiments use rectangular fill with the primary goal of minimizing height variations.

4.3 Effectiveness of Multi-objective Fitness Score

As shown in Equation (8), our proposed multi-objective fitness score consists of density score, height score, and fill amount score. To ensure the fill solution adhere to density constrains and

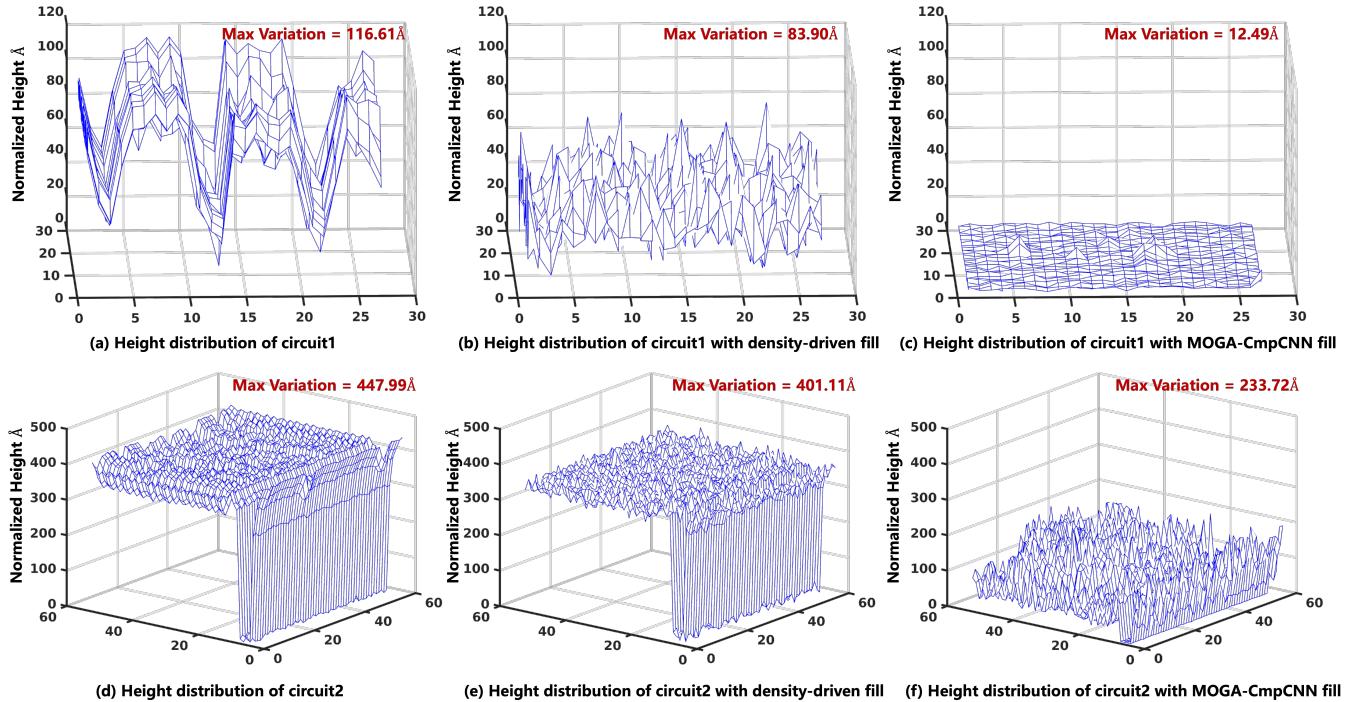


Figure 5: The height distribution comparison of circuit1 and circuit2, including (a) (d) the original circuits, (b) (e) circuits with density-driven fill, and (c) (f) circuits with our MOGA-CmpCNN fill.

Table 3: Comparisons with different c_4 on circuit1 of ICCAD 2018 contest dataset.

	OV (Å)	MV (Å)	OD (%)	LC (pF)	FA
original	40.65	116.61	8.87	9.00	0
$c_4 = 0$	1.98 ($\downarrow 95\%$)	11.83 ($\downarrow 90\%$)	43.03	16.42	3146289
$c_4 = 1$	3.19 ($\downarrow 92\%$)	12.49 ($\downarrow 89\%$)	43.52	16.18	1878829
$c_4 = 2$	3.90 ($\downarrow 90\%$)	14.27 ($\downarrow 88\%$)	43.51	16.17	1776562
$c_4 = 3$	4.49 ($\downarrow 89\%$)	12.46 ($\downarrow 89\%$)	43.45	16.17	1724134
$c_4 = 4$	4.81 ($\downarrow 88\%$)	13.67 ($\downarrow 88\%$)	43.41	15.99	1692236
$c_4 = 5$	5.16 ($\downarrow 97\%$)	14.72 ($\downarrow 87\%$)	43.41	15.59	1665411

minimize height variations, the constant parameters of density score and height score (c_1 , c_2 , and c_3) are set as 100 from our preliminary study. Notice that small fill amount scores have less impact on reducing FA, while large fill amount scores result in greater height variations. We conduct sensitivity analysis on parameter c_4 with different values to identify optimal FA with acceptable height variations trade-off. Table 3 reports comparisons of fill solutions obtained with different fitness scores on circuit1. When c_4 is set to 0, it results in a complete absence of fill amount optimization, resulting in a maximum FA with minimum OV, which represents a 95% reduction compared to the original layout. Setting c_4 to 1 leads to a substantial decrease in FA, reaching only 60% of the FA when $c_4 = 0$. Additionally, the LC decreases by 0.24 pF, while the OV and MV increase by 1.21 Å and 0.66 Å, respectively. However, as c_4 is further increased, the reduction in FA becomes less pronounced, accompanied by a proportional increase in height variations. The OD stabilizes around 43%, as the

Table 4: Comparisons with different optimization algorithms on circuit1 of ICCAD 2018 contest dataset.

	OV (Å)	MV (Å)	OD (%)	LC (pF)	FA
original	40.65	116.61	8.87	9.00	0
SA	4.50 ($\downarrow 89\%$)	11.79 ($\downarrow 90\%$)	44.08	18.09	2793341
DE	4.05 ($\downarrow 90\%$)	13.40 ($\downarrow 89\%$)	43.92	18.38	2529661
GA (Ours)	3.19 ($\downarrow 92\%$)	12.49 ($\downarrow 89\%$)	43.52	16.18	1878829

density score guides the algorithm towards a fill solution that meets the density constraints ($\geq 40\%$) and avoids excessive overfilling. In summary, the proposed multi-objective fitness score achieves a smaller FA and LC while maintaining relatively consistent height variations. For subsequent experiments, c_4 is set to 1, as it achieves significant reduction in FA with only a minor increase in OV and MV.

4.4 Effectiveness of Genetic Algorithm

In this work, we utilize the genetic algorithm (GA) to optimize the fill parameters in a multi-objective and discrete setting. GA is chosen due to its well-established effectiveness in handling discrete and combinatorial optimization problems [26]. To further demonstrate the effectiveness of GA, we conduct comparative analyses with other optimization algorithms, including simulated annealing (SA) and differential evolution (DE), all using the same fitness score. Table 4 presents a comparative performance analysis of different optimization algorithms on circuit1 of the ICCAD 2018 contest dataset. It is observed that GA outperforms other algorithms in OV, OD, LC, and FA. This substantial improvement is

Table 5: Comparisons of density-driven metal fill algorithm and ours on ICCAD 2018 contest dataset.

	Circuit	OV (Å)	MV (Å)	OD (%)	LC (pF)	FA
Density -Driven	1	36.99 (↓ 9%)	83.90 (↓ 28%)	40.36	17.50	2000410
	2	116.53 (↓ 17%)	401.11 (↓ 10%)	41.34	42.61	6687829
	3	38.14 (↓ 7%)	66.72 (↓ 38%)	40.37	7.59	330452
	4	36.83 (↓ 4%)	74.56 (↓ 21%)	40.29	15.50	646784
	5	35.77 (↓ 9%)	66.56 (↓ 34%)	40.30	29.57	1188753
Ours	1	3.19 (↓ 92%)	12.49 (↓ 89%)	43.52	16.18	1878829
	2	54.71 (↓ 61%)	233.72 (↓ 48%)	52.70	50.06	6199051
	3	5.47 (↓ 87%)	27.36 (↓ 75%)	43.47	7.14	316868
	4	2.43 (↓ 94%)	6.98 (↓ 93%)	43.28	14.67	622782
	5	2.54 (↓ 94%)	7.13 (↓ 93%)	43.21	28.22	1166630

attributed to GA’s ability to maintain diversity in the search space while effectively avoiding traps of local optima. In summary, GA is ideally suited for our metal fill optimization task.

4.5 Performance Comparison

To showcase the efficacy of our MOGA-CmpCNN algorithm, we conducted a comparative analysis against density-driven algorithm. To ensure a fair comparison, we kept the flow of the density-driven algorithm identical to ours, with the sole distinction being that the fitness score was computed based on the density score. Table 5 reports the detailed comparisons. Compared to the original layouts, which exhibited *OV* exceeding 38 Å, layouts after density-driven algorithm achieved a reduction in *OV* ranging from approximately 4% to 17%. This indicates a certain superiority of density-driven algorithm in improving layout planarity. Remarkably, our MOGA-CmpCNN algorithm achieves a substantial reduction in *OV*, ranging from approximately 61% to 94%, surpassing the performance of the density-driven algorithm. Besides, our algorithm leads to a reduced *FA* from 2% to 7% compared to the density-driven algorithm. Additionally, our algorithm typically results in smaller *LC* with reductions ranging from approximately 4% to 8%. The *OD* of the density-driven algorithm is marginally lower compared to ours due to its strict adherence to density rules, while ours additionally focuses on minimizing height variation within the same constraints. Circuit2 exhibits a larger *LC* and *OD*, primarily due to significant height variations in the original layout (*MV* = 447.99 Å), necessitating a more extensive fill area for layout planarity. Fig. 5 shows the height distribution of the circuit1 and circuit 2, including the original circuits, circuits with density-driven fill, and circuits with our MOGA-CmpCNN fill. In summary, our MOGA-CmpCNN framework demonstrates outstanding performance in minimizing *OV* and *MV* with smaller *FA* and *LC*.

5 CONCLUSIONS

In this paper, we have proposed the MOGA-CmpCNN framework, leveraging a multi-objective genetic algorithm that integrates a CmpCNN model to effectively enhances layout planarity. We have also designed a multi-objective fitness score to reduce both fill amount and lateral capacitance while maintaining minimum post-CMP height variation. Furthermore, we investigate the impact of various fill patterns on height and capacitance. The

experimental results demonstrate that our proposed MOGA-CmpCNN framework produce a high quality fill solution with minimum height variations and performance degradation.

REFERENCES

- P. B. Zantye *et al.*, “Chemical mechanical planarization for microelectronics applications,” *Materials Science and Engineering: R: Reports*, vol. 45, no. 3, pp. 89–220, June 2004.
- D. O. Ouma, “Modeling of chemical mechanical polishing for dielectric planarization,” Ph.D. dissertation, Massachusetts Institute of Technology, 1998.
- C. Feng *et al.*, “Efficient Approximation Algorithms for Chemical Mechanical Polishing Dummy Fill,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 3, pp. 402–415, 2011.
- R. Tian *et al.*, “Model-based dummy feature placement for oxide chemical-mechanical polishing manufacturability,” in *ACM/IEEE Design Automation Conference (DAC)*, 2000, pp. 667–670.
- X. Wang *et al.*, “A min-variance iterative method for fast smart dummy feature density assignment in chemical-mechanical polishing,” in *Sixth International Symposium on Quality Electronic Design*, 2005, pp. 258–263.
- A. Kahng *et al.*, “Filling algorithms and analyses for layout density control,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 4, pp. 445–462, 1999.
- B. Jiang *et al.*, “FIT: Fill Insertion Considering Timing,” in *ACM/IEEE Design Automation Conference (DAC)*, ser. DAC ’19, 2019.
- T.-C. Luo *et al.*, “A novel design flow for dummy fill using Boolean mask operations,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 25, no. 3, pp. 468–479, 2012.
- Y. Lin *et al.*, “High Performance Dummy Fill Insertion with Coupling and Uniformity Constraints,” in *ACM/IEEE Design Automation Conference (DAC)*, 2015.
- S. Sinha *et al.*, “Model based layout pattern dependent metal filling algorithm for improved chip surface uniformity in the copper process,” in *Asia and South Pacific Design Automation Conference*, 2007, pp. 1–6.
- J. Cai *et al.*, “A Novel and Unified Full-Chip CMP Model Aware Dummy Fill Insertion Framework With SQP-Based Optimization Method,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 3, pp. 603–607, 2021.
- J. Cai *et al.*, “NeurFill: Migrating full-chip CMP simulators to neural networks for model-based dummy filling synthesis,” in *ACM/IEEE Design Automation Conference (DAC)*, 2021, pp. 187–192.
- R. Ghulghazaryan *et al.*, “Application of Machine Learning and Neural Networks for Generation of Pre-CMP Profiles of Advanced Deposition Processes for CMP Modeling,” in *Proceedings of International Conference on Planarization/CMP Technology (ICPT)*, October 2017, pp. 1–6.
- R. Ghulghazaryan *et al.*, “Application of Neural Network-Based Oxide Deposition Models to CMP Modeling,” *ECS Journal of Solid State Science and Technology*, vol. 8, no. 5, pp. 3154–3162, 2019.
- H. Bao *et al.*, “A CNN-Based CMP Planarization Model Considering LDE Effect,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 10, no. 4, pp. 723–729, 2020.
- Q. Zhang *et al.*, “CmpCNN: CMP Modeling with Transfer Learning CNN Architecture,” *ACM Transactions on Design Automation of Electronic Systems*, vol. 28, no. 4, pp. 1–18, 2023.
- L. He *et al.*, “Variability-driven considerations in the design of integrated-circuit global interconnects,” in *VLSI Multilevel Interconnection (VMIC)*, 2004, pp. 214–221.
- X. Dong *et al.*, “New metal fill considerations for nanometer technologies,” in *International Conference on ASIC*, vol. 2, 2005, pp. 802–805.
- A. B. Kahng *et al.*, “CMP Fill Synthesis: A Survey of Recent Studies,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 1, p. 3–19, 2008.
- P. Morey-Chaisemartin *et al.*, “CMP monitoring and prediction based metal fill,” in *International Symposium on Quality Electronic Design*, 2011, pp. 1–6.
- B. Yang *et al.*, “ICCAD-2018 CAD contest in timing-aware fill insertion,” in *Proc. International Conference on Computer-Aided Design (ICCAD)*, 2018.
- A. Mohammadi *et al.*, “OpenGA, a C++ genetic algorithm library,” in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 2051–2056.
- A. Paszke *et al.*, “PyTorch: An imperative style, high-performance deep learning library,” in *Proceedings of Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.
- M. Graphics, *Calibre YieldEnhancer User’s Manual*, Mentor Graphics, 2021.
- C. Hui *et al.*, “Hotspot detection and design recommendation using silicon calibrated CMP model,” in *Design for Manufacturability through Design-Process Integration III*, vol. 7275, March 2009, pp. 495–506.
- M. Shahsavar *et al.*, “Statistical design of genetic algorithms for combinatorial optimization problems,” *Mathematical Problems in Engineering*, vol. 2011, 2011.