IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 5, 2019, accepted November 16, 2019, date of publication November 28, 2019,
date of current version December 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2956595

DOTFloor–A Diffusion Oriented Time-Improved Floorplanner for Macrocells

SOMNATH ROY CHOUDHURY AND SAMBHU NATH PRADHAN
Department of ECE, National Institute of Technology Agartala, Agartala 799046, India

Corresponding author: Somnath Roy Choudhury (som.vlsi@gmail.com)

This work has been carried out in the research facility established by the fund received from the Ministry of Electronics & Information Technology (MeitY), Government of India under the Special Manpower Development Program for Chip to System Design (SMDP-C2SD) Project at the Department of ECE, National Institute of Technology Agartala, India.

ABSTRACT This work presents a unique time-efficient and reliable floorplan algorithm DOTFloor (Diffusion Oriented Time-improved Floorplanner), built around a SA (Simulated Annealing) engine and targeted to optimize the peak on-chip temperature along with the traditional design metrics like chip area and wire length. This paper also proposes a novel heat-diffusion based stochastic thermal model called the FATT (Fast Assumption Technique for Temperature) which provides a fast assumption of the degree of hotness during the optimization process. The incorporation of FATT in DOTFloor results in a significant improvement in the run time of the optimization process. Upon experimentation on MCNC (Microelectronics Center of North Carolina) benchmark circuits with the proposed floorplanner, a good optimization in area, wire length metric and peak on-chip temperature with a significant reduction in execution time have been achieved over the existing floorplanning tool, the HotFloorplan.

INDEX TERMS Floorplanner, heat-diffusion, MCNC, optimization, simulated annealing.

I. INTRODUCTION

In the realm of thermal management in processors, floorplanning plays a key role at the physical design level of VLSI technology. Thermal-aware floorplanning facilitates the mitigation of on-chip hotspots and temperature gradients along with the optimization of other design metrics like area, wire length, routability, delay, etc. Besides, the reliability and time efficiency of such algorithms are very critical in the arena of CAD tool development. The floorplanner for its operation requires Models of Computations (MOCs) for the estimation of its metrics like temperature, area, interconnect delay, etc. Improvement in reliability and time efficiency of the floorplanners can be achieved at the algorithmic stage and/or at the level of MOCs. For example at the MOCs level, a major chunk of time is generally consumed for the temperature estimation and accurate techniques like finite element method (FEM) [1], finite difference method (FDM) [2], Green's function method [3] and compact thermal model [4], [6] require considerable time for the computation of temperature. Several efforts have been made to improve the performance of floorplanners and some of them have been discussed below.

A Genetic Algorithm (GA) based thermal-aware floorplanner has been presented in [7], which optimizes the chip area and on-chip temperature of slicing floorplan structure. A thermal-aware SA based floorplanning tool called HotFloorplan has been introduced by [8], which optimizes the peak on-chip temperature and performance indicated by the wire length delay for slicing floorplan. Authors in [9] present a hybrid PSO-GA algorithm for minimizing chip area, wire length, and temperature of non-slicing floorplan arrangement. Authors in [10] present a thermal-aware floorplanner for slicing floorplan by modeling the temperature-dependent wire delay, routing congestion and reliability factors and including the same in addition to the chip area and temperature metrics in the cost function of the HotFloorplan tool. The work shown in [11] presents an Application Specific NoC synthesis approach involving floorplan which targets to optimize the chip area, peak on-chip temperature and communication cost. The algorithm has been developed in Mixed Integer Linear Programming and Simulated Annealing. A Simulated Annealing based pre-RTL tool framework for floor-planning has been developed in [12] using the python language to optimize chip area and the peak temperature of SoC and chip multiprocessors. All of the floorplanners presented in [7]– [12] invoke the HotSpot tool [4], [5] for the temperature estimation. HotSpot tool is accurate but it takes a considerable time for solving the temperature. Moreover, a thermal-aware placement algorithm has been presented in [13] to optimize the peak on-chip temperature, temperature

The associate editor coordinating the review of this manuscript and approving it for publication was Fuhui Zhou.

gradient, and the wire length. Here temperature has been estimated from a thermo-resistive network of the chip and by using the modified nodal analysis on it, which again incurs a considerable computation overhead. Authors in [14] also present a fast but accurate temperature estimation technique by adopting an iterative conjugate gradient method as an alternative to the LU decomposition for solving the HotSpot tool thermal model. The technique presented in [14] still consumes considerable time when included for the complete floorplan optimization process. Authors in [15] have presented a floorplan strategy built around Satisfiability Modulo Theory to handle hard as well as flexible functional blocks. In this work, optimization has been done for the chip area alone. The work in [16] shows a Genetic Algorithm (GA) based floorplanner for optimizing the chip area and the wire length of free as well as fixed outline floorplans. Here the performance of the GA has been improved by including an entropy function that takes care of enhancing the diversity of solutions for the sake of obtaining the global optimum instead of settling at a local minimum. Paper [17] presents a novel floorplan algorithm based on Particle Swarm Optimization. The proposed floorplanner targets to achieve optimization in the chip area and the wire length. Work presented in [18] reveals a fast fixed outline multilevel floorplanner based on a thermal-aware non-linear model to simultaneously optimize the chip temperature and wire length. The floorplanner uses a power blurring analytical method to estimate the temperature and gives improved floorplan as well as runtime over the Corblivar floorplanning tool. The time overhead of exact temperature computation has been avoided in [19] and authors have proposed a thermal-aware placement algorithm for standard cells based on game theory to optimize the chip temperature. To achieve an improvement in time, authors in [20] reduced the complexity involved in temperature computation by developing a stochastic heat-diffusion model for the temperature estimation. Further a Simulated Annealing based temperature-aware floorplaner has been constructed in [20] by incorporating chip-temperature to the objective function of the Parquet floorplanning tool presented in [21]. Here optimization has been obtained for chip area, wire length, and on-chip temperature. The author in [22] presents two fast thermal-aware algorithms, one greedy and the other SA-Greedy hybrid algorithm for optimizing area and temperature of nonslicing floorplan. Like [20], the complexity of temperature computation was also avoided in [22]. Instead of any temperature quantification in the objective function, hotspot minimization has been mechanized in [22] by forming groups containing one hot and three non-hot modules in each and further placing them in floorplan, such that a hot module remains surrounded by the three non-hot modules of its group. Authors in [23] have considered the area and peak power density (of a thermal zone) as metrics of optimization in floorplanning, but peak power density alone cannot account for the degree of hotness in a chip. The techniques presented in [20], [22], [23] are fast but do not consider the influence of the adiabatic die-boundary on the thermal characterization of a chip.

Our work exhibits a thermal-aware floorplanner and a fast novel heat diffusion based stochastic thermal model which avoids the complexity overhead of accurate temperature computation (incurred by some of the previous works) during the optimization process and also includes the adiabatic die-boundary influence (ignored by earlier works). The proposed thermal-aware floorplanner achieves a significant runtime improvement while maintaining equivalent qualitative outputs compared to the floorplanning tool, the HotFloorplan.

Our paper renders the following salient contributions –

(i) It presents a unique thermal-aware floorplanner called the DOTFloor (Diffusion Oriented Time-improved Floorplanner), developed around the classical Simulated Annealing (SA) algorithm [24]–[26] to handle slicing floorplan structures composed of fixed as well as rotatable and hard functional-blocks. Utilizing a lateral heat spread management by the redistribution of functional block locations on the chip floor; the floorplanner gives successful optimization of the peak on-chip temperature as well as the chip area and the wire length metric with a considerable relative improvement in the execution time.

(ii) It introduces a novel thermal model called FATT (Fast Assumption Technique for Temperature) for obtaining an assumption of the degree of hotness by modifying the heat diffusion function presented in [20] and [27], instead of exactly computing the temperature thereby reducing the computational complexity in thermal characterization. FATT approximates the local and peripheral (die boundary) heat-diffusion effects to give a fast and good assumption of the hotness of the chip. The inclusion of FATT in the DOTFloor is responsible for the relative improvement in time efficiency for the floorplanner.

(iii) It defines a weighted cost function that evaluates the fitness of a floorplan solution and also facilitates the adjustment of tradeoffs between the cost metrics for the attainment of the final optimized solution.

II. MOTIVATION OF THE WORK

The thermal model presented by Han and Korean in [20] and [27] facilitates a fast thermal characterization of floorplan structures. According to Han and Korean, in a floorplan, the heat diffusion Q_i of a block *Blk i* with its neighboring blocks *Blk j*, having power densities *pdi* and *pdj* respectively and mutually sharing a boundary of shared length SL_{ij} is given by,

$$Q_i = \sum_j Q\,(pdi, pdj) = \sum_j (pdi - pdj) \times SL_{ij} \quad (1a)$$

Similarly, the total heat diffusion Q for the set of all functional-blocks $\{Blk\ i\}$ in the floorplan is given by,

$$Q = \sum_i Q_i \quad (1b)$$

This thermal model assumes that maximizing Q leads to an increase in heat diffusion and it subsequently results in a minimization of the chip-temperature.

Now, as an example, let us consider two gate-array floorplans of 36 functional-blocks each, arranged in a 6x6 matrix arrangement as shown in Fig. 1a and Fig. 1b, where every rectangle denotes a functional-block with a block number label in it. Let the height and width of each block be 1652 μm and 1218 μm respectively. Power of blocks Blk15, Blk21 and Blk22 are considered to be 4829126.4 μW each and zero for the rest of the 33 blocks. By fitting the considered parameters in (1a) and (1b), Table 1 has been constructed. The temperature profiles in Fig. 1 show that Floorplan-2 has a higher peak temperature (PT).

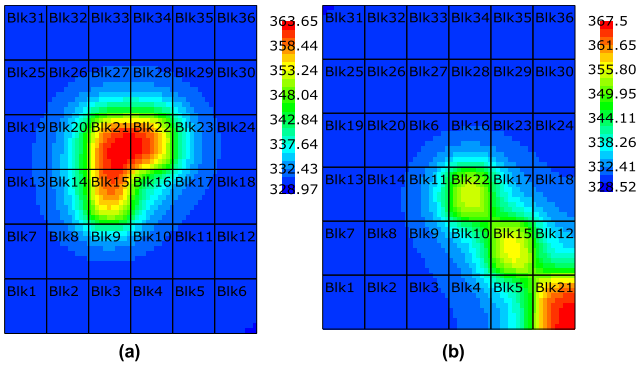

FIGURE 1. Temperature profile (in Kelvin) of Gate array (a) Floorplan-1 (b) Floorplan-2.

TABLE 1. Thermal characterization of Floorplans according to the thermal model in [20] and [27].

Sample	Q_i (W . m^{-1})	Q (W . m^{-1})	PT (K)
Floorplan-1	Q_{15} = 10852.8 Q_{21} = 6888 Q_{22} = 9811.2	27552.0	360.98
Floorplan-2	Q_{15} = 13776 Q_{21} = 6888 Q_{22} = 13776	34440.0	363.04

In Table 1 the terms Q_{15}, Q_{21}, and Q_{22} denote the heat diffusion values of blocks Blk15, Blk21, and Blk22 respectively according to 1(a). Term Q denotes the total heat diffusion according to (1b). Table 1 shows that Floorplan-2 has higher heat diffusion value Q, and according to the assumption of the thermal model of Han and Korean, it is supposed to have lesser peak temperature (PT) than Floorplan-1. But from Table 1 and Fig. 1 it is observed that Floorplan-2 has higher peak temperature than Floorplan-1 (obtained from simulation results) and it contradicts the assumption of the thermal model of Han & Korean. Moreover, the hottest cell Blk21 has equal Q_i values in both the floorplans but still has a higher temperature in Floorplan-2 which is unexplainable by (1a). The discrepancy arises because the die-boundary is actually adiabatic in nature and the thermal model does not include the adiabatic die-boundary influence on the temperature.

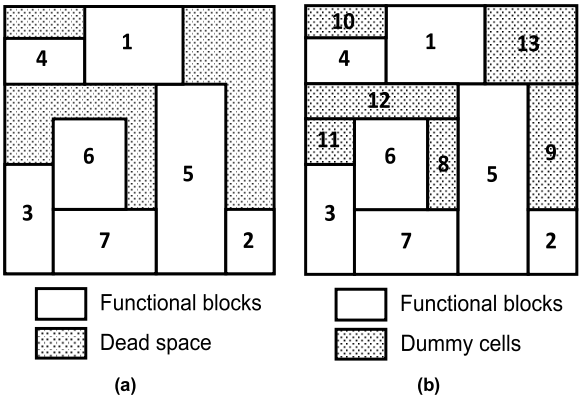

FIGURE 2. (a) Slicing Floorplan. (b) Completely sliced structure.

It can be observed from the compact thermal model of Fig. 2 in [6], that every heat-generating node (excluding those on the periphery) of the die has four lateral and one vertical conductive heat flow path. All the heat flow paths at a node help in draining out the heat from it. As a heat source location approaches closer to the periphery, the number of lateral and the vertical heat flow paths surrounding the source goes on decreasing. Finally, at the periphery location, one lateral conductive heat flow path is blocked and at the corner, two heat flow paths are blocked resulting in a reduced lateral heat spread or increased heat accumulation like an adiabatic system. As a result, the heat sources closer to the die periphery witness higher temperatures. Blk21 is hotter in Floorplan-2 because it is relatively nearer to the die-boundary than in Floorplan-1. Therefore, we modify the diffusion thermal model presented in [20] and [27] by including the adiabatic behavior of the chip-boundary, for the improved fast thermal characterization of floorplan structures.

III. MODELS OF COMPUTATIONS (MOCs)

A. PROPOSED THERMAL MODEL, FATT (FAST ASSUMPTION TECHNIQUE FOR TEMPERATURE)

1) DEFINITIONS RELATED TO FATT

(a) Dummy cells: These are the rectangular cells constructed out of the dead spaces and have zero power density. The dummy cells are obtained by converting a slicing floorplan (Fig. 2a) into a completely sliced structure (Fig. 2b). The shaded rectangular blocks in Fig. 2b are the dummy cells.

(b) Neighbor cells: The functional blocks or dummy cells which share the geometrical boundary with a cell-i are defined as the neighbor cells of cell-i.

(c) Critical cells: These are the high power density functional blocks with power density pdi defined as $pdi \geq pd_M + pd_{MD}$, where pd_M is the mean power density and pd_{MD} is the mean deviation in power density of the functional blocks.

(d) Max-cell: Considering all the functional blocks of heights $\{Hi\}$, widths $\{Wi\}$ and power densities $\{pdi\}$, Max-cell is an imaginary reference cell considered to have the maximum height $H = \max\{Hi\}$, maximum width $W = \max\{Wi\}$ and maximum power density $P = \max\{pdi\}$.

2) ASSUMPTIONS OF THE MODEL

(a) FATT assumes that critical cells are responsible for the creation of hotspots. Hence the model focuses only on the critically hot cells for the thermal metric evaluation. (b) FATT considers the cumulative heat diffusion from the critical cells, as a representation of the degree of hotness in the chip. More the diffusion lesser is the degree of hotness. (c) For computational simplicity, FATT visualizes the hotness of a critical cell to be majorly influenced by the heat diffusion between the critical cell and its neighbor cells and secondly by their (critical cell) proximity from the die wall. (d) FATT utilizes the dependence of heat diffusion on the location of heat source on the die floor. The vertical heat diffusion being independent of the location in die, the model considers lateral heat diffusion only along the orthogonal x and y directions. (e) The die boundary is assumed to be adiabatic in nature and behaves as a reflector to the heat flux coming from the functional blocks. (f) The die boundary is not a power generating source and every point on the die boundary is considered to have zero power density. (g) FATT considers two thermal processes viz. Neighbor diffusion and Heat accumulation discussed later. (h) FATT uses the Fourier's heat flow equation to model the heat diffusion processes.

3) NEIGHBOR DIFFUSION

FATT considers the lateral diffusion of heat between a critical cell and its neighbor cells according to the Fourier's heat equation. The rate of heat flow q between two points of a conductor, Δx distance apart (along the x-axis), perpendicular to cross-sectional area A_x, having a temperature gradient of $\Delta T/\Delta x$ and thermal conductivity K, according to Fourier is given by, $q = KA_x(\Delta T/\Delta x)$. Since temperature is directly proportional to the power density it follows; $\Delta T \propto \Delta pd$ where ΔT and Δpd are respectively the temperature and power density differences between the two heat exchanging points. So it follows that,

$$q = constant \times KA_x(\Delta pd/\Delta x) \tag{2a}$$

Now we consider two heat exchanging cells (i and j) as shown in Fig. 3a, facing each other to have (a) power densities pdi and pdj respectively so that $\Delta pd = pdi - pdj$, (b) common superposable edge length Lij, so that the common superposable cross-sectional area for heat flow is $A_x = Lij \times t$, (c) perpendicular distance dij between the facing edges (of cell i and j) so that heat diffusion occurs through a channel of length $\Delta x = dij + \delta$ and cross-section A_x. Here t is the thickness of the die and δ is a very small length element included to avoid singularity of (2a) in cases when the two cells come in contact resulting in $dij = 0$. So the rate of heat diffusion qij between two cells i and j according to (2a) is given by,

$$qij = constant \times KLij \times t\,\{(pdi - pdj)\,/\,(dij + \delta)\} \tag{2b}$$

Considering heat diffusion among the neighbor cells (i.e. cells in contact with $dij = 0$) only and since parameters

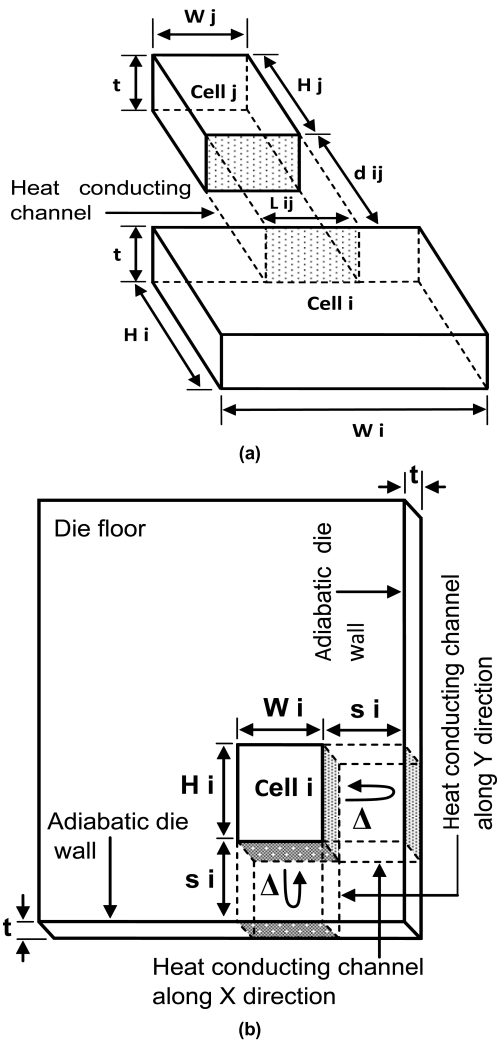

FIGURE 3. (a) Lateral heat diffusion channel between two cells. (b) Heat Δ produced from hot cell-i getting reflected from the die walls back to the cell through the heat channels in x and y directions.

K and t are constants, (2b) becomes, $qij = constant \times Lij\,\{(pdi - pdj)\,/\delta\}$. Hence the total rate of heat diffusion $Diff_i$ from a critical cell-i to all its j-neighbors is given by,

$$Diff_i = \sum_j qij = constant \times \sum_j Lij\,\{(pdi - pdj)\,/\delta\} \tag{2c}$$

4) HEAT ACCUMULATION

FATT further assumes that a component of heat generated from the critical functional block travels laterally (along x and y directions) up to the adiabatic die boundary walls and gets reflected back to the critical block at the same rate. This leads to a reflective heat accumulation in the critical block resulting in a rise of temperature. The heat reflection is accounted from the two adjacent die walls which are nearer to the critical cell. The process has been illustrated in Fig. 3b. The rate of heat reflection or rate of heat accumulation (Ri) in cell-i is given similar to (2b) by, $Ri = constant \times KDi \times t\,\{(pdi - pdj)\,/\,(Si + \delta)\}$. As shown in Fig. 3b, Di is the length of the edge (i.e. Height Hi or width Wi) of the critical

4	3	10	V	1	V	2	V	8	V	6	5	V	7	V	9	H	V	H

FIGURE 4. A random NPE representing a floorplan of Xerox benchmark circuit. The numbers {*i*} in the NPE represent the operands or the functional blocks {*Blk i*} and the letters V, H represent the cutline operators.

cell-*i* facing the reflector walls, *pdi* is the power density of the critical cell-*i* facing the reflecting die boundary wall and *Si* the perpendicular distance between the corresponding cell edge and the die wall (along x or y directions). In this case heat diffusion occurs through a conducting channel of length $Si + \delta$ and cross-section $Di \times t$. The power density of reflecting point on the die wall is $pdj = 0$ and K, t being constants,

$$Ri = constant \times Di \{pdi/ (Si + \delta)\} \tag{3a}$$

As the critical block approaches closer to the die walls the reflective heat accumulation increases, leading to a rise in temperature and the same can be observed from (3a). In (3a) it is observed that with an increase in the closeness of a critical block with the die wall the parameter *Si* decreases, resulting in an increase in the heat accumulation component *Ri*. The total reflected or accumulated heat component for the critical cell-*i* is given by,

$$Acc_i = (Ri)_{along\,x} + (Ri)_{along\,y} \tag{3b}$$

5) THERMAL METRIC

Maximum neighbor diffusion (*max_diff*) occurs for the Max-cell when all its neighbor cells are dummy and in such a case form (2c) it is obtained that, $max\_diff = constant \times 2(H + W)P/\delta$. Maximum heat accumulation (*max_acc*) for Max-cell occurs when it is at a corner of the die-boundary (such that $Si = 0$ along x and y directions) and from (3a), (3b) it is observed that, $max\_acc = constant \times (H + W)P/\delta$. FATT formulates a net heat-diffusion term Φ_i for critical cell-*i*, consisting of the corresponding normalized diffusion and accumulation terms. The terms have been normalized so that they may not overpower each other's values.

$$\Phi_i = \left(Diff_i/max\_diff\right) - (Acc_i/max\_acc) \tag{4a}$$

Since the net diffusion Φ_i, increases with the increase in neighbor diffusion and decreases with the increase in heat accumulation, the first term is positive and the second term is negative in the RHS of (4a). For the entire floorplan, the thermal metric (TM) for all *i*-hot critical cells, is given by,

$$TM = \sum_i \Phi_i \tag{4b}$$

Finally, FATT assumes that maximizing the thermal metric *TM* will increase the net normalized heat-diffusion and consequently will reduce the on-chip temperature.

B. WIRE LENGTH MODEL

The total wire length metric (WLM) for N functional blocks or cells in a floorplan is computed according to the wire length model given in [8] as,

$$WLM = \sum_i \sum_j Aij \times \rho ij; \quad \text{for } i,\ j = 1\ to\ N \tag{5}$$

Here *Aij* is the Manhattan distance between centers (*xi, yi*) and (*xj, yj*) of blocks *Blk i* and *Blk j* respectively and ρij is the density of connections between them. This model collects the cell coordinates and dimensions from the Area Model, finds ρij, as well as computes *Aij* as, $Aij = |xi - xj| + |yi - yj|$. Also, for an optimally segmented wire, the total wire delay can be obtained from the product of WLM with the delay per unit length ξ, according to [8] and [28]. From [8] and [28] it can be derived that $\xi = l_{crit} rc \left(2a + \sqrt{2ab}\right)$, where l_{crit} is the optimal length of a wire section, r and c are the resistance and capacitance per unit length of wire. The parameters l_{crit}, r and c depend on the technology node and the metal layer. As per [28] the parameters $a = 0.4$ and $b = 0.7$ for 50% swing.

C. AREA MODEL

In this work, an area model has been developed to handle fixed as well as rotatable and hard block floorplans. The model has been built according to the methodology proposed in [29] and shape function arithmetic as shown in section 8.2 of [30]. A slicing floorplan can be represented by a skewed slicing tree with the functional blocks present as leaves at the lowest level, and the complete floorplan being represented by the daughter composite block to be formed at the topmost level or root node of the tree. Corresponding to every slicing tree the optimal orientation of the functional blocks is determined in two steps, viz. Merger of corner points and Backtrack.

Step1 Merger of corner points: Every rectangular functional block or cell has an associated shape function or graph composed of corner points whose coordinates represent the possible height and width of the block. At every block-merging step, shape graphs of two parent blocks merge on the basis of the cutline operator to form the shape graph of the daughter composite block. Different combinations of corner point pairs with one from each parent block are considered to merge and form the corner points of the daughter shape graph. Redundant points from the list of corner points of the daughter block so formed are removed as per the method shown in section 8.2 of [30]. The process of stepwise shape graph merging continues until the shape graph of the final composite daughter block representing the entire floorplan is obtained. As an example, the stepwise shape graph formation for the normalized polish expression (NPE) of Fig. 4, has been illustrated in Fig. 5 where Fig. 5a to Fig. 5i represent the

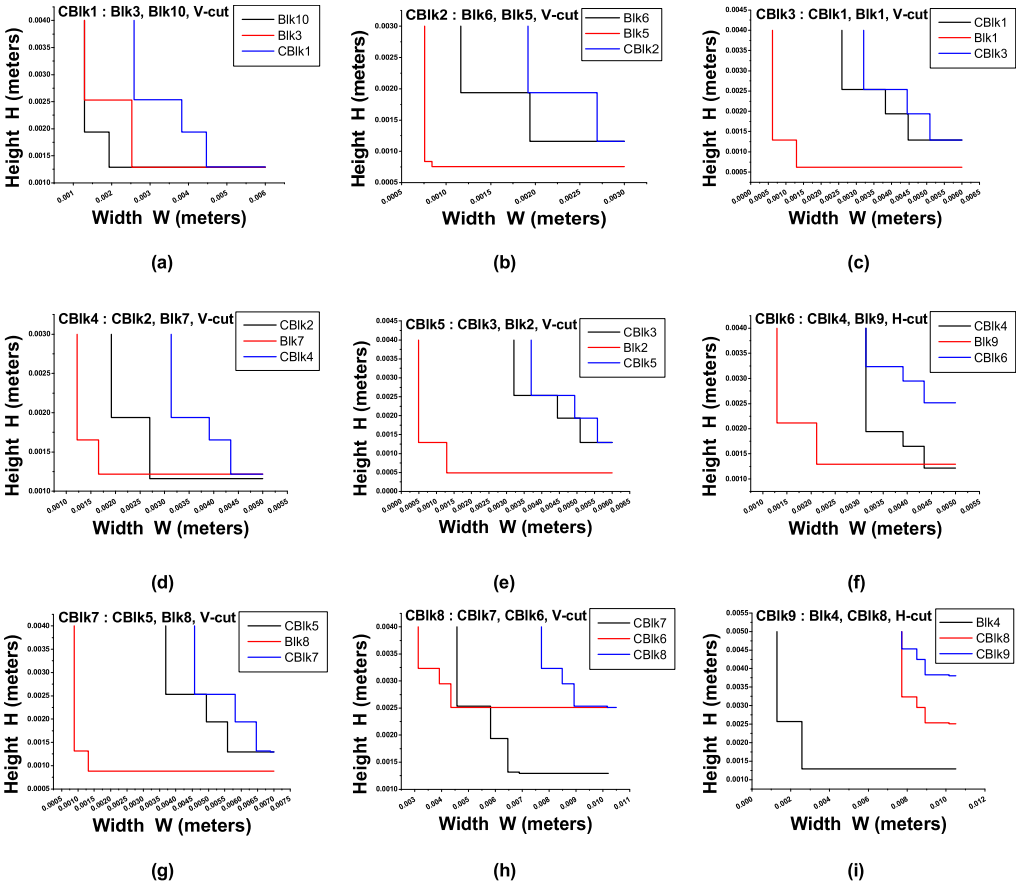

FIGURE 5. Stepwise shape graph formation of the higher Composite Blocks (CBlk) as per NPE of Fig. 4, upon merger of parent blocks defined by the cutline operator. Fig. 5a to Fig. 5i represents the gradual formation of shape graph of the final composite block of the floorplan.

different stages of block merger. It starts with Fig. 5a where shape graphs of parent blocks Blk3 and Blk10 merge in V-cut to form the graph of composite daughter block CBlk1 and finally in Fig. 5i the shape graph of the final composite block CBlk9 or the root node is obtained by the shape graph merger of the parent blocks Blk4 and CBlk8 in H-cut.

Step2 Backtrack: In order to find the minimum floorplan area, the corner point giving the least area of the root node is identified from its shape graph. In Fig. 5i the third corner point from the top in the shape graph of CBlk9 is the suitable corner point giving a minimum area of the root node CBlk9. The problem now is to backtrack and find the chain of corner points of the parent blocks at every stage and finally end up with obtaining the particular corner points of the functional blocks which are associated with the formation of the suitable corner point of the root node CBlk9. In the given example the backtrack-process scans backward from Fig. 5i to Fig. 5a. The optimal height and width of the functional blocks or the leaf cells are obtained from their respective finalized corner points after the backtrack-process.

For fixed block floorplan, the functional block dimensions stay unaltered as the original dimensions. Hence, in this case, every block has only one corner point and thus in the shape graph implementation, only the first step i.e. Merger of corner points is performed to obtain the floor area. The backtrack-operation is not required. But in case of floorplan with rotatable blocks, both steps of Corner point merger and Backtrack are performed. Table 2 shows the results of shape graph

TABLE 2. Comparison of functional block dimensions and chip area associated with the NPE of Fig. 4 for fixed and rotatable block Floorplans implemented by shape graph arithmetic.

Block	Fixed Block Floorplan			Rotatable Block Floorplan		
	Original dimensions		Chip dimension and Area	Optimal dimensions		Chip dimension and Area
	H (10^{-3} m)	W (10^{-3} m)		H (10^{-3} m)	W (10^{-3} m)	
Blk1	1.3	0.6	Chip Height = 0.006622 m Chip Width = 0.009198 m Area = 6.0909 x 10^{-5} m^2	0.6	1.3	Chip Height = 0.003829 m Chip Width = 0.008925 m Area = 3.4174 x 10^{-5} m^2
Blk2	1.3	0.5		0.5	1.3	
Blk3	2.5	1.3		2.5	1.3	
Blk4	1.3	2.6		2.6	1.3	
Blk5	0.8	0.8		0.8	0.8	
Blk6	1.2	1.9		1.9	1.2	
Blk7	1.2	1.7		1.7	1.2	
Blk8	1.3	0.9		1.3	0.9	
Blk9	1.3	2.1		2.1	1.3	
Blk10	1.9	1.3		1.9	1.3	

implementation for the normalized polish expression (NPE) of Fig. 4 considering both fixed and rotatable block floorplans. In Table 2 the parameters H and W denote the functional block heights and widths respectively. It becomes clear from Table 2 that incase of floorplan with free blocks, the shape graph arithmetic efficiently incorporates the block rotation to achieve the optimal block dimensions and subsequently ensures the minimum area for every NPE. Shape graph implementation for fixed blocks in Table 2 manifests that fixed block floorplan has higher chip area. In both cases of fixed and rotatable block floorplans, the process initiates with the original block dimensions. The coordinates of the functional blocks in the chip floor are computed only after ensuring the final block dimensions for both cases of rotatable and fixed block floorplans. The block dimensions and the coordinates so obtained by the area model are also used by the wire length and the thermal models.

IV. PROPOSED FLOORPLANNING ALGORITHM DOTFLOOR (DIFFUSION ORIENTED TIME-IMPROVED FLOORPLANNER)

A. PROBLEM DEFINITION

Given a set of N hard rectangular functional blocks FB $= \{Blk\ i\}$, with $i = 1, 2 \ldots N$ and each $Blk\ i$ having,

- Width $= Wi$, Height $= Hi$,
- Associated interconnection set $\{\rho ij\}$ denoting the number of interconnections between $Blk\ i$ and remaining functional blocks $Blkj \epsilon$FB,
- Power density $= pdi$ and Power $pi = pdi \times Wi \times Hi$,

the problem is to obtain an optimal floorplan by arranging the functional blocks on the chip floor such that the overall area, wire length metric (WLM) and the peak on-chip temperature of the floorplan are optimized. The solution is to be obtained once by considering the blocks $\{Blk\ i\}$ to be of fixed orientation (Aspect ratio $\gamma i = Hi/Wi$) and again by considering the blocks $\{Blk\ i\}$ to be of free orientation i.e. rotatable by 90^o (Aspect ratio $\gamma i = Hi/Wi$ or $\gamma i = Wi/Hi$).

The proposed floorplan algorithm DOTFloor has been built around a classical Simulated Annealing engine (SA) given in [24], [25] and sections 2.4.4 and 3.3.2 of [26] and the salient steps adopted in the algorithm have been listed below.

B. FLOORPLAN ENCODING

The proposed algorithm DOTFloor encodes a slicing floorplan by the normalized polish expression (NPE) [31].

C. COST FUNCTION

The proposed floorplanner DOTFloor incorporates the Models of Computations (MOCs) viz. area model, wire length model and the thermal model FATT (Fast Assumption Technique for Temperature) to operate over and characterize every encoded floorplan to record its Area, wire length metric (WLM) and temperature metric (TM). DOTFloor then evaluates the fitness of the floorplan solution by the normalized weighted Cost function as,

$$Cost = W_1 \times \frac{Area}{Initial\_Area} + W_2 \times \frac{WLM}{Initial\_WLM} + W_3 \times \frac{Initial\_TM}{TM} \quad (6)$$

Parameters *Initial_Area*, *Initial_WLM* and *Initial_TM* are the corresponding three cost metrics obtained for a randomly generated initial solution. Parameters W_1, W_2, W_3 are the weights for refining the area, wire length metric (WLM) and thermal metric (TM) respectively. Each weight parameter W_k varies between 0 and 1 such that $\sum_k W_k = 1$. Fitness of a solution decreases with the increase in its cost. Since the floorplan degrades with the increase in Area and wire length metric (WLM), they occur in numerators and as the floorplan improves with the increase in thermal metric (TM), it occurs in the denominator of the cost function in (6). Since the cost metrics are dimensionally different and value of one metric may overpower other metrics, they have been normalized.

D. PERTURBATION

The stimulus to create a new floorplan solution is achieved by the DOTFloor, via any of the three perturbation mechanisms on a normalized polish expression (NPE): (a) Swap of randomly chosen adjacent operands or functional blocks, (b) Swap of randomly selected operand with its adjacent cut line operator, (c) Inverting cutline operators in a randomly chosen cutline or operator chain. Block rotation is not applied as a perturbation as it is taken care of during the shape curve optimization discussed in section III(C).

A new solution is created by the perturbation of the present floorplan solution. The cost difference between the new solution *New_sol* and present solution *Present_sol*, is given by $\Delta h = Cost\ (New\_sol) - Cost(Present\_sol)$. The new solution is better if $\Delta h < 0$ and is inferior if $\Delta h > 0$.

E. METROPOLIS ACCEPTANCE CRITERIA

The following probability criterion has been adopted for accepting a new solution generated upon perturbation.

1) The probability P for accepting a superior solution ($\Delta h < 0$) is given by $P = 1$.
2) The probability P of accepting an inferior solution ($\Delta h > 0$) at annealing temperature T is given by,

$$P = exp(-\Delta h/T) \quad (7)$$

An inferior solution is accepted if the probability P defined in (7) is greater than a randomly generated number varying between 0 and 1.

F. STOPPING CRITERIA

DOTFloor adopts the following conditions for stopping the optimization process.

1) LOCAL STOPPING CRITERIA

It is the criteria of annealing cycle due to which it stops when any of the conditions are met: (a) Number of iterations in the

annealing cycle exceeds the iteration limit $local\_iter = 2Nk$. (b) Number of probabilistically accepted inferior solutions exceeds the limit $\sigma = Nk$. Here N is the number of functional blocks and k is an integral value to be called as the local iteration multiplier. An increase in N ensures the increase in iterations in an annealing cycle necessary for optimization.

2) GLOBAL STOPPING CRITERIA

It is the criteria of the overall optimization process due to which it terminates when any of the conditions are met: (a) Total number of annealing cycles exceed the maximum limit $global\_iter$. (b) The ratio of the number of rejected solutions (rejection count) to the total solutions (accepted and rejected), exceeds the critical limit ϵ. (c) Annealing temperature falls below the final value T_f.

G. SIMULATION PARAMETERS

The parameters involved in realizing the algorithm have been defined in Table 3. Maintaining $\Delta h\_avg = 0.001$, $global\_iter = 1000$ and $\epsilon = 0.99$, the parameters P_i, P_f, α, k have been varied according to the different combination orders **PCj**(P_i, P_f, α, k) as follows:

PC1(0.8, 0.1, 0.85, 5), PC2(0.8, 0.01, 0.85, 5), PC3(0.8, 0.005, 0.85, 5), PC4(0.9, 0.1, 0.85, 5), PC5(0.9, 0.01, 0.85, 5), PC6(0.9, 0.005, 0.85, 5), PC7(0.95, 0.01, 0.85, 5), PC8(0.99, 0.01, 0.85, 5), PC9(0.99, 0.01, 0.85, 6), PC10(0.99, 0.01, 0.85, 7), PC11(0.99, 0.01, 0.85, 8), PC12(0.99, 0.01, 0.85, 9), PC13(0.99, 0.01, 0.9, 5), PC14(0.99, 0.01, 0.95, 5), PC15(0.99, 0.01, 0.99, 5), PC16(0.99, 0.01, 0.99, 6), PC17(0.99, 0.01, 0.99, 7), PC18(0.99, 0.01, 0.99, 8), PC19(0.99, 0.01, 0.99, 9). As a test case the parameter combinations have been applied for the optimization of ami33 benchmark circuit with weight combination $W_1 = 1$, $W_2 = 0$, $W_3 = 0$ associated with the cost function as defined in (6).

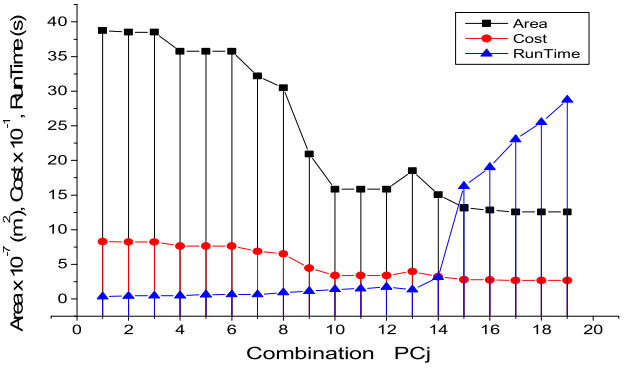

FIGURE 6. Variation of optimized cost, optimized area and run time of optimization with respect to the different simulation parameter combinations PCj for ami33 benchmark circuit with rotatable blocks.

It can be observed from the graph in Fig. 6 that the Cost and the area metric show negligible variance with the final acceptance probability P_f (combinations PC1 to PC3 or PC4 to PC6) since the solution converges much before the reaching the final annealing temperature T_f characterized by the final acceptance probability P_f. The cost and the area metric show a downward trend with the increase in the value of initial acceptance probability P_i (combinations PC2, PC5, PC7, and PC8), yet involving a minor increase in the run time. The Cost and area metric also show a downward trend with the increase in α (combinations PC8, PC13, PC14, PC15). However, there is an increase in the runtime with an increase in cooling rate α. For lower values of cooling rate α, the cost and the area metric shows noticeable improvement with increasing values of the local iteration multiplier k (combinations PC8 to PC12) and becomes constant after a certain value of k (combination PC10). But for higher values of the cooling rate α, there is a minor improvement in cost and area metric with an increase in local iteration multiplier k (combinations PC15 to PC19). Observation shows that the runtime increases with an increase in k. It is clear from the graph in Fig. 6 that combinations PC17 to PC19 give the minimum optimized cost. That is these combinations PC17 to PC19 provide the global minima. In this work, we have chosen the simulation parameter combination PC17 where $P_i = 0.99$, $P_f = 0.01$, $\alpha = 0.99$, $k = 7$. The choice of parameters in PC17 ensures a very high initial acceptance probability P_i (99%) and a very low final acceptance probability P_f (1%) as is required for the operation of the classical Simulated Annealing. Moreover, the combination PC17 provides the global minimum cost at the expense of lesser runtime. The same trend has been observed with other cost metrics and other benchmark circuits as well.

TABLE 3. Simulation parameters of algorithm.

Parameters	Description
$global\_iter$	Maximum number of annealing cycles in the complete optimization process.
$local\_iter$	Maximum iterations in each annealing cycle.
Δh_avg	Average cost difference of acceptable inferior solutions.
P_i	Initial probability of accepting inferior solutions at T_i.
P_f	Final probability of accepting inferior solutions at T_f.
T_i	Initial high annealing temperature. As derived from (7), $T_i = -\Delta h\_avg / ln\ (P_i)$
T_f	Final low annealing temperature. As derived from (7), $T_f = -\Delta h\_avg / ln\ (P_f)$
α	Decay rate or cooling rate. It is a fractional valued parameter varying between 0 and 1.
σ	Maximum inferior solutions acceptable in an annealing cycle.
ϵ	Upper bound on the ratio of total unacceptable solutions to the total solutions of the complete optimization process.
k	Local iteration multiplier. It has integral value.

H. ALGORITHM FLOW

Step1: Provide inputs to the Algorithm viz. (a) dimensions, (b) connectivity and (c) steady-state power dissipation of the functional blocks.

TABLE 4. Best solution attributes for different weight combinations according to the proposed algorithm, DOTFloor.

CO_i	Metrics	Non Rotatable Blocks					Rotatable Blocks				
		apte	xerox	hp	ami33	ami49	apte	xerox	hp	ami33	ami49
CO_1	Area (10^{-6} m^2)	47.5287	20.7317	9.3721	1.3664	42.5477	46.9248	20.088	9.201	1.2567	38.3194
	WLM (m)	1.4600	2.6625	0.5239	1.1453	4.3521	2.262	1.5188	0.3614	1.4204	4.2931
	PT (K)	362.90	365.09	346.54	331.38	364.67	364.95	359.58	351.95	333.34	364.84
CO_2	Area (10^{-6} m^2)	48.2118	24.8010	10.6389	2.2691	70.5571	48.4967	21.0508	9.8988	1.8934	55.0468
	WLM (m)	0.9568	0.9688	0.2170	0.9861	2.2744	0.9537	0.9568	0.201	0.9304	1.9762
	PT (K)	363.520	358.920	343.66	331.14	361.63	363.26	358.84	342.5	329.49	367.49
CO_3	Area (10^{-6} m^2)	107.1727	72.7709	32.5736	12.3727	236.648	73.5593	34.1463	16.9591	4.2093	85.9697
	WLM (m)	1.6200	2.2930	0.9422	3.2555	8.7773	1.4144	1.4743	0.5989	2.1531	5.0174
	PT (K)	359.610	354.190	332.590	322.670	352.290	359.68	354.4	333.84	324.09	354.63
CO_4	Area (10^{-6} m^2)	48.0979	20.8586	9.4403	1.4524	46.1398	48.1641	21.0163	9.575	1.3243	40.5877
	WLM (m)	0.9573	1.0627	0.2896	1.1153	3.2985	1.0122	0.9976	0.2671	1.0343	2.5985
	PT (K)	362.250	358.180	342.800	330.210	359.100	361.81	362.03	338.18	332.46	361.29
CO_5	Area (10^{-6} m^2)	48.0599	23.2745	10.4076	1.8004	49.5954	48.4967	21.7335	9.8988	1.4796	44.1137
	WLM (m)	0.9570	0.9770	0.2161	1.0008	2.4062	0.9546	0.972	0.2133	0.9413	2.1609
	PT (K)	362.28	357.38	341.3	328.52	361.23	363.23	360.57	342.22	330.2	360.79
CO_6	Area (10^{-6} m^2)	48.4782	23.6351	10.0842	1.7919	54.8600	48.457	21.701	10.3118	1.4847	58.937
	WLM (m)	0.9992	0.9937	0.2427	1.0534	2.8360	1.0366	0.9997	0.2564	1.0929	3.0315
	PT (K)	361.990	356.450	339.360	328.460	358.130	360.31	357.09	337.3	329.72	358.21

Step2: Randomly generate the encoded initial floorplan solution. The initial solution defines the first present solution termed as *Present_sol*.

Step3: Operate the present solution by the function *Cost*(), to determine the cost of the present solution given by $Present\_cost = Cost(Present\_sol)$.

Step4: Initialize the best solution *Best_sol* as, $Best\_sol = Present\_sol$ and the best solution cost *Best_cost* as, $Best\_cost = Present\_cost$.

Step5: Execute the n^{th} Annealing cycle at a constant annealing temperature T_n. (For the 1st annealing cycle $n = 1$ and $T_n = T_i$). Step5 consists of the following sub-steps.

- **Step5a:** Operate the function *Perturb*() on the present solution to generate the new solution *New_sol*, given by $New\_sol = Perturb(Present\_sol)$.
- **Step5b:** Operate the new solution by the function *Cost*(), to determine the cost of the new solution New_cost, given by $New\_cost = Cost(New\_sol)$.
- **Step5c:** Accept the new solution if it satisfies the Metropolis acceptance criteria. Else the solution is rejected and the rejection count is incremented by unity.
- **Step5d:** If the new solution is accepted, then
  - ➢ Update the present solution and present cost as, $Present\_sol = New\_sol$ and $Present\_cost = New\_cost$.
  - ➢ If the condition $New\_cost < Best\_cost$ is satisfied, then update the best solution and best cost as $Best\_sol = New\_sol$ and $Best\_cost = New\_cost$ respectively.

Execute Step5 until the local stopping criteria is attained.

Step6: For the next $(n+1)^{th}$ annealing cycle update the new annealing temperature T_{n+1} according to a geometric cooling schedule as $T_{n+1} = \alpha^n T_n$.

Step7: Repeat Step5 for the next Annealing cycle until the global stopping criteria is met.

The output of the algorithm is the optimal floorplan solution denoted by the *Best_sol*.

V. RESULTS AND ANALYSIS

The proposed algorithm DOTFloor (Diffusion Oriented Time-improved Floorplanner) has been experimented on MCNC (Microelectronics Center of North Carolina) benchmark circuits [32] by considering a random power density distribution for the functional blocks ranging between 0.022×10^6 W/m^2 and 2.4×10^6 W/m^2. The average power densities considered are 0.593831×10^6 W/m^2 for apte, 0.729076×10^6 W/m^2 for xerox, 0.72433×10^6 W/m^2 for hp, 0.729749×10^6 W/m^2 for ami33, and 0.728304×10^6 W/m^2 for ami49.

A. WEIGHT PARAMETERS IN COST FUNCTION OF DOTFloor AND OPTIMIZED RESULTS

The weight parameters W_k in (6) are varied between 0 and 1 during optimization to track the optimal and the other best solutions. The different weight combinations $\mathbf{CO_i}(W_1, W_2, W_3)$ in (6) considered by DOTFloor for optimization are CO_1(1,0,0), CO_2(0,1,0), CO_3(0,0,1), CO_4(0.5,0.25,0.25), CO_5(0.25,0.5,0.25), and CO_6(0.25,0.25,0.5). The trade-offs between the cost metrics with the relative variation in their corresponding weight parameter W_k can be observed in Table 4. Moreover, the cost metric shows an improving trend with the increase in the value of its associated weight parameter W_k. In Table 4 area shows the maximum improvement when the corresponding weight parameter W_1 is given 100% optimization weightage (i.e.$W_1 = 1$) in combination CO_1. Combination CO_1 gives only area-aware optimization. Similarly, CO_2 and CO_3 provide only (wire length metric) WLM-aware and only temperature-aware optimizations respectively. Table 4 shows the best solution attributes obtained by DOTFloor corresponding to the different weight combinations CO_i. DOTFloor finds the optimal solution according to the method discussed in section V(C). For the purpose of comparison and validation, the HotSpot tool has been used at the end to quantify the peak temperature (PT) of the optimized solutions of DOTFloor.

TABLE 5. Best solution attributes according to HotFloorplan.

Solution	Metrics	Non Rotatable Blocks					Rotatable Blocks				
		apte	xerox	hp	ami33	ami49	apte	xerox	hp	ami33	ami49
Area-aware	Area (10^{-6} m^2)	47.9141	20.7317	9.8255	1.5418	43.01	46.9248	20.3862	9.2857	1.5126	38.67
	WLM (m)	2.441	2.609	0.533	1.228	4.5765	2.3815	2.4675	0.593	1.19	3.4985
	PT (K)	365.43	368.34	345.44	333.74	363.74	365	365.07	352.3	331.29	365.72
WLM-aware	Area (10^{-6} m^2)	48.2119	24.8744	11.8878	1.9757	67.533	48.4967	23.1006	14.2468	1.9158	55.86
	WLM (m)	0.957	1.01	0.22	0.995	2.8155	0.9545	0.964	0.2	0.9425	1.982
	PT (K)	363.58	357.79	339.92	331.73	364.1	363.29	359.16	341.85	329.45	364.82
Temperature -aware	Area (10^{-6} m^2)	104.288	47.0474	37.8358	10.8756	221.11	66.44	31.6366	19.9111	4.8383	144.866
	WLM (m)	1.9685	2.082	1.454	3.205	9.689	1.333	1.926	0.8825	1.9915	6.825
	PT (K)	359.29	354.44	332.53	323.86	355.93	359.48	355.78	335.31	324.88	353.88
Optimal	Area (10^{-6} m^2)	48.0979	24.8010	10.6389	1.7008	49.961	48.4967	21.701	11.7312	1.6673	54.4159
	WLM (m)	0.967	0.969	0.244	1.029	2.4045	0.9545	0.9645	0.2055	0.9325	1.9975
	PT (K)	362.84	358.91	343.83	328.63	361.99	363.29	363.58	341.65	330.07	365.37

B. WEIGHT PARAMETERS IN COST FUNCTION OF HotFloorplan AND OPTIMIZED RESULTS

For validating the performance of the proposed algorithm, the HotFloorplan tool has been used from the HotSpot tool package available at [33]. For experimentation purposes, the HotFloorplan tool has been configured to handle fixed as well as free and hard macrocells. The cost function of the HotFloorplan tool is

$$Cost = \lambda_A \times Area + \lambda_T \times Tmax + \lambda_W \times 3 \times WL \quad (8)$$

Here in (8) Area denotes the chip area, *Tmax* represents the peak temperature and *WL* indicates the wire length metric of the floorplan. The wire length *WL* in (8) is computed similar to the wire length metric given in (5). Parameters λ_A, λ_T, and λ_W are the weights of the cost metrics. The default values given in the HotFloorplan tool [33] for the optimal solution are $\lambda_A = 5 \times 10^6$, $\lambda_T = 1$, $\lambda_W = 350$. The weight combinations considered in this work for only area optimization is ($\lambda_A = 5 \times 10^6$, $\lambda_T = 0$, $\lambda_W = 0$), for only temperature optimization is ($\lambda_A = 0$, $\lambda_T = 1$, $\lambda_W = 0$), and for only wire length metric (WLM) optimization is ($\lambda_A = 0$, $\lambda_T = 0$, $\lambda_W = 350$). It can be observed from Table 5 that cost metrics improve with the increase in the corresponding weight parameter value. Table 5 depicts the attributes of the area-aware, WLM-aware, temperature-aware and optimal solutions as obtained from the HotFloorplan tool by the adjustment of the weight parameters in its cost function.

C. OPTIMAL COMBINATION AND OPTIMAL SOLUTION IN DOTFLOOR (DIFFUSION ORIENTED TIME-IMPROVED FLOORPLANNER)

DOTFloor defines a unique method to determine the optimal floorplan solution. It characterizes a 3D Cartesian coordinate frame representing the solution space, to have the following salient features.

(i) The frame has three orthogonal axes viz. x, y and z, each representing the normalized area, wire length metric, and peak temperature respectively. The three axes being normalized get calibrated in the range of zero to one.

(ii) Every point in the 3D-frame represents a floorplan solution of a circuit for a particular weight combination CO_i given in Table 4. The x, y, and z coordinate of the point represent the normalized values of its area, wire length metric, and peak temperature.

(iii) The solution set of a circuit (representing the solution points) are plotted in the frame by normalizing the cost metrics with respect to the corresponding highest metric values of the circuit. For example, considering the solution set of Xerox given in Table 4, the highest area metric value is 72.7709×10^{-6} m^2. The area metric of solutions (corresponding to the different combinations CO_i) of Xerox is normalized by dividing with the highest area metric value. The similar process follows for obtaining the normalized values of other metrics as well.

(iv) The solution point of a particular circuit closest to the origin of the frame provides its optimal solution.

Table 6 lists the Euclidian distances of the solution points of different circuits corresponding to the different CO_i weight combination values. Parameter *D_Avg* represents the average Euclidian distance (taking an average of the five different circuits) for each CO_i weight combination. Steps (i), (ii), (iii) and (iv) have been repeated, firstly for fixed block and secondly for free block floorplans. It is clear from Table 6 that the minimum average Euclidian distance is obtained with the CO_5 combination for both cases of floorplans with fixed and free oriented blocks.

D. COMPARATIVE STIDY OF EXPERIMENTAL RESULTS

1) SINGLE METRIC OPTIMIZATION

Table 7 and Fig. 7 show the percentage improvement, obtained by DOTFloor over the HotFloorplan, upon single objective optimization of the targeted cost metric. Table 7 has been prepared by comparing the area metric of the area-aware solutions, wire length metric(WLM) of the WLM-aware solutions and peak temperature (PT) of the temperature-aware solutions of DOTFloor and HotFloorplan tool given in Table 4 and Table 5 respectively. The temperature improvement in Table 7 has been computed in units of degree Celsius.

TABLE 6. Euclidean distances of the solutions (Given in TABLE 4) from the origin of 3D-Cartesian Frame. Average Euclidean distance to be used for obtaining the optimal solution.

CO_i	Non Rotatable Blocks						Rotatable Blocks					
	apte	xerox	hp	ami33	ami49	D_Avg	apte	xerox	hp	ami33	ami49	D_Avg
CO_1	1.42	1.44	1.2	1.07	1.13	1.247	1.55	1.53	1.29	1.23	1.38	1.396
CO_2	1.25	1.1	1.1	1.06	1.07	1.1089	1.27	1.33	1.18	1.17	1.25	1.24
CO_3	1.73	1.64	1.7	1.72	1.71	1.7005	1.54	1.70	1.70	1.72	1.71	1.674
CO_4	1.24	1.1	1.1	1.06	1.07	1.1095	1.27	1.35	1.20	1.15	1.21	1.236
CO_5	1.24	1.09	1.1	1.05	1.05	1.0986	1.27	1.34	1.19	1.14	1.19	1.226
CO_6	1.26	1.09	1.1	1.05	1.06	1.1042	1.27	1.35	1.21	1.17	1.34	1.268

TABLE 7. Percentage improvement by DOTFloor over HotFloorplan considering single cost metric optimization.

Single objective Solution	% improvement in targeted cost metric	Non Rotatable Blocks					Rotatable Blocks				
		apte	xerox	hp	ami33	ami49	apte	xerox	hp	ami33	ami49
Area-aware	Area	0.804	0	4.614	11.375	1.075	0.000	1.463	0.912	16.920	0.907
WLM-aware	WLM	0.018	4.078	1.351	0.891	19.220	0.084	0.747	-0.500	1.284	0.293
Temperature-aware	Temperature	-0.371	0.308	-0.101	2.347	4.397	-0.056	0.388	0.438	0.243	-0.212

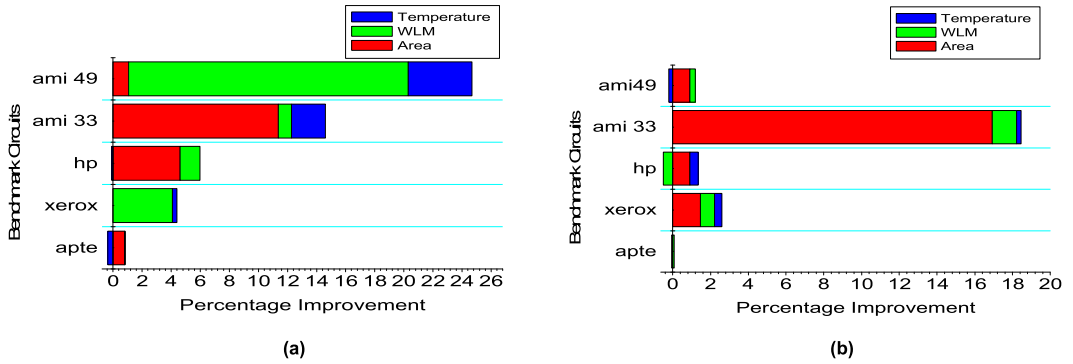

FIGURE 7. Average percentage improvement in area, WLM and temperature metrics of the area-aware, WLM-aware and temperature-aware solutions respectively by DOTFloor over the HotFloorplan for (a) fixed macrocell floorplans, (b) rotatable macrocell floorplans.

The following observations can be made from Table 7 and Fig. 7:

(a) DOTFloor achieves an average improvement of 3.57% area in the area-aware solutions, 5.11% in the wire length metric (WLM) of WLM-aware solutions and, 1.32% peak temperature in the temperature-aware solutions respectively over the HotFloorplan tool, considering the case of fixed macrocell floorplans.

(b) DOTFloor achieves an average improvement of 4.04% area in the area-aware solutions, 0.38% wire length metric (WLM) in the WLM-aware solutions, 0.16% peak temperature in the temperature-aware solutions respectively over the HotFloorplan tool, considering the case of rotatable macrocell floorplans.

2) OPTIMAL SOLUTION

Table 8 and Fig. 8 show a comparative study of the cost metrics of the optimal solutions of DOTFloor and HotFloorplan. The temperature improvement in Table 8 has been computed in units of degree Celsius. The following observations can be made from Table 8 and Fig. 8:

(a) DOTFloor has achieved an average improvement of 0.66% area, 2.86% wire length metric (WLM) and 1.41% peak temperature in the optimal floorplan solution composed of fixed macrocells with respect to the HotFloorplan.

(b) DOTFloor has achieved an average improvement of 9.13% area, -2.74% wire length metric (WLM) and 0.38% peak temperature in the optimal floorplan solution composed of rotatable macrocells with respect to the HotFloorplan.

The average percentage improvement for a cost metric has been computed from the mean of the improvement percent of the metric obtained in the different test circuits (given in Table 7 and Table 8). Hence it is noticeably monitored that the DOTFloor provides almost the equivalent quality of floorplan solutions as the HotFloorplan tool and this proves that the developed floorplanner DOTFloor is quite reliable.

Fig. 9 and Fig. 10 represent the temperature profile and the floorplans of area-aware, temperature-aware and optimal solutions of DOTFloor for the ami49 benchmark-circuit considering the case of fixed blocks (in Fig. 9) and rotatable blocks (in Fig. 10). It is clear from Fig. 9, Fig. 10 and Table 4 that the temperature-aware solution has the least peak on-chip temperature but the highest area, the area-aware solution has the least area but the highest peak temperature, the optimal solution has a better area than the temperature-aware

TABLE 8. Percentage improvement obtained by DOTFloor over HotFloorplan for optimal solution.

% improvement in metric	Non Rotatable Blocks					Rotatable Blocks				
	apte	xerox	hp	ami33	ami49	apte	xerox	hp	ami33	ami49
Area	0.079	6.155	2.174	-5.857	0.732	0.0000	-0.1498	15.6199	11.2561	18.9323
WLM	1.035	-0.827	11.444	2.738	-0.069	-0.0105	-0.7776	-3.7956	-0.9437	-8.1802
Temperature	0.624	1.784	3.580	0.198	0.855	0.0165	0.8279	-0.1668	-0.0394	1.2535

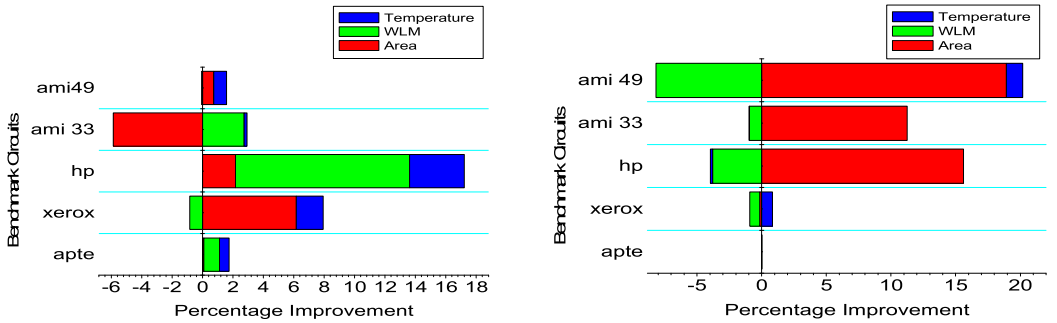

FIGURE 8. Average percentage improvement in cost metrics of the optimal floorplan solution provided by DOTFloor over HotFloorplan for (a) fixed macrocell floorplans, (b) rotatable macrocell floorplans.

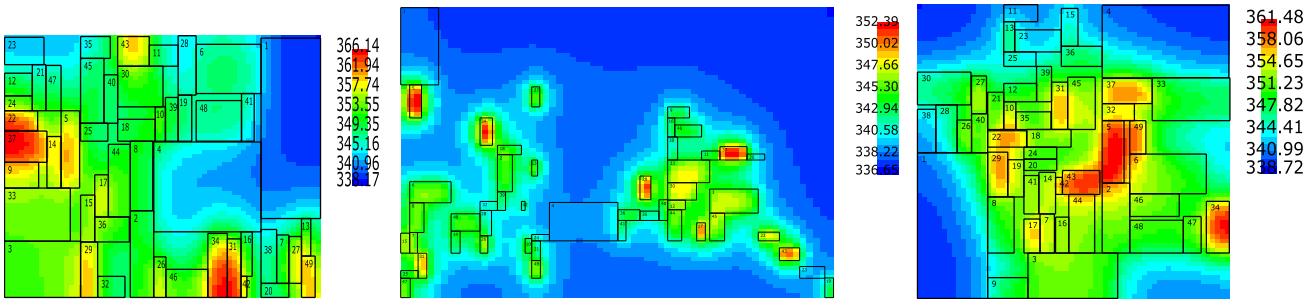

FIGURE 9. Temperature map and floorplan of ami49 (in Kelvin) from DOTFloor with fixed macrocells (a) area-aware solution (b) temperature-aware solution (c) optimal solution.

TABLE 9. Execution time comparison between DOTFloor and HotFloorplan.

Floorplan Solution	Run time	Non Rotatable Blocks					Rotatable Blocks				
		apte	xerox	hp	ami33	ami49	apte	xerox	hp	ami33	ami49
Area-aware	Proposed (s)	0.72	2.05	0.86	13.66	28.24	1.34	1.73	1.3	23.03	75.18
	HotFloorplan (s)	28	48	75	4052	18086	36	68	90	3950	17341
WLM-aware	Proposed (s)	0.6	1.16	1.09	7.84	29.57	2.24	2.06	3.56	18.99	83.37
	HotFloorplan(s)	9	23	43	3227	14288	12	22	86	3231	9443
Temperature-aware	Proposed (s)	1.74	1.07	1.31	10.14	28.73	4.3	1.64	6.84	31.09	97.74
	HotFloorplan (s)	34	56	79	4068	18409	45	72	101	3993	18244
Optimal	Proposed (s)	0.88	0.83	0.76	9.36	36.61	1.86	1.71	2.06	22.56	86.41
	HotFloorplan (s)	14	33	50	3835	12804	13	20	60	3566	11142

solution as well as better peak temperature than the area-aware solution.

Table 9 depicts the comparison between the execution times of the DOTFloor and HotFloorplan tool under the different optimization conditions. Fig. 11 clearly shows that DOTFloor provides a very high run-time saving with respect to the HotFloorplan tool for floorplans composed of fixed as well as rotatable and hard macrocells. The DOTFloor gives an average 96.67% improvement in run time over the HotFloorplan tool while maintaining almost the equal quality of floorplan solutions. For a particular solution (Area-aware, WLM-aware, Temperature-aware or optimal) the average run time improvement percent has been computed from the mean of the percentage improvement in run time obtained for the different test circuits (given in Table 9).

From Table 10, it has been observed that the proposed floorplanner DOTFloor achieves improved results in terms of area utilization with respect to the previous works.

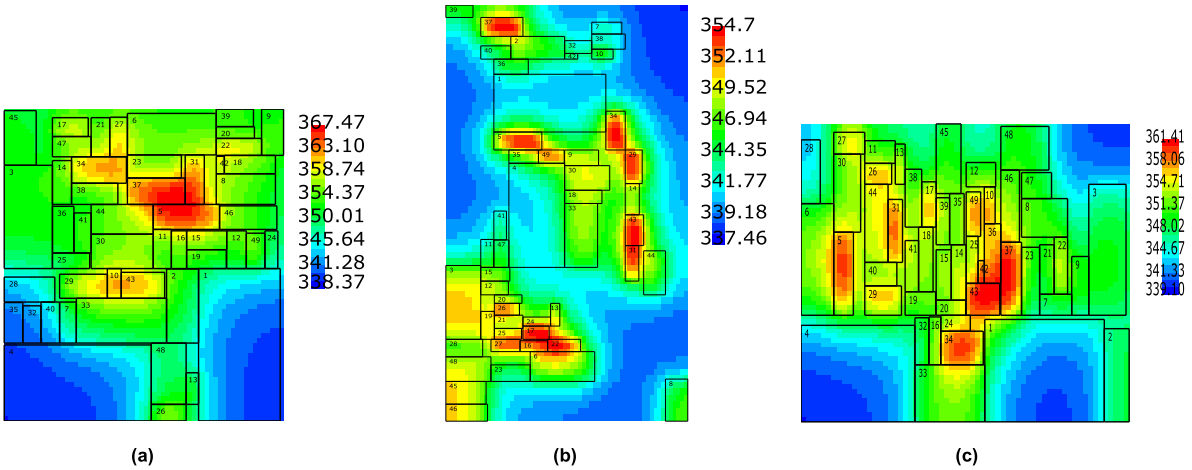

FIGURE 10. Temperature map and floorplan of ami49 (in Kelvin) from DOTFloor with rotatable macrocells (a) area-aware solution (b) temperature-aware solution (c) optimal solution.

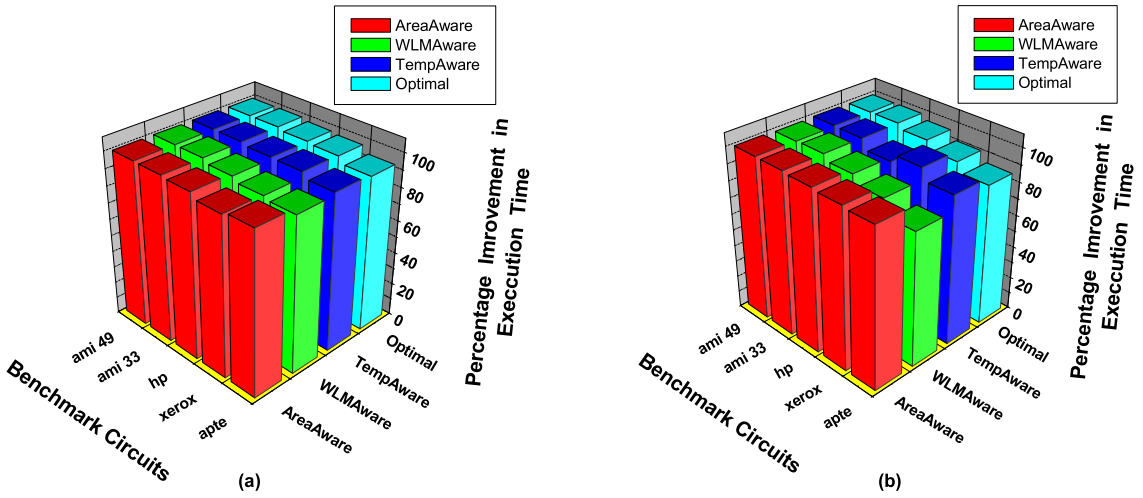

FIGURE 11. Percentage improvement in execution time of DOTFloor over HotFloorplan under different floorplan optimization conditions with (a) fixed Macrocells (b) rotatable macrocells.

TABLE 10. Comparison of chip area of the area-aware solution.

Circuit	Non Rotatable Blocks		Rotatable Blocks			
	Area (10^{-6} m^2)		Area (10^{-6} m^2)			
	[15]	Proposed	[7]	[17]	[15]	Proposed
apte	48.05	47.5287	47.52	46.92	48.12	46.9248
xerox	20.44	20.7317	20.26	20.38	20.01	20.088
hp	9.36	9.3721	9.44	-	9.26	9.201
ami33	1.37	1.3664	1.27	1.29	1.28	1.2567
ami49	42.37	42.5477	39.16	38.93	40.04	38.3194

TABLE 11. Percentage improvement in peak temperature of the temperature-aware solution.

Circuit	Rotatable Blocks	
	% improvement in peak temperature	
	[7] Approx	Proposed
apte	17.27	3.67
xerox	19.31	11.86
hp	2.44	19
ami33	0.89	14.96
ami49	3.03	13.53
Average % improvement in peak temperature	8.588	12.604

Since the power considerations in [7] and the present work differ, instead of a direct temperature comparison, a thermal assessment has been done between [7] and the proposed floorplanner by considering the percentage improvement in temperature of the temperature-aware solution over the area-aware solution (which is actually temperature un-aware) of corresponding papers. Table 11 shows that the proposed algorithm achieves better average improvement in peak temperature over [7].

VI. CONCLUSION AND FUTURE SCOPE

This paper presents a thermal-aware floorplanning algorithm that is suitable for handling fixed as well as rotatable and hard

functional blocks or macrocells. The proposed floorplanner DOTFloor (Diffusion Oriented Time-improved Floorplanner) is based on SA and has been developed using the C language and experiments have been conducted in a Linux platform operating on a 3.0 GHz Core-i5 Intel processor. The algorithm adopts a heat-diffusion based thermal model FATT (Fast Assumption Technique for Temperature) which helps in reducing the execution time by providing an assumption of the hotness instead of accurate temperature estimation during the run time of the optimization process. The results of this work show very significant achievement in the runtime efficiency of DOTFloor while simultaneously maintaining an equivalent qualitative performance compared to the Hotfloorplan. Moreover, the proposed floorplanner also shows improved performance over previous works as well. The proposed florrplanner successfully gives a good optimization of area, wire length metric and peak temperature metrics. As a future scope, the work has a good potential of being extended to the design of non-slicing floorplan and 3D IC floorplan as well.

REFERENCES

[1] B. Goplen and S. Sapatnekar, "Efficient thermal placement of standard cells in 3D ICs using a force directed approach," in *Proc. ICCAD*, San Jose, CA, USA, 2003, pp. 86–89.
[2] C.-H. Tsai and S.-M. Kang, "Cell-level placement for improving substrate thermal distribution," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 19, no. 2, pp. 253–266, Feb. 2000.
[3] Y. Zhan and S. S. Sapatnekar, "High-efficiency Green function-based thermal simulation algorithms," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 9, pp. 1661–1675, Sep. 2007.
[4] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 5, pp. 501–513, May 2006.
[5] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, "Temperature-aware microarchitecture: Modeling and implementation," *ACM Trans. Archit. Code Optim.*, vol. 1, no. 1, pp. 94–125, Mar. 2004.
[6] W. Huang, M. R. Stan, and K. Skadron, "Parameterized physical compact thermal modeling," *IEEE Trans. Compon. Packag. Technol.*, vol. 28, no. 4, pp. 615–622, Dec. 2005.
[7] W.-L. Hung, Y. Xie, N. Vijaykrishnan, C. Addo-Quaye, T. Theocharides, and M. Irwin, "Thermal-aware floorplanning using genetic algorithms," in *Proc. ISQED*, San Jose, CA, USA, Mar. 2005, pp. 634–639.
[8] K. Sankaranarayanan, M. Stan, and K. Skadron, "A case for thermal-aware floorplanning at the microarchitectural level," *J. Instruct.-Level Parallelism*, vol. 7, no. 1, pp. 8–16, 2005.
[9] P. Sivaranjani and A. S. Kumar, "Thermal-aware non-slicing VLSI floorplanning using a smart decision-making PSO-GA based hybrid algorithm," *Circuits, Syst., Signal Process.*, vol. 34, no. 11, pp. 3521–3542, 2015.
[10] A. T. Winther, W. Liu, A. Nannarelli, and S. Vrudhula, "Thermal aware floorplanning incorporating temperature dependent wire delay estimation," *Microprocessors Microsyst.*, vol. 39, no. 8, pp. 807–815, 2015.
[11] P. Mukherjee, S. D'souza, and S. Chattopadhyay, "Area constrained performance optimized ASNoC synthesis with thermal-aware white space allocation and redistribution," *Integration*, vol. 60, pp. 167–189, Jan. 2018.
[12] S. H. Gade, P. Kumar, and S. Deb, "A Pre-RTL floorplanner tool for automated CMP design space exploration with thermal awareness," in *Proc. VDAT*, Guwahati, India, May 2016, pp. 1–6.
[13] J. Song, Y.-M. Lee, and C.-T. Ho, "ThermPL: Thermal-aware placement based on thermal contribution and locality," in *Proc. VLSI-DAT*, Hsinchu, Taiwan, Apr. 2016, pp. 1–4.
[14] Z. Jiang and N. Xu, "HotSpot thermal floorplan solver using conjugate gradient to speed up," *Mobile Inf. Syst.*, vol. 2018, Apr. 2018, Art. no. 2921451.
[15] S. Banerjee, A. Ratna, and S. Roy, "Satisfiability modulo theory based methodology for floorplanning in VLSI circuits," in *Proc. ISED*, Patna, India, Dec. 2016, pp. 91–95.
[16] A. Singh and L. Jain, "VLSI floorplanning using entropy based intelligent genetic algorithm," in *Proc. ICAN*, Chandigarh, India, 2017, pp. 53–71.
[17] G. Chen, W. Guo, and Y. Chen, "A PSO-based intelligent decision algorithm for VLSI floorplanning," *Soft Comput.*, vol. 14, no. 12, pp. 1329–1337, 2010.
[18] J.-M. Lin, T.-T. Chen, Y.-F. Chang, W.-Y. Chang, Y.-T. Shyu, Y.-J. Chang, and J. M. Lu, "A fast thermal-aware fixed-outline floorplanning methodology based on analytical models," in *Proc. ICCAD*, San Diego, CA, USA, Nov. 2018, pp. 1–8.
[19] A. Sinharay, P. Roy, and H. Rahaman, "VLSI thermal placement issues: A cooperative game theory based approach," in *Proc. ISED*, Patna, India, Dec. 2016, pp. 106–111.
[20] Y. Han and I. Koren, "Simulated annealing based temperature aware floorplanning," *J. Low Power Electron.*, vol. 3, no. 2, pp. 141–155, 2007.
[21] S. N. Adya and I. L. Markov, "Fixed-outline floorplanning: Enabling hierarchical design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 6, pp. 1120–1135, Dec. 2003.
[22] L. Wang, "Fast algorithms for thermal-aware floorplanning," *J. Circuits, Syst. Comput.*, vol. 23, no. 7, 2014, Art. no. 1450098.
[23] A. Das, S. R. Choudhury, B. K. Kumar, and S. N. Pradhan, "An elitist area-power density trade-off in VLSI floorplan using genetic algorithm," in *Proc. ICECE*, Dhaka, Bangladesh, Dec. 2012, pp. 729–732.
[24] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
[25] D. Delahaye, S. Chaimatanan, and M. Mongeau, "Simulated annealing: From basics to applications," in *Handbook of Metaheuristics* (International Series in Operations Research & Management Science), vol. 272. Cham, Switzerland: Springer, 2019, ch. 1, pp. 1–35.
[26] S. M. Sait and H. Youssef, *VLSI Physical Design Automation: Theory and Practice*. Singapore: World Scientific, 2001.
[27] Y. Han, I. Koren, and C. A. Moritz, "Temperature aware floorplanning," in *Proc. 2nd Workshop Temp.-Aware Comput. Syst. (TACS)*, 2005, pp. 1–10.
[28] R. H. J. M. Otten and R. K. Brayton, "Planning for performance," in *Proc. DAC*, San Francisco, CA, USA, 1998, pp. 122–127.
[29] L. Stockmeyer, "Optimal orientations of cells in slicing floorplan designs," *Inf. Control*, vol. 57, nos. 2–3, pp. 91–101, 1983.
[30] S. H. Gerez, *Algorithms for VLSI Design Automation*. Hoboken, NJ, USA: Wiley, 1999.
[31] D. F. Wong and C. L. Liu, "A new algorithm for floorplan design," in *Proc. DAC*, Las Vegas, NV, USA, Jun./Jul. 1986, pp. 101–107.
[32] *MCNC Benchmark Circuits*. Accessed: Jul. 10, 2017. [Online]. Available: https://s2.smu.edu/~manikas/Benchmarks/MCNC_Benchmark_Netlists.html
[33] *HotSpot and HotFloorplan Package*. Accessed: May 15, 2017. [Online]. Available: http://lava.cs.virginia.edu/hotspot

SOMNATH ROY CHOUDHURY received the M.Tech. degree in VLSI design from the Vellore Institute of Technology, India, in 2009. He is currently pursuing the Ph.D. degree with the Department of ECE, National Institute of Technology Agartala, India. He is currently an Assistant Professor with the Department of ECE, National Institute of Technology Agartala. His research interests include VLSI physical design automation, thermal management in processors, and CAD tool development.

SAMBHU NATH PRADHAN received the M.E. degree from Bengal Engineering and Science University, India, in 2004, and the Ph.D. degree from IIT Kharagpur, India, in 2010. He is currently an Associate Professor with the Department of ECE, National Institute of Technology Agartala, India. His research interests include VLSI physical design automation, low-power design and testing, and thermal-aware logic synthesis.