Received 24 January 2023, accepted 8 February 2023, date of publication 13 February 2023, date of current version 16 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3244346

RESEARCH ARTICLE

A Novel Multicriteria Optimization Technique for VLSI Floorplanning Based on Hybridized Firefly and Ant Colony Systems

B. SRINIVASAN[1], R. VENKATESAN[1], BELQASEM ALJAFARI[2], (Member, IEEE), KETAN KOTECHA[3], V. INDRAGANDHI[4], AND SUBRAMANIYASWAMY VAIRAVASUNDARAM[1]

[1]School of Computing, SASTRA Deemed University, Thanjavur 613401, India
[2]Department of Electrical Engineering, Najran University, Najran 11001, Saudi Arabia
[3]Symbiosis Centre for Applied Artificial Intelligence, Symbiosis International (Deemed University), Pune 412115, India
[4]School of Electrical Engineering, Vellore Institute of Technology, Vellore 632014, India

Corresponding authors: Ketan Kotecha (head@scaai.siu.edu.in) and Subramaniyaswamy Vairavasundaram (vsubramaniyaswamy@gmail.com)

This work was supported by the Symbiosis International (Deemed University), Pune, India.

ABSTRACT In VLSI circuit design, physical design is one of the main steps in placing the circuit into the chip area. Floorplanning is a crucial step in the physical design of IC design, which generates a blueprint for the placement of circuit modules into the chip. A floorplanning step accepts a netlist as its input, given by the circuit-partitioning step of physical design. The floorplanning step generates optimal placements for the circuit modules. The netlist contains the modules' dimensions, size, and interconnect information. During the floorplan generation, the chip area, wire length required for connecting modules and the heat generated by the chips can be estimated. A good floorplan makes placements and routing simple. In order to improve the circuit performance by minimizing chip area, wire length, and peak temperature, it is essential to generate an optimized floorplan by developing metaheuristic optimization algorithms. A novel Hybridized Multicriteria Ant Colony and Firefly Optimization (HMAC-FO) technique is introduced to generate an optimized floorplan. The primary focus of HMAC-FO technique is to design a model for generating efficient floorplanning. The standard MCNC benchmark dataset has a number of modules with their connections. Firefly optimization is the main algorithm in HMAC-FO, which has been used in generating efficient floorplan with the optimized area, wire length and thermal. The firefly algorithm initially requires some number of solutions as the fireflies' population. In the firefly algorithm, usually, the populations are generated randomly. But, to improve the firefly optimization algorithm's performance and obtain better optimum results, the proposed technique uses ACO to generate the initial population, which are all optimal solutions. The firefly algorithm uses the set of optimal solutions as the initial population and generates the globally optimal result. The proposed technique has experimented with the standard MCNC benchmark circuits, and the results prove that the proposed algorithm reduces the area by 3.48%, the wire length by 0.64% and the temperature by 3.33% than the best existing methodology. The proposed methodology generates a good floorplan with all the required optimization, such as area, wire length and heat generation.

INDEX TERMS VLSI floorplanning, firefly optimization, ant colony optimization, multi-objective optimization, hybrid optimization method.

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

I. INTRODUCTION

Very Large-Scale Integration (VLSI) is the process of manufacturing an Integrated Circuit (IC) from huge numbers of transistors that are integrated into a single chip. The integrated circuit is usually a tightly packed electronic circuit on a small piece of crystal silicon evaluating tiny millimetres, including active devices, passive devices and their interconnections. The standard design process of VLSI physical design determines the physical position of active circuits and the boundary of the IC within which the circuits are interconnected. Floorplanning is an essential step in the physical design of VLSI. Floorplanning is the process of creating a pre-plan or blueprint for organizing a set of rectangular modules. The rectangular module is a small region in the chip containing a group of interconnected circuits. During the floorplan, the area required for dumping entire transistors required for a chip and the length of wire required for interconnecting the transistors is estimated. The floorplan primarily focuses on minimizing the area and wire length. In addition, heat generated by the circuit can also be estimated, and it needs to be minimized. A good floorplan optimizes these parameters such as area, wire length and heat generation. VLSI floorplanning generation problem is an NP-hard problem, and different optimization techniques can solve it.

Many evolutionary and optimization algorithms are available to optimize the floorplan in terms of reducing area and wire length. The key objective of the floorplan optimization issue is to reduce the events such as required area and wire length between modules. VLSI design is utilized to obtain the optimal solution with minimum time consumption. The floorplan is applied for categorizing into two types: slicing floorplans and non-slicing floorplans. A floorplan is generated by recursively cutting the floorplan horizontally or vertically. The non-slicing floorplan is the one which cannot be generated by cutting the floorplan and joining.

A nature-inspired meta-heuristic algorithm named Lion Optimization Algorithm (LOA) was designed in [1], focusing on optimizing area and wire length. A new optimization algorithm called BCSA (B* tree crossover simulated annealing algorithm) is introduced in [2], which generates floorplans with fixed-outline constraints. In this paper, the dead space and wire length parameters are considered to optimize. In [3], the hybridized PSO-GA (particle swarm optimization-genetic algorithm) algorithm was designed to generate temperature-aware floorplans. A new algorithm named MOFO-FP (multi-objective firefly optimization-based floorplanning) was designed in [4] to reduce the chip's energy consumption and hot spots.

A new algorithm, adaptive particle swarm optimization, was introduced in [5] for obtaining efficient floorplanning, which can minimize the area and wire length. The temperature reduction had not been considered in the algorithm. A multi-objective Lion's pride-inspired algorithm was introduced for optimizing area in [6]. A modified particle swarm optimization approach was proposed in [7], which minimizes the power consumption for VLSI. But, other basic optimization parameters, such as area and wire length, were not focused on.

For optimizing area, wire length and temperature in VLSI circuits, a new floorplanner named Diffusion Oriented Time-improved Floorplanner was designed in [8]. But, this floorplanner failed to address power consumption. A new hybrid technique based on genetic paired mutation was developed in [9] for optimizing VLSI placements. The technique focused on minimizing the execution time, but it failed to address multi-objective constraints.

In [10], a solution for the rectangular floorplan problem was developed. However, it focuses only on minimizing area, whereas other parameters, wire length and temperature, were not discussed. In [11], a hybrid algorithm based on harmony search and a Genetic Algorithm was designed for optimizing area and wire length in VLSI floorplanning. In [12], an evolutionary computation technique-based multi-objective design space exploration framework was designed, which optimizes the area and wire length. An adjusted adaptive algorithm based on symbiosis was designed in [13] to generate an optimal floorplan which occupies a minimum area.

In [14], an algorithm based on simulated annealing was designed to obtain efficient floorplans with minimum peak temperature. Since the primary objective is peak temperature, it could have minimized area and wire length. A nature-inspired optimization algorithm named Multi-Objective Bat Algorithm (MOBA) was developed in [15] for generating efficient floorplans with multiple objectives such as minimizing area, wire length and temperature.

Various heuristic and meta-heuristic approaches were introduced to generate an efficient multi-objective floorplan in [16]. However, the tree-based algorithm was unsuitable for reducing the floorplanning's complexity. In [17], a new conjugate gradient scheme was introduced to enhance the thermal resistance performance for obtaining a multi-objective floorplan.

Based on simulated annealing, a new optimization temperature-aware floorplanning algorithm was developed in [18] to reduce the chip temperature, area and wire length efficiently. However, the efficiency of the produced floorplan was not up to the mark. In [19], a thermal analysis technique was introduced for generating a fixed-outline 3D floorplan. In [20], a flow-based partitioning algorithm was designed, which minimizes the wire length and area cost. But, this algorithm was not suitable for reducing heat and energy consumption.

In [21], a new algorithm based on communication-centric parameters was designed, which uses the Simulated Annealing method for obtaining floorplan with a minimum area. But, other parameters such as wire length and chip peak temperature should have been considered. In [22], the flow-based netlist partitioning algorithm was designed to optimize the area required for the circuit. The floorplan was represented through a graph, and the max flow min cut algorithm was proposed. Based on the design of various functional circuits and learning automata, a new multi-objective optimization algorithm based on learning automata was designed in [23].

In [24], a temperature-aware 3D IC floorplanning algorithm was proposed, which reduces the peak temperature in the 3D chip by placing the hot modules at the bottom of the chip. Since the focus is on reducing thermal, the optimization of the length of the wire and other placement constraints are sacrificed. A novel methodology based on a hyperparameter optimization technique was designed for allocating blocks to tiers in [25]. The algorithm reduces heat generated on the chip by evenly distributing the power through all the blocks. In order to optimize the thermal parameters along with the wire length and area, a new hot floorplanning algorithm was proposed in [26]. The estimation of wire reliability and congestion in routing are also addressed in the proposed work. The recent research on generating floorplan with reduced chip temperature, wire length, and area delivers decent improvements.

Since the number of objectives to be minimized in generating floorplan is being increased, we still have space to improve the circuit performance by reducing peak temperature through designing an exact algorithm by mixing the best essence of different swarm-based meta-heuristic algorithms such as Ant Colony Optimization [27], Particle Swarm Optimization [28] and Firefly Optimization [29] algorithms. The traditional swarm-based optimization algorithms can be combined effectively with various supervised Machine Learning (ML) algorithms, such as Rao Optimization algorithm [30] and Support Vector Machine (SVM) [31], to improve the efficiency and accuracy of the results.

A. METAHEURISTIC OPTIMIZATION

Based on the problem's solution space, the algorithm designed for a problem can be classified as polynomial-time and exponential-time algorithms. Based on the solution and working behaviour of the algorithm, we may classify algorithms as deterministic and nondeterministic. Usually, if we are writing an algorithm for a problem, it is deterministic. It is deterministic if we know clearly each step and how the algorithm works. A nondeterministic algorithm is an algorithm that, even for the same input, can show different behaviours on different runs, contrasting to a deterministic algorithm. When an exact solution is difficult or expensive to develop using a deterministic algorithm, we may go for nondeterministic algorithms, which run on polynomial time to find approximate solutions. For most real-world problems, the solution space is huge and developing a polynomial time deterministic algorithm is impossible. For such problems, the exact solution is difficult to find, but we can find the nearly best solution, which is said to be the optimum solution. The optimum solution is the best solution among all the feasible solutions. The nondeterministic polynomial time algorithm developed for finding an optimum solution is called an optimization algorithm. A heuristic is a technique that aims to solve a problem faster when existing techniques are too slow. The higher-level heuristic technique for finding a sufficiently good solution for the optimization problem is called metaheuristic. Generally, the metaheuristic techniques are classified into nine different groups based on biology, physic, swarm, social, music, chemistry, sport, math [32] & light [33]. A music-based metaheuristic intelligence algorithm, Harmony Search Algorithm (HAS) has been efficiently used for generating honeyword in [34]. In addition, a new physics-based metaheuristic search and optimization method named Optics Inspired Optimization (OIO) has been developed [35]. The meta-heuristic algorithms have been used for finding an optimal solution to large-scale computing problems and have become more popular due to their reduced polynomial time complexity.

Usually, the meta-heuristic algorithms are inspired by nature, the behaviour of insects or animals, nature, evolutionary concepts and natural phenomena types [36]. The classification based on the inspiration of meta-heuristic techniques is shown in Figure 1. They are grouped into five different categories: Evolutionary, Swarm-Intelligence, Physics-based, Nature-inspired and Bio-stimulated.

In this work, the swarm-intelligence algorithm, Ant Colony Optimization (ACO), is combined with the nature-inspired algorithm, Firefly Optimization (FO), to improve the optimum result's efficiency. This hybridized technique improves the correctness and effectiveness of conventional optimization techniques.

II. PROPOSAL METHODOLOGY

In the VLSI domain, floorplanning is one of the crucial phases in physical design. During floorplanning, the relative positions of modules inside the fixed outline are determined. The main objective of floorplanning is not only simply finding the positions of the modules but also reducing the dead space occupied in the chip and the approximate length of the wire required for connecting blocks. In addition to the area and wire length, the heat generated in the chip also depends on the blueprint generated in the floorplanning step. Reducing area causes increases the possibility of short-circuiting, which affects the reliability of the chip. So, while focusing on reducing area and wire length, the reduction of heat generation of the chip has also been considered during floorplanning.

In current research works, some multi-objective optimization was considered. However, in most methods, area and wire length were alone considered as optimization parameters. The impact of energy and heat was not focused on. Also, most of the research works use existing bio-inspired optimization algorithms with some enhancements. However, the hybrid of two or more optimization algorithms is not used predominantly.

In order to improve efficiency, a new technique based on the hybridization of Ant Colony and Firefly Optimization called HMAC-FO is developed to improve the correctness and effectiveness of the conventional optimization techniques. The primary step in floorplanning is to estimate

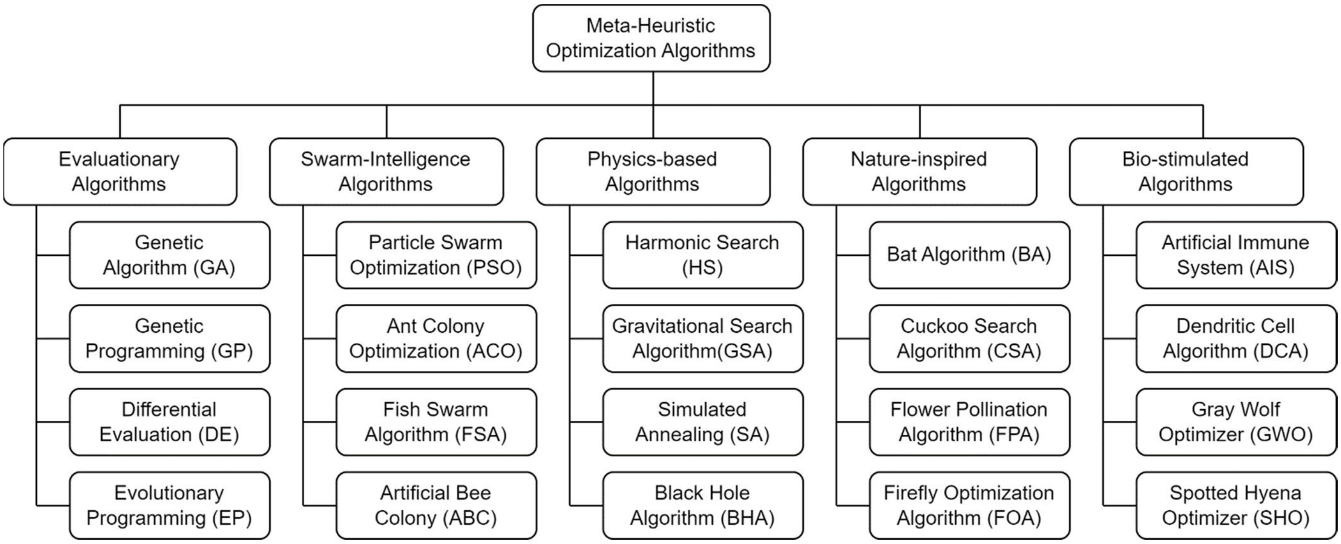
Meta-Heuristic Optimization Algorithms
Evaluationary Algorithms
Genetic Algorithm (GA)
Genetic Programming (GP)
Differential Evaluation (DE)
Evolutionary Programming (EP)
Swarm-Intelligence Algorithms
Particle Swarm Optimization (PSO)
Ant Colony Optimization (ACO)
Fish Swarm Algorithm (FSA)
Artificial Bee Colony (ABC)
Physics-based Algorithms
Harmonic Search (HS)
Gravitational Search Algorithm(GSA)
Simulated Annealing (SA)
Black Hole Algorithm (BHA)
Nature-inspired Algorithms
Bat Algorithm (BA)
Cuckoo Search Algorithm (CSA)
Flower Pollination Algorithm (FPA)
Firefly Optimization Algorithm (FOA)
Bio-stimulated Algorithms
Artificial Immune System (AIS)
Dendritic Cell Algorithm (DCA)
Gray Wolf Optimizer (GWO)
Spotted Hyena Optimizer (SHO)

FIGURE 1. Meta-heuristic optimization technique classification.

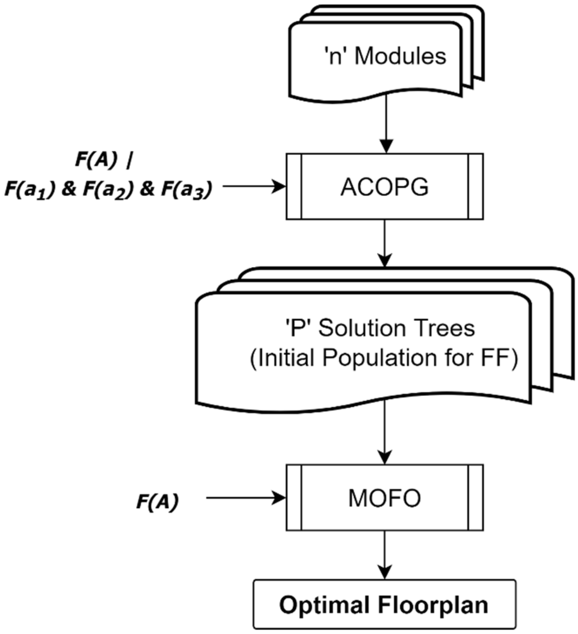
'n' Modules
F(A) / F(a1) & F(a2) & F(a3)
ACOPG
'P' Solution Trees (Initial Population for FF)
F(A)
MOFO
Optimal Floorplan

FIGURE 2. Proposed HMAC-FO technique.

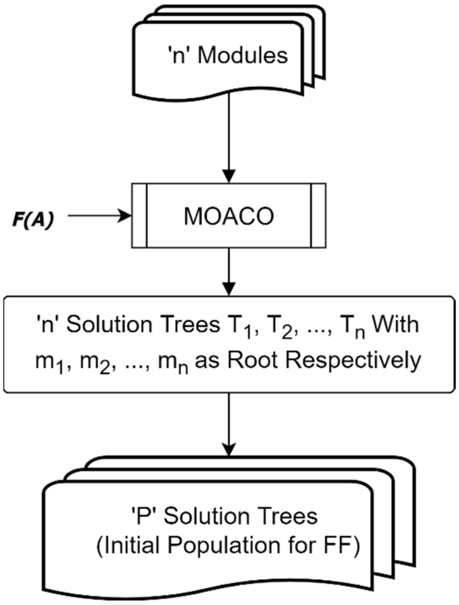
'n' Modules
F(A)
MOACO
'n' Solution Trees T1, T2, ..., Tn With m1, m2, ..., mn as Root Respectively
'P' Solution Trees (Initial Population for FF)

FIGURE 3. Multi-objective population generation.

the approximate chip area and approximate wire length with the consideration of module arrangements and their interconnections. In a floorplanning problem, each module consists of huge numbers of cells that can perform a unique special operation. The proposed HMAC-FO technique is represented through the following block diagram.

The major steps involved in the proposed HMAC-FO technique are given in Figure 2, in which there are two main stages are used to generate an optimal floorplan. The input for the proposed method is ‘n’ modules with full details such as height, weight and area of the modules. The modules’ details are taken from the dataset.

At first, the number of modules is gathered from the dataset. The modules are passed as input for Ant Colony Optimization Population Generator (ACOPG) algorithm. The ACOPG effectively generates ‘n’ numbers of optimal solutions, which are then passed as input for the Multi-objective Firefly Optimization (MOFO) as the initial population of fireflies. The proposed technique mimics the foraging behaviour of ants and the mating behaviour of fireflies. The ants and fireflies are hybridized as the ants help fireflies in selecting brighter fireflies for mating and increase the brightness of fireflies by reducing the distance between them. Fireflies are born through the foraging behaviour of ants. The population generation algorithm has been designed in

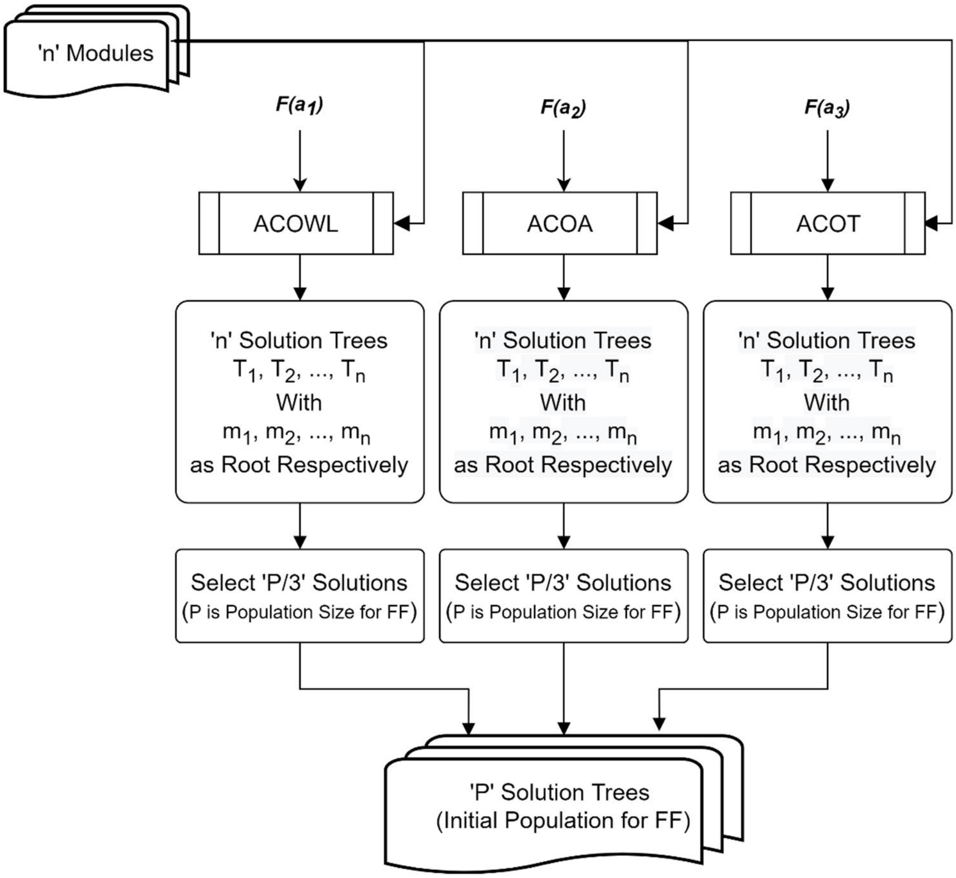
'n' Modules
F(a1)
F(a2)
F(a3)
ACOWL
ACOA
ACOT
'n' Solution Trees T1, T2, ..., Tn With m1, m2, ..., mn as Root Respectively
'n' Solution Trees T1, T2, ..., Tn With m1, m2, ..., mn as Root Respectively
'n' Solution Trees T1, T2, ..., Tn With m1, m2, ..., mn as Root Respectively
Select 'P/3' Solutions (P is Population Size for FF)
Select 'P/3' Solutions (P is Population Size for FF)
Select 'P/3' Solutions (P is Population Size for FF)
'P' Solution Trees (Initial Population for FF)

FIGURE 4. Population generation with variable optimization parameters.

two ways as multi-objective optimized solutions and various single-objective solutions.

A. ANT COLONY OPTIMIZATION POPULATION GENERATOR (ACOPG)

Ant Colony Optimization is a bio-inspired meta-heuristic algorithm used to solve many optimization problems. In the proposed work, the Ant Colony algorithm is used in obtaining initial healthy populations. The Firefly algorithm generates the overall best solution. In the conventional firefly algorithm, the initial population of fireflies are selected randomly. Because of the random selection, it is possible to miss the brightest firefly or all the selected fireflies are dull. Due to this, it is possible to miss the overall best solution. If we select all the fireflies in the initial population, the brighter fireflies, the overall solution can be the best solution. The solution trees obtained from ACO have different root nodes. The ACO is used in generating brighter fireflies, and those are initial healthy solutions. In the proposed work, the ACO has been used effectively in two ways to generate two different sets of populations. The aim of HMAC-FO is to generate a multi-objective optimized floorplan. The algorithm uses three objective parameters to be improved, area, wire length and heat. The ACO is applied in two different ways to obtain the best populations.

In the first method, Ant Colony Optimization Population Generator – Multi-Objective Populations (ACOPG-MOP), ACO is designed to produce ‘n’ populations with enhanced area, wire length and heat. In this method, the populations are multi-optimized solutions.

Figure 3 given above, shows the steps in generating multi-objective solutions. The total ‘n’ modules of the standard benchmark circuit are given as input for the Multi-Objective Ant Colony Optimization (MOACO) algorithm. The multi-objective fitness function F(A) with the area, wire length and thermal as optimization parameters are used in ACO to optimize the different parameters. The result is obtained as ‘n’ numbers of solution trees (B* Tree) which can be used as the initial population for the firefly algorithm.

The second method, ACOPG-Variable Objectives Populations (ACOPG-VOP), is designed to obtain the solutions, which are mixed of single-optimized floorplans with variable optimization parameters such as area, wire length, and heat.

Figure 4 shows the detailed steps in generating various single optimization solutions as the initial population for the firefly algorithm. ACO is applied three times for the same input modules; each version of ACO generates ‘*n*’ solutions individually, the first ‘*n*’ solutions set are area-optimal solutions, the second ‘*n*’ solutions set are wire length optimal solutions, and the last ‘*n*’ solutions set are

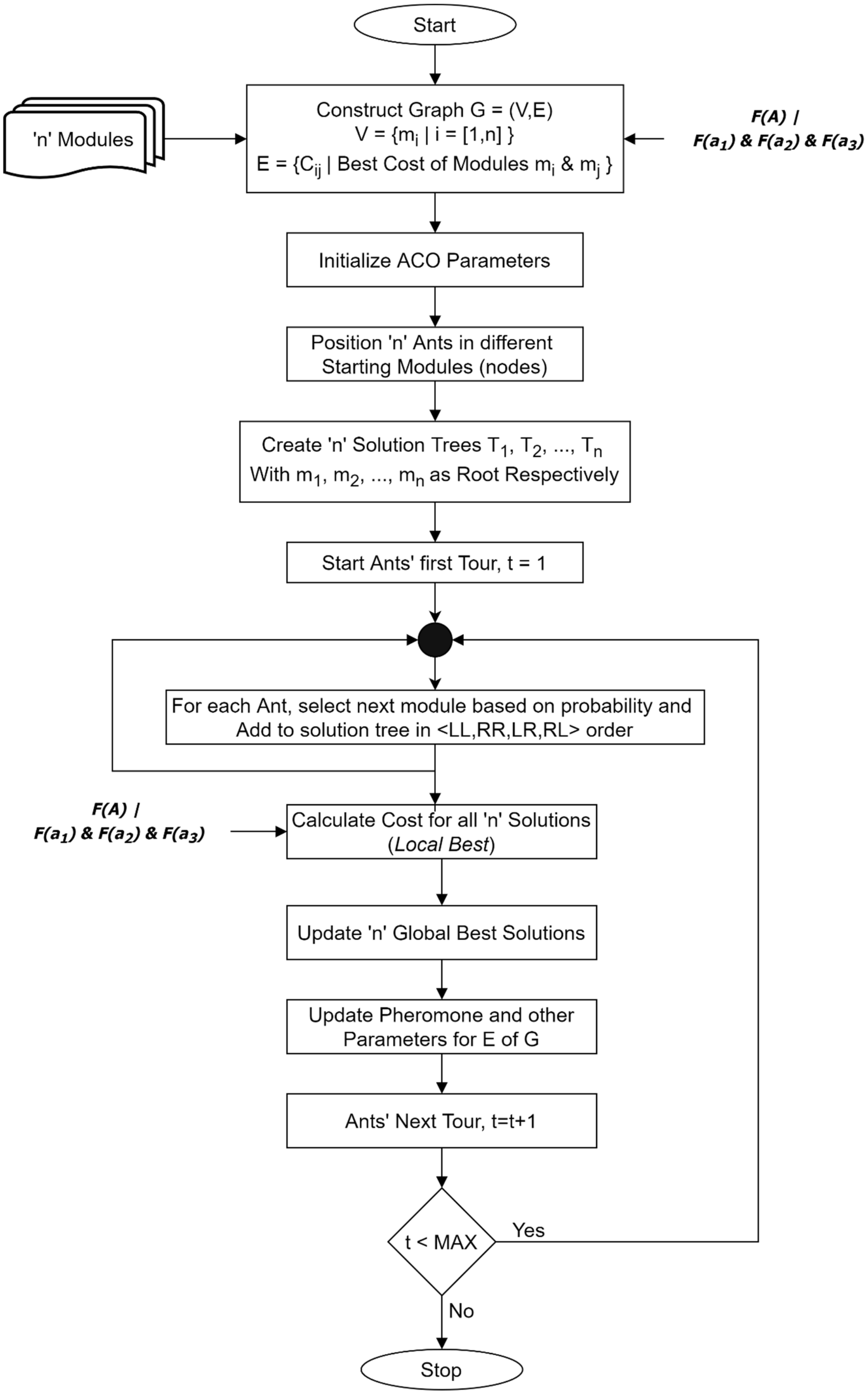
Start
'n' Modules
Construct Graph G = (V,E)
V = {m_i | i = [1,n] }
E = {C_{ij} | Best Cost of Modules m_i & m_j }
F(A) |
$F(a_1)$ & $F(a_2)$ & $F(a_3)$
Initialize ACO Parameters
Position 'n' Ants in different Starting Modules (nodes)
Create 'n' Solution Trees T_1, T_2, ..., T_n With m_1, m_2, ..., m_n as Root Respectively
Start Ants' first Tour, t = 1
For each Ant, select next module based on probability and Add to solution tree in <LL,RR,LR,RL> order
F(A) |
$F(a_1)$ & $F(a_2)$ & $F(a_3)$
Calculate Cost for all 'n' Solutions (Local Best)
Update 'n' Global Best Solutions
Update Pheromone and other Parameters for E of G
Ants' Next Tour, t=t+1
t < MAX
Yes
No
Stop

FIGURE 5. Flowchart for population generation using ACO.

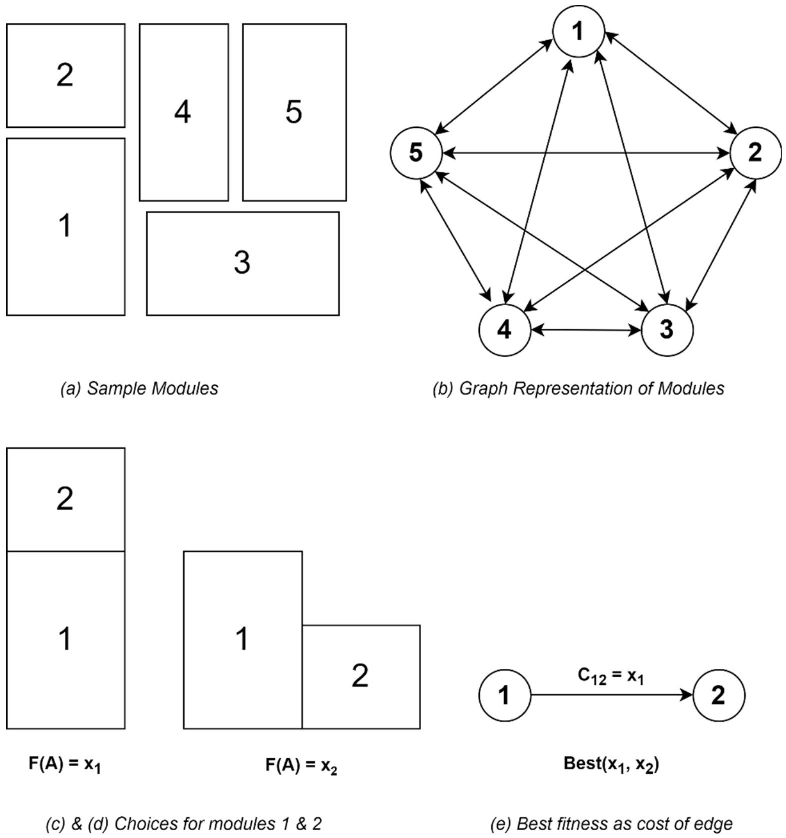
2
4
5
1
3
(a) Sample Modules
1
5
2
4
3
(b) Graph Representation of Modules
2
1
1
2
F(A) = x1
F(A) = x2
(c) & (d) Choices for modules 1 & 2
C12 = x1
1
2
Best(x1, x2)
(e) Best fitness as cost of edge

FIGURE 6. Graph representation of modules.

thermal-optimal solutions. The initial populations for the firefly algorithm are taken from these '*3 x n*' numbers of solutions. Since the firefly algorithm's population size is *P* and we are having '*3 x n*' solutions, the best '*P/3*' solutions are taken as initial healthy population. If we select *P* which is equivalent to '*3 x n*', then all the heathy solutions obtained from ACO will be taken as an initial population for the firefly algorithm so that the chance of missing the brighter firefly can be avoided. There are three fitness functions, $F(a_1)$ to obtain wire length optimal solutions, $F(a_2)$ to obtain area optimal solutions and $F(a_3)$ to obtain thermal-optimal solutions used in the ACO algorithm for generating individual-best solutions.

Figure 5 illustrates the steps in obtaining optimal solutions, which are the initial population for the firefly algorithm to generate a globally optimal floorplan. Ant Colony Optimization was initially applied to solve the famous optimization graph problem named Travelling Salesperson Problem to find the shortest trip for a salesperson who wants to visit all the cities exactly once and return to home city. To apply ACO for the floorplanning problem, the problem needs to map with the graph problem. The input benchmark circuit is first represented as a graph as follows. The graph is represented as a collection of vertices and edges, where the vertices are the modules, and the edges are connecting the modules with the cost of the edge which connects two vertices and is equivalent to the best fitness value of the two modules m_i and m_j, $E = \{C_{ij}|\ Best\ Cost\ of\ Modules\ m_i\ and\ m_j$

The modules are the vertices of the graph, and the best fitness between any two modules is represented as the edge cost between the modules. The cost of the edge is calculated through the fitness function. The constructed graph is a Directed Complete Graph, which is used in finding the optimal floorplan by applying Ant Colony Optimization in a similar way that the TSP problem is solved using ACO.

As per the ACO, the first step is initializing the basic parameters. After that, there are 'n' different ants placed in 'n' different vertices. It means that the algorithm will generate 'n' different floorplans, each having a different starting vertex, i.e., the left bottom corner module in the floorplan. For example, if the number of modules, 'n' is 5, then there are five different floorplans, and each one has a different left bottom corner module, which are obtained from the graph by applying ACO. The solutions are obtained as trees with starting module (or vertex) as the root. The ants start their travel from the starting city, which is the root of solution trees. Each ant will select its next city (module) based on the distance (cost), and it will be added as its child. The child will be added with its parent in a particular order as Left of Left subtree (LL), Right of Right subtree (RR), Left of Right subtree (LR) and Right of Left subtree (RL).

TABLE 1. **Summary of solution space and time complexity.**

Floorplan Representation	Solution Space	Time Complexity
B*-Tree	$O[(n!\,2^{2n-1})/(n\sqrt{n})]$	$O(n)$
O-Tree	$O[(n!\,2^{2n-1})/(n\sqrt{n})]$	$O(n)$
Corner Block List	$O[(n!\,2^{3n-3})/(n\sqrt{n})]$	$O(n)$
Moving Block Sequence	$O[(n!\,2^{2n-1})$	$O(n^2)$
Sequence Pair	$O[(n!)^2]$	$O(n^2)$
Graph	$O[(n!)^2]$	$O(n)$

The main task in floorplanning is finding the proper position for the rectangular blocks without overlapping each other. The basic constraint in placing modules is non-overlapping modules. In the graph, if there is a directed edge between two vertices, it represents there is minimum cost admissible placement exists between the two modules without overlapping modules. The graph construction from the given input modules is shown in Figure 6. There are five sample modules with different widths and heights, and their corresponding graph is shown in Figures 6(a) and (b). The edge cost is calculated through the fitness function, which is illustrated in Figures 6(c) to (e). The two possibilities for module 1 followed by module 2 with fitness values x_1, and $x_{2,}$ respectively. Among these two choices, the best fitness x_1 is identified and is set as the cost for the edge *(1, 2)* in the graph. Similarly, the cost for other edges, such as *(2, 1)*, *(1, 3)*, *(3, 1)*, etc., are also calculated. In reference to other standard floorplan representations such as B* Tree, O-Tree, Corner Block List and Sequence Pair, the Directed Complete Graph representation provides equivalent or better computational complexity. The solution space and time complexity of different representations are summarized in Table 1. For representing a floorplan with n modules, Sequence Pair and Moving Block Sequence (MBS) requires $O(n^2)$ running time, whereas other representations B*-Tree, O-Tree, Corner Block List and the proposed Graph representations require $O(n)$ running time, which is linear time complexity. In conclusion, the Graph representation gives the best performance even for the larger solution space.

The multi-objective fitness function (C) is for computing the cost of the edges involves area, wire length and heat generation as optimization parameters as follows

$$C = \mu_1 \frac{c_1}{c_1{*}} + \mu_2 \frac{c_2}{c_2{*}} + \mu_3 \frac{c_3}{c_3{*}} \tag{1}$$

where C represents the fitness, 'μ_1', 'μ_2' and 'μ_3' represents a constant value whose values lie between 0 to 1. Since there is a trade-off between area, wire-length and temperature, the constant values are assigned equally as $\mu_1 = 0.34$, $\mu_2 = 0.33$ and $\mu_3 = 0.33$. In the above equation (1), the parameters c_1, c_2 and c_3 represents multi-criteria functions such as c_1 is the area, c_2 is the wire length and c_3 denotes heat generation. The average values of area, wire length and heat generation for the randomly selected thousands of floorplans are represented as c_1*, c_2* *and* c_3* respectively. The input for the problem is 'n' number of modules, each of which with different heights and widths. The resultant floorplan will find the relative position of the modules in the larger chip.

Initially, the space occupied by the modules (c_1) is the area of the outer rectangle which includes all the modules and which is calculated as follows

$$c_1 = W \times H \tag{2}$$

With,

$$W = \left(\underset{1\le i\le n}{max}\ (w_i + x_i) - \underset{1\le i\le n}{min}\ (x_i) \right) \tag{3}$$

$$H = \left(\underset{1\le i\le n}{max}\ (w_i + x_i) - \underset{1\le i\le n}{min}\ (x_i) \right) \tag{4}$$

where w_i & h_i are the width and height of the i^{th} module and (x_i, y_i) is the left bottom corner position of the i^{th} module. The area is measured in square millimetres (mm^2).

The estimation of wire required for connecting modules is refered as wire length (c_2) of the circuit, and it is calculated through Half-Perimeter Wire length (HPWL) as follows.

$$c_2 = \Delta X + \Delta Y \tag{5}$$

With,

$$\Delta X = \left(\underset{1\le i\le n}{max}\ (x_i) - \underset{1\le i\le n}{\min}\ (x_i) \right) \tag{6}$$

$$\Delta Y = \left(\underset{1\le i\le n}{max}\ (y_i) - \underset{1\le i\le n}{\min}\ (y_i) \right) \tag{7}$$

where, (x_i, y_i) is the position of the i^{th} module. The wire length is measured in millimetre (mm).

Finally, the estimation of temperature is done with the help of Hotspot, which provides a simple thermal compact model. The modules' temperature depends on the relative position of the modules as well as the position of their neighbouring module. It also depends on the power consumption of the module. In addition to the dimensions of the modules, temperature calculation needs power distribution in each module. For a given power distribution on the floorplan, the modules' temperature is calculated as follows.

$$\begin{bmatrix} T_1 \\ \vdots \\ T_n \end{bmatrix} = \begin{bmatrix} R_{11} & \cdots & R_{1n} \\ \vdots & \ddots & \vdots \\ R_{n1} & \cdots & R_{nn} \end{bmatrix} \begin{bmatrix} P_1 \\ \vdots \\ P_n \end{bmatrix} \tag{8}$$

where the vector P represents the power consumption in the functional modules and the vector T represents the temperature of the modules. By giving a floorplan for a set of modules, the Hotspot tool will give the transfer thermal resistance matrix. The thermal resistance transfer R_{xy} of the module x with respect to y is the increase in the temperature of module x due to the unit of power consumption at the module y as follows.

$$R_{xy} = \frac{\Delta T_{xy}}{P_y} \tag{9}$$

After the construction of the graph, each ant is positioned in some city, which has been selected randomly. The movement of an ant k from a starting city i to the ending city j is based on the following probabilistic transition rule:

$$p_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{k\epsilon A_k}[\tau_{ik}(t)]^{\alpha}[\eta_{ik}]^{\beta}} & if\ j\epsilon A_k \\ 0 & otherwise \end{cases} \tag{10}$$

where η_{ij} is the desirability of the state transition, usually, the value of η_{ij} is $1/d_{ij}$, where d_{ij} is the distance between city i and city j. The amount of pheromone deposited for the transition from state i to j is denoted as $\tau_{ij}(t)$. The two positive parameters α and β are used to control the relative weights of the pheromone trail and of the heuristic visibility. The selected next city will be added to the solution tree, which is used to represent the floorplan. Since the solution is a tree, the decision to be taken is to add the next selected city with the current city as left or right child. There are four possibilities are considered for joining the next city with the current city, Left of Left Subtree (LL), Right of Right Subtree (RR), Left of Right Subtree (LR) and Right of Left Subtree (RL)

Once the tour is completed by each ant, the amount of pheromone on each path will be updated based on $(1-\rho)$. The pheromone decay parameter $(0 < \rho < 1)$ represents the trail evaporation when the ant chooses a city and decides to move. The total number of ants is denoted by n, the length of the tour performed by ant k is represented by L_k and Q is an arbitrary constant. After all the ants complete their tour, the pheromone is required to be updated as follows.

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{11}$$

With,

$$\Delta\tau_{ij}(t) = \sum\nolimits_{k=1}^{n} \Delta\tau_{ij}^{k}(t) \tag{12}$$

$$\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k}, & if\ (i,j)\ \epsilon\ tour_k \\ 0, & Otherwise \end{cases} \tag{13}$$

Once the maximum number of iterations are completed, there are n numbers of global best solution trees are obtained, and these solutions are sent to the Multi-Objective Firefly Optimization (MOFO) algorithm as initial population to find the optimized solution.

B. MULTI-OBJECTIVE FIREFLY OPTIMIZATION (MOFO)

After constructing initial best optimal solutions through ACOPG, the resultant solutions are given as input to the Multi-Objective Firefly Optimization (MOFO) algorithm, which is designed to produce efficient floorplanning. The MOFO is designed based on the flashing behaviours of fireflies. In MOFO, at first, all fireflies are numbered randomly from 1 to n with the solutions trees obtained from ACOPG. The current positions of the fireflies are optimized further in order to enhance the objectives such as area, wire length and heat generation.

The same multi-objective function that is used in the ACOPG algorithm is also used as a multi-objective function in the MOFO algorithm. The light intensity of the fireflies is proportional to this multi-objective function. The relation between the light intensity and objective function is given below,

$$I\ \alpha\ C \tag{14}$$

The attractiveness 'L' of the firefly gets changed with the distance between the firefly $'i'$ and the firefly $'j'$ which is represented as 'd_{ij}'. When the light intensity is reduced, it changes the attractiveness and absorption. The changes in the light intensity with respect to current light intensity is given by the following function.

$$I\left(d_{ij}\right) = I_o e^{-\gamma d_{ij}} \tag{15}$$

Here, the current intensity is 'I_o', and the absorption coefficient is 'γ'. The firefly attractiveness 'L' is represented as,

$$L = L_o e^{-\gamma d^{n}}, n \geq 1 \tag{16}$$

Here, the distance is denoted as 'd', the attractiveness at the distance $d = 0$ is denoted as L_o and 'n' represents the number of fireflies. Euclidean distance between any two fireflies 'f_i' and 'f_j' respectively given as,

$$d_{ij} = \sqrt{\left(f_j - f_i\right)^2} \tag{17}$$

Here, the mathematical formulation for the movement of the firefly 'f_i' with less brightness towards the firefly 'f_j with high brightness is given as,

$$f_i^{q+1} = f_i^{q} + L_o e^{-\gamma d_{ij}^{2}}\left(f_j^{q} - f_i^{q}\right) + \alpha_q \epsilon_i^{q} \tag{18}$$

Here, q is the step number or iteration number, the first term is current fitness of i^{th} firefly, the second term is due to attraction towards j^{th} firefly and the last term is for applying randomness in the movement. The term α_q is randomization parameter whose value is usually in between 0 and 1 and ϵ_i^{q} represents the vector drained from Gaussian or other distribution at time q. If the parameter L_o is used in controlling the attractiveness and if it is 0, the i^{th} firefly will be moved randomly.

The term α_q primarily controls the randomness of the solutions. So, this parameter needs to be controlled in each iteration, and it will be varied with the iteration number q. The parameter α_q is defined as,

$$\alpha_q = \alpha_0 \delta^{q}, 0 < \delta < 1 \tag{19}$$

Here, δ denotes the cooling factor, and the initial randomness is denoted as α_0. The strength of the randomness needs to be controlled, and this will be done by the parameter δ.

In the proposed HMAC-FO technique, the MOFO algorithm performs the firefly optimization with the aim of reducing area, wire length and hot spots in the chip. The light intensity is calculated from the three objective functions such as minimizing space required, minimizing wire length and minimizing heat generation in the chip. In the MOFO algorithm, the fireflies update their new position based on the attractiveness and light intensity of other fireflies. At the end of each iteration, the fireflies are ranked as per their updated light intensity. The step-by-step process is shown in Algorithm 1.

Algorithm 1 Multi-Objective Firefly Optimization (MOFO) Algorithm

```
Algorithm MOFO ( T[1..n], F )
   Input: 'n' numbers of floorplans,
      T[1..n] generated by ACOPG
   Output: Efficient Floorplan, F
Initialize fireflies populations, P[1..n] as follows:
For $i \leftarrow 1$ to $n$ Do
   P[i] ← T[i]
End For
Define multi-objective function $C$ as in (1)
Calculate Light Intensity 'I' from (1) and (14)
Describe absorption coefficient '$\gamma$'
Initialize step count, $q \leftarrow 1$
While $q \leq TMAX$ Do
   For $i \leftarrow 1$ to $n$ Do
     For $j \leftarrow 1$ to $n$ Do
       /* Use the equation (17) to find distance
       between fireflies $i$ and $j$*/
       $d_{ij} \leftarrow$ ComputeDistance(P,i,j)
       /* Update position for fireflies based on intensity */
       If $I_j > I_i$ Then
         /* Move firefly $j$ towards $i$*/
         MoveFirefly(P,j,i)
       End If
       /* Modify the attractiveness based on the
       distance '$d_{ij}'$*/
       ModifyAttractivness(P,i,j, $d_{ij}$)
       /* Update the light intensity after calculating a
       novel solution */
       UpdateIntensity(P,i,j)
       End For
   End For
   /*Based on light intensity, rank the fireflies */
   RankFireflies(P)
   /* Update Iteration */
   q ← q+1
  EndWhile
   /* Return the firefly with best fitness as optimum
   floorplan */
   Return $P_{Optimum}$ as F
End MOFO
```

Algorithm 1 illustrates the process of firefly optimization for generating optimum floorplan in VLSI Design. The list of parameters used in the algorithm are described in Table 2.

At first, the list of initial populations is constructed with the help of the solutions obtained from the ACOPG algorithm. The initial fitness values are calculated for the populated fireflies. The light intensity of the fireflies is calculated through the fitness function defined in equation (1) which is to optimize the area, wire length and heat generation. Once the fitness is calculated, the list of modules is sorted in order to identify the local best solution. The firefly algorithm then accepts these local best solutions to identify the global best solution. The movement of the firefly is decided based on the light intensity. If one firefly's intensity is high, then it will be moved toward the firefly with low intensity. The new positions for the fireflies are updated after movements. The same process is then repeated for a maximum number of times. At last, the ranking of fireflies is done based on the light intensity of the fireflies, which helps in finding the best optimal solution.

TABLE 2. Parameters description.

Parameter	Description
n	Number of fireflies / floorplans
$T[1..n]$	List of 'n' floorplan trees
$P[1..n]$	List of 'n' populations
$I[1..n]$	Intensity of 'n' fireflies
F	Resultant Floorplan
$P_{Optimum}$	Optimum Floorplan
C	Fitness Function as in (1)
q	Step Count
d_{ij}	Distance between fireflies 'i' and 'j'
$TMAX$	Maximum Steps
γ	Randomization Parameter
L_o	Randomization Parameter
α_q	Randomization Parameter
ϵ_i^q	Vector drained from Gaussian

The global optimum solution has been determined through a multi-criteria hybridized optimization algorithm in the proposed technique. The solutions obtained through the hybrid ant colony and firefly algorithms are better than the solutions obtained by applying a single ant colony or single firefly algorithm.

The combination of the firefly algorithm and ant colony optimization technique efficiently determines the global best floorplan, which optimizes more than one parameter, such as area, wire length and heat generation. With multi-criteria optimization, the hybrid approach is to discover the optimal paths in the graphs that represent the floorplan. Based on the behaviour of real ants, a population-based metaheuristic algorithm is designed in the proposed technique, which solves the major issues in multi-criteria optimization. Usually, the real ants are doing searching for their food source, and while searching food, the ants are being moved between different places. During this process, the ants produce a phenomenon which is an organic compound on their path. The pheromone trails are used to make communications be-tween the ants. The deposition of deposition depends on the number of foodstuffs the ant bears. Afterwards, other ants then follow the path, which contains more pheromones, by smelling the deposited pheromone trails. The path followed by the most number of ants is identified as the shorter path.

TABLE 3. Standard MCNC benchmark netlists.

Circuit	Modules	Nets	I/O pad	Pins	Area (mm^2)
apte	9	97	73	287	46.5616
ami33	33	123	42	522	1.1564
ami49	49	408	22	953	35.4454
xerox	10	203	2	698	19.3503
hp	11	83	45	309	8.83

III. EXPERIMENTAL SETUP

The proposed HMAC-FO has been experimented in two different ways, named as HMAC-FO-Multi-Objectives Population (HMAC-FO-MOP) and HMAC-FO-Variable Objectives Populations (HMAC-FO-VOP) based on the way of generating the initial population through ACOPG-MOP and ACOPG-VOP techniques. The proposed algorithms are implemented in MATLAB 2015b with 2.10 GHz Intel Core i5 processor, 8GB RAM, and Windows 10 platform. Experimental evaluation of HMAC-FO-MOP and HMAC-FO-VOP techniques and Line Optimization Algorithm (LOA) [1], B* tree Crossover Simulated Annealing (BCSA) algorithm [2], Hybrid Particle Swarm Algorithm – Genetic Algorithm (HPSOGA) [3] and Multi-Objective Firefly Optimization-based FloorPlanning (MOFO-FP) [4] are implemented with the input data set which is obtained from (https://s2.smu.edu/∼ manikas/Benchmarks/MCNC_Benchmark_Netlists.html). The existing and proposed techniques are simulated with the same software and hardware platform. Since the same experimental conditions are used for the simulation, the comparison of the obtained results is valid. The experiment was repeated for 1000 times, and the average results are recorded in the Tables 3, 4 and 5.

The standard circuits apte, ami33, ami49, Xerox and hp are considered for the testing proposed technique. Among these, the three circuits named apte, Xerox and hp are having fewer numbers of modules, whereas the circuits ami33 and ami49 have large numbers of circuits. The experimental results show how the proposed technique generates an optimal solution for modules of various sizes. The modules are nothing but a group of interconnected transistors called as circuit. The following Table 3 defines the description of MCNC Benchmark circuits.

The standard five circuits such as hp, xerox, ami49, ami33 and apte MCNC Benchmark are described in Table 3.

IV. RESULTS ANALYSIS

The performance results of HMAC-FO-MOP and HMAC-FO-VOP techniques are compared with the results obtained for LOA [1], BCSA [2], HPSOGA [3] and MOFO-FP [4] with various metrics called area used by circuits, required wire length and peak temperature generation. The results prove that the proposed algorithm reduces the area by 3.48%, the wire length by 0.64% and the temperature by 3.33% than the best existing methodologies. Between the two proposed techniques, HMAC-FO-VOP gives much better improvements than HMAC-FO-MOP. Since in the Variable Objectives Population (VOP) method, the populations are obtained as the best solutions for individual objective parameters, between the two proposed techniques, HMAC-FO-VOP gives much better improvements than HMAC-FO-MOP for the circuits with a larger number of modules. The proposed methods provide significant improvements for the circuits with a smaller number of modules. The results obtained for various techniques on various metrics are discussed through the following tables and their graphical representations. The statistics given in the table show the mean values of 1000 runs of simulated algorithms.

The area occupied c_1 by the various MCNC benchmark circuits for various techniques are described in Table 4. The results obtained for the space occupied using six methods namely the proposed techniques HMAC-FO-MOP and HMAC-FO-VOP and the existing LOA [1], BCSA [2], HPSOGA [3] and MOFO-FP [4] are listed in Table 4. The above results indicate that the HMAC-FO-VOP occupies minimum space than other techniques. When applying the HMAC-FO-VOP strategy for the apte circuit, it occupies 46.61 mm^2 of space is occupied whereas the 48.03 mm^2, 48.03 mm^2, 47.44 mm^2 and 46.71 mm^2 are obtained using LOA [1], BCSA [2], HPSOGA [3] and MOFO-FP [4] respectively. While comparing the proposed methods, HMAC-FO-VOP is giving better results for larger circuits such as apte and ami49, whereas HMAC-FO-MOP is giving better results for smaller modules. The experimental results show that the new hybrid approach beats all other competitive techniques.

The experimental result for minimizing area engaged by the modules using six strategies, the proposed techniques HMAC-FO-MOP and HMAC-FO-VOP and the existing LOA [1], BCSA [2], HPSOGA [3] and MOFO-FP [4] is shown in Figure 7. The horizontal direction represents the various circuits and the vertical direction represents the area in mm^2. In the graphical chart, the proposed methods HMAC-FO-MOP and HMAC-FO-VOP are represented in the last two different green colours. In contrast, the first four-colour columns indicate space occupied results of existing [1], [2], [3], and [4], respectively. The graphical result exhibits that the proposed HMAC-FO-VOP approach performs sounder execution for all circuits, whereas HMAC-FO-MOP achieves better performance for smaller circuits. The main reason for this substantial enhancement is due to applying the hybridization of Firefly and Ant Colony Systems. The hybridized technique determines the modules and their positions with lesser space consumption. The proposed algorithm reasonably reduces the dead space in floorplan.

Table 5 shows the estimated length of wire, c_2 (in mm) required for interconnecting modules for the different MCNC circuits. The experimental result is obtained with the ami33 circuit, the wire length of the proposed HMAC-FO-MOP and HMAC-FO-VOP techniques is $\mathbf{48.36}\ \boldsymbol{mm}$ and the results of wire length using LOA [1], BCSA [2],

TABLE 4. AREA occupied by circuits.

Circuit	Area (c_1) (in mm^2) - $\mu_1 = 0.34, \mu_2 = 0.33$ and $\mu_3 = 0.33$					
	LOA	**BCSA**	**HPSOGA**	**MOFOFP**	**HMAC-FO-MOP**	**HMAC-FO-VOP**
apte	48.03	48.05	47.44	46.71	46.69	**46.61**
ami33	1.23	1.23	1.24	**1.16**	**1.16**	**1.16**
ami49	38.45	38.1	38.6	35.64	35.56	**35.49**
xerox	20.5	20.42	20.2	19.65	**19.61**	**19.61**
hp	9.89	9.25	--	8.88	**8.87**	**8.87**
Standard Deviation (σ)	19.81	20.45	19.28	18.76	18.52	18.69

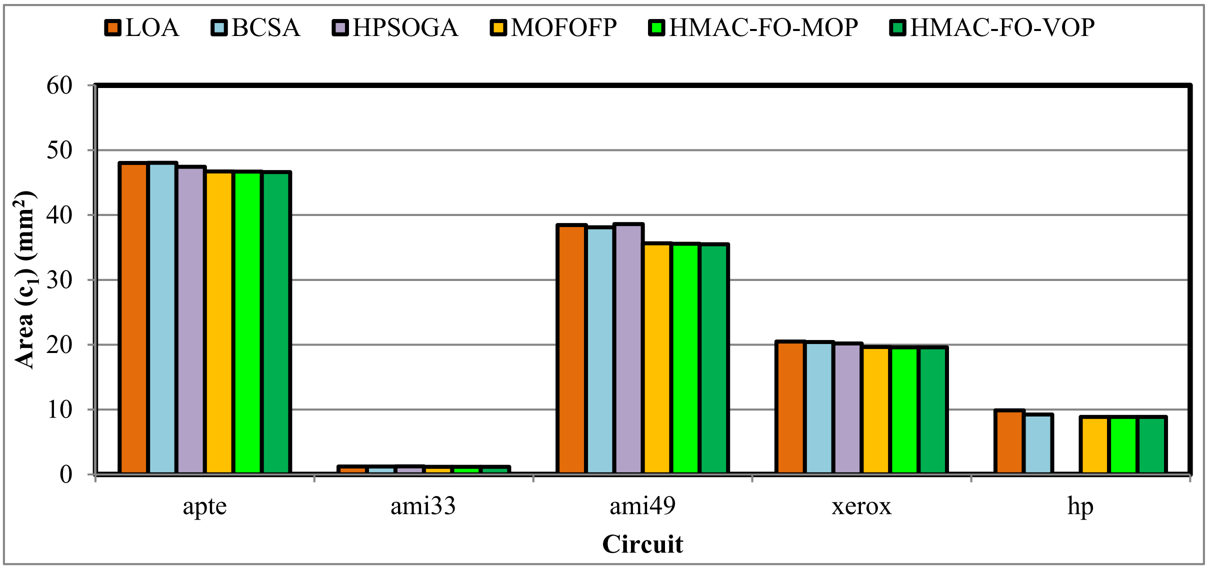

FIGURE 7. Results of area occupied by circuits.

TABLE 5. Wire length estimation for the floorplan.

Circuit	Wire-length (c_2) (in mm) - $\mu_1 = 0.34, \mu_2 = 0.33$ and $\mu_3 = 0.33$					
	LOA	**BCSA**	**HPSOGA**	**MOFOFP**	**HMAC-FO-MOP**	**HMAC-FO-VOP**
apte	316.49	474.62	480	402.78	316.47	**313.42**
ami33	52.22	53.55	51.6	48.41	**48.36**	**48.36**
ami49	822.77	705.2	703	707.91	702.97	**695.95**
xerox	337.04	385.21	502.8	394.04	337.01	**332.96**
hp	142.95	157.07	--	148.1	142.92	**141.89**
Standard Deviation (σ)	258.48	259.12	257.87	256.94	255.34	255.67

HPSOGA [3] and MOFO-FP [4] are observed as $\mathbf{52.22}\ \boldsymbol{mm}$, $\mathbf{53.55}\ \boldsymbol{mm}$ $\mathbf{51.6}\ \boldsymbol{mm}$ and $\mathbf{48.36}\ \boldsymbol{mm}$ respectively. The complete performance results of wire length of the proposed HMAC-FO-MOP and HMAC-FO-VOP techniques is compared with existing methods.

The analysis of required wire length (c_2) for various MCNC circuits is shown in Figure 8. The chart designates that the proposed HMAC-FO-MOP and HMAC-FO-VOP techniques outperform well in obtaining a minimum wire length than the other existing methods. The major reason for obtaining the minimum wire length of the circuit is due to its less dead space. The modules occupy lesser space and minimal wire length due to the hybrid optimization technique.

Table 6 displays the heat generation of the proposed techniques HMAC-FO-MOP and HMAC-FO-VOP and the existing LOA [1], BCSA [2], HPSOGA [3], and MOFO-FP [4]. The heat generated in different circuits with the optimal floorplan obtained through the proposed techniques is described in the table. For every iteration, the input number of modules taken is varied. The experimental result shows

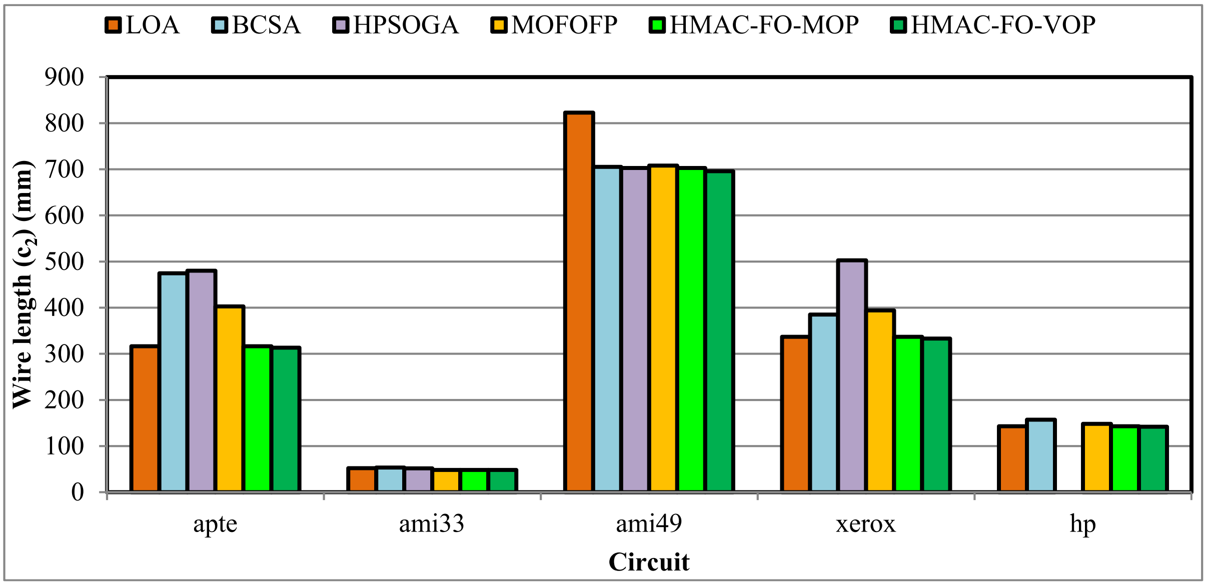

FIGURE 8. Results of required wire length.

TABLE 6. Heat generation by the circuit.

Circuit	Heat Generation (c_3) (in ºC) - $\mu_1 = 0.34, \mu_2 = 0.33$ and $\mu_3 = 0.33$					
	LOA	**BCSA**	**HPSOGA**	**MOFOFP**	**HMAC-FO-MOP**	**HMAC-FO-VOP**
apte	--	--	73.61	66.78	64.54	**63.29**
ami33	--	--	101.14	95.41	90.21	**88.54**
ami49	--	--	66.26	60.87	58.76	**57.18**
xerox	--	--	84.38	79.98	75.65	**73.89**
hp	--	--	--	72.98	69.2	**67.92**
Standard Deviation ($\boldsymbol{\sigma}$)	--	--	15.37	13.39	14.15	13.42

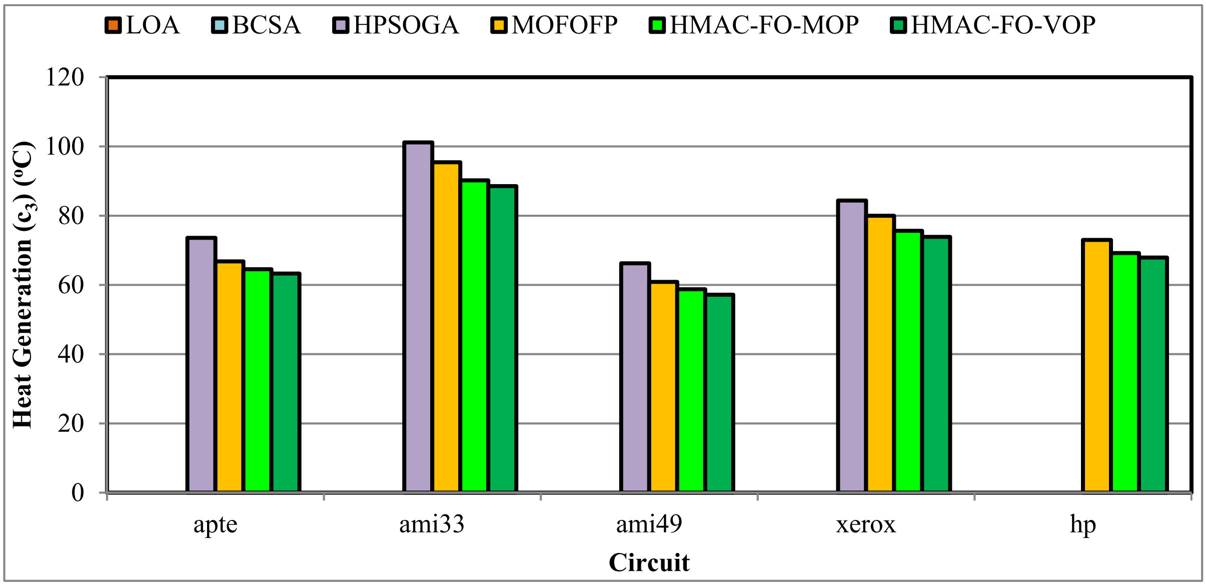

FIGURE 9. Results of heat generation.

that the proposed HMAC-FO-VOP technique generates less heat than the other three conventional techniques. The statistical analysis of the obtained results proves the betterment of the proposed technique. From the circuits

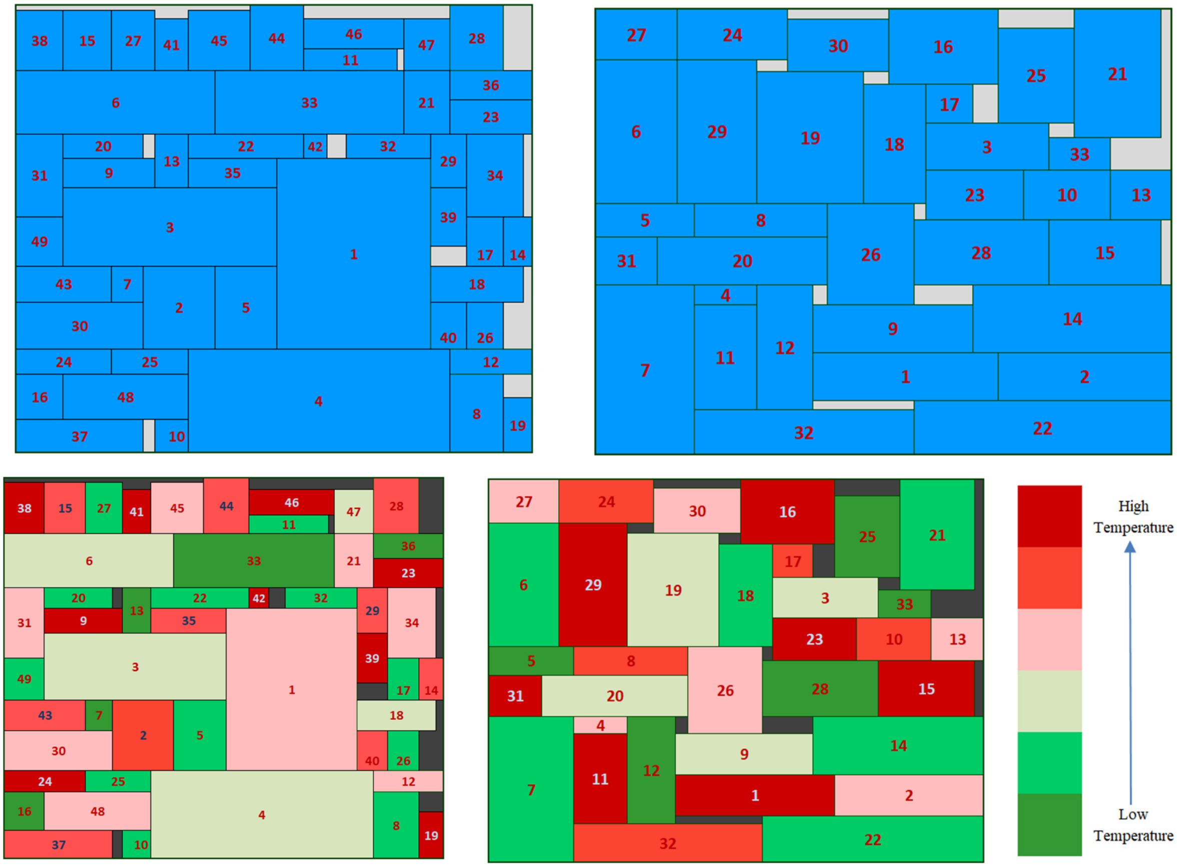

FIGURE 10. Floorplan Layouts of (a)ami49, (b)ami33 and Hot Floorplan Layouts of (c)ami49 (d)ami33.

taken for experiments, the one with the maximum number of modules is ami49 circuit, whose module count is 49. Using the STHMAC-BDFOFP technique, the estimated heat generated in the ami49 circuit is 57.18 °C. In contrast, the heat generated in the same circuit with the existing HPSOGA [3] and MOFO-FP [4] methods are 66.26 °C and 60.87 °C, respectively. The different types of modules' different heat generation experimental results are analyzed for each method. The complete comparison results show that the heat generation is reasonably reduced with the proposed HMAC-FO-VOP technique.

The experimental results of heat generation (c_3) for different methods is interpreted graphically in Figure 9. The vertical axis illustrates the heat generation (in °C) and the horizontal direction indicates various MCNC circuits. The above experimental results prove that the amount of heat generated in HMAC-FO-VOP approach is lesser than the existing HPSOGA [3] and MOFO-FP [4] techniques, respectively. This significant improvement is obtained by applying the hybridized optimization technique. The proposed HMAC-FO-MOP and HMAC-FO-VOP technique uses the hybridization of firefly and ant colony systems.

The floorplan layouts and the corresponding hot floorplan layouts obtained for the larger benchmark circuits ami49 and ami33 are shown in Figure 10. In the hot floorplan layouts, the different colours of the modules represent the amount of heat generated by the modules. From these hot floorplan layouts, it is observed that the modules with a high amount of heat generation are mostly placed at some distance to reduce the effective heat generation of the chip. The total number of modules in ami49 and ami33 are 49 and 33, respectively. The numbered boxes in figure 9 indicate the different blocks of the benchmark circuit, and the area without a number represents the unused space (dead space). The results show that the proposed algorithm reduces dead spaces by 3.48% compared to the existing techniques.

V. CONCLUSION AND FUTURE RESEARCH

A novel meta-heuristic optimization algorithm, HMAC-FO has been designed to generate optimal floorplan. HMAC-FO is designed in two variations based on the objective functions with hybrid optimization techniques. The multiple objectives, such as minimizing space occupied, minimizing wire length and lowest heat generation, are addressed in the generation of the floorplan with the proposed method. Initially, the famous Ant Colony Optimization is applied to generate the population of fireflies. The optimal solutions generated by ACO are used as the initial population for the proposed MOFO algorithm. Hybrid ACO with FO aims to reduce the distance between the selected fireflies. ACO helps in selecting optimal solutions as the initial population required for the MOFO. The hybrid technique is experimented with to analyze the performance of the HMAC-FO-Multi-Objective Populations (HMAC-FO-MOP) and HMAC-FO-Variable Objectives Populations (HMAC-FO-VOP) techniques over the four basic approaches and various performance metrics such as area required for packing circuits, wire length and heat generation. Statistical results prove that HMAC-FO-MOP delivers enhanced performance in reducing the length of wire by 0.52%, the area required by 2.9% and heat generation by 2.15% over the traditional methods and HMAC-FO-VOP delivers enhanced performance in reducing the length of wire by 0.64%, the area required by 3.48% and heat generation by 3.33% over the traditional methods. The hybridization of optimization techniques with deep learning is used to achieve an efficient heat and energy-aware floorplanning. Other than these basic constraints, area, wire length and heat generation, some more parameters can also be considered for the floorplan optimization. One of the main constraints to be considered is placement constraints such as preplace constraint, range constraint, alignment, abutment, clustering, boundary constraint, etc., in placing modules. The proposed work can be extended in future as a unified technique to handle all the placement constraints simultaneously along with the basic optimization parameters.

ACKNOWLEDGMENT

The authors acknowledge SASTRA Deemed University, Thanjavur, for extending infrastructural support to carry out this research work.

REFERENCES

[1] M. Shunmugathammal, C. C. Columbus, and S. Anand, "Nature inspired optimization algorithm for VLSI fixed-outline floorplanning," in *Analog Integrated Circuits and Signal Processing*, vol. 103. Cham, Switzerland: Springer, 2020, pp. 173–186.

[2] M. Shunmugathammal, C. Christopher Columbus, and S. Anand, "A novel B*tree crossover-based simulated annealing," *Circuits, Syst., Signal Process.*, vol. 39, pp. 900–918, Feb. 2020.

[3] P. Sivaranjani and A. Senthil 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, Nov. 2015.

[4] B. Srinivasan and R. Venkatesan, "Multi-objective optimization for energy and heat-aware VLSI floorplanning using enhanced firefly optimization," *Soft Comput.*, vol. 25, no. 5, pp. 4159–4174, Mar. 2021.

[5] S. B. V. Kumar, P. V. Rao, and M. K. Singh, "Optimal floor planning in VLSI using improved adaptive particle swarm optimization," *Evol. Intell.*, vol. 15, pp. 925–938, Jul. 2019.

[6] L. L. Laudis and N. Ramadass, "A Lion's pride inspired algorithm for VLSI floorplanning," *J. Circuits, Syst. Comput.*, vol. 29, no. 1, 2020, Art. no. 2050003.

[7] S. B. V. Kumar, P. V. Rao, H. A. Sharath, B. M. Sachin, U. S. Ravi, and B. V. Monica, "Review on VLSI design using optimization and self-adaptive particle swarm optimization," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 32, no. 10, pp. 1095–1107, 2018.

[8] S. R. Choudhury and S. N. Pradhan, "DOTFloor—A diffusion oriented time-improved floorplanner for macrocells," *IEEE Access*, vol. 7, pp. 172074–172087, 2019.

[9] V. V. Ignatyev, V. A. Kovalev, B. O. Spiridonov, M. V. Kureychik, S. A. Ignatyeva, and B. I. Safronenkova, "Hybrid genetic-paired-permutation algorithm for improved VLSI placement," *ETRI J.*, vol. 43, no. 2, pp. 260–271, 2021.

[10] K. Shekhawat, "Enumerating generic rectangular floor plans," *Autom. Construct.*, vol. 92, pp. 151–165, Aug. 2018.

[11] S. Venkatraman and M. Sundhararajan, "Area and interconnect length optimization for VLSI floor planning problem by using harmony search algorithm," *Int. J. Recent Technol. Eng. (IJRTE)*, vol. 7, no. 6S3, pp. 139–143, 2019.

[12] S. Dey, S. Nandi, and G. Trivedi, "PGOpt: Multi-objective design space exploration framework for large-scale on-chip power grid design in VLSI SoC using evolutionary computing technique," *Microprocessors Microsyst.*, vol. 81, Mar. 2021, Art. no. 103440.

[13] L. L. Laudis, N. Ramadass, S. Shyam, R. Benschwartz, and V. Suresh, "An adaptive symbiosis based metaheuristics for combinatorial optimization in VLSI," *Proc. Comput. Sci.*, vol. 167, pp. 205–212, Jan. 2020.

[14] J. Chen, Y. Liu, Z. Zhu, and W. Zhu, "An adaptive hybrid memetic algorithm for thermal-aware non-slicing VLSI floorplanning," *Integration*, vol. 58, pp. 245–252, Jun. 2017.

[15] L. L. Laudis, S. Shyam, C. Jemila, and V. Suresh, "MOBA: Multi objective bat algorithm for combinatorial optimization in VLSI," *Proc. Comput. Sci.*, vol. 125, pp. 840–846, Jan. 2018.

[16] S. N. Hussain and K. H. Kishore, "Performance evaluation of heuristic algorithms in floor planning for ASIC design," *Int. J. Eng. Technol.*, vol. 7, nos. 1–5, pp. 56–61, 2018.

[17] Z. Jiang and N. Xu, "HotSpot thermal floorplan solver using conjugate gradient to speed up," *Mobile Inf. Syst.*, vol. 2018, pp. 1–8, Apr. 2018.

[18] T. Ni, H. Chang, S. Zhu, L. Lu, X. Li, Q. Xu, H. Liang, and Z. Huang, "Temperature-aware floorplanning for fixed-outline 3D ICs," *IEEE Access*, vol. 7, pp. 139787–139794, 2019.

[19] Q. Xu and S. Chen, "Fast thermal analysis for fixed-outline 3D floorplanning," *Integration*, vol. 59, pp. 157–167, Sep. 2017.

[20] K. Blutman, H. Fatemi, A. Kapoor, A. B. Kahng, J. Li, and J. P. de Gyvez, "Logic design partitioning for stacked power domains," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 11, pp. 3045–3056, Nov. 2017.

[21] J. T. John, N. Dahare, B. Chaithanya, and J. Reuben, "Communication centric floorplanning of NoC based system on chip," *Proc. Comput. Sci.*, vol. 93, pp. 259–268, Jan. 2016.

[22] A. Patil, "Flow based logic design partitioning," *Int. J. Eng. Res. Technol. (IJERT)*, vol. 9, no. 7, pp. 1419–1424, 2020.

[23] N. S. Shahraki and S. H. Zahiri, "An improved multi-objective learning automata and its application in VLSI circuit design," *Memetic Comput.*, vol. 12, no. 2, pp. 115–128, Jun. 2020.

[24] T. Ni, H. Chang, S. Zhu, L. Lu, X. Li, Q. Xu, H. Liang, and Z. Huang, "Temperature-aware floorplanning for fixed-outline 3D ICs," *IEEE Access*, vol. 7, pp. 139787–139794, 2019.

[25] J.-M. Lin, W.-Y. Chang, H.-Y. Hsieh, Y.-T. Shyu, Y.-J. Chang, and J.-M. Lu, "Thermal-aware floorplanning and TSV-planning for mixed-type modules in a fixed-outline 3-D IC," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 9, pp. 1652–1664, Sep. 2021.

[26] 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.

[27] H. Abubakar, A. Muhammad, and S. Bello, "Ants colony optimization algorithm in the Hopfield neural network for agricultural soil fertility reverse analysis," *Iraqi J. for Comput. Sci. Math.*, vol. 3, pp. 32–42, Jan. 2022.

[28] A. H. Raed, S. N. Shahab, and M. A. Abdullah, "Correlation with the fundamental PSO and PSO modifications to be hybrid swarm optimization," *Iraqi J. Comput. Sci. Math.*, vol. 2, no. 2, pp. 25–32, 2021.

[29] J. Wang, M. Zhang, H. Song, Z. Cheng, T. Chang, Y. Bi, and K. Sun, "Improvement and application of hybrid firefly algorithm," *IEEE Access*, vol. 7, pp. 165458–165477, 2019.

[30] S. N. Abd, M. Alsajri, and H. R. Ibraheem, "Rao-SVM machine learning algorithm for intrusion detection system," *Iraqi J. Comput. Sci. Math.*, vol. 1, no. 1, pp. 23–27, Jan. 2020.

[31] N. Khan, I. Khaleel, and E. Daghighi, "Improved feature selection method for features reduction in intrusion detection systems," *Mesopotamian J. CyberSecurity*, vol. 2021, pp. 9–15, Jan. 2021.

[32] S. Akyol and B. Alatas, "Plant intelligence based metaheuristic optimization algorithms," *Artif. Intell. Rev.*, vol. 47, no. 4, pp. 417–462, May 2017.

[33] B. Alatas and H. Bingol, "Comparative assessment of light-based intelligent search and optimization algorithms," *Light Eng.*, vol. 28, no. 6, pp. 51–59, 2020.

[34] R. T. Rasheed, Y. Niu, and S. N. Abd, "Harmony search for security enhancement," *Mesopotamian J. CyberSecur.*, vol. 2021, pp. 5–8, Jan. 2021.

[35] H. Bingol and B. Alatas, "Chaos based optics inspired optimization algorithms as global solution search approach," *Chaos, Solitons Fractals*, vol. 141, Dec. 2020, Art. no. 110434.

[36] A. Kumar and S. Bawa, "A comparative review of meta-heuristic approaches to optimize the SLA violation costs for dynamic execution of cloud services," *Soft Comput.*, vol. 24, no. 6, pp. 3909–3922, Mar. 2020.

B. SRINIVASAN received the B.E. degree in computer science and engineering from Bharathidasan University, India, in 2004, and the M.Tech. degree in advanced computing from SASTRA Deemed University, India, in 2007, where he is currently pursuing the Ph.D. degree in computer science. Since June 2007, he has been with the School of Computing–IT, SASTRA Deemed University, where he is also an Assistant Professor-III. His current research interests include algorithm design, optimization techniques, and network security. He is a Life Member of the Indian Society of Systems for Science and Engineering (ISSE), India.

R. VENKATESAN received the M.C.A. degree from Indira Gandhi National Open University (IGNOU), India, in 2005, and the M.E. degree in computer science and engineering and the Ph.D. degree in computer applications from Anna University, Chennai, India, in 2009 and 2017, respectively. In 2010, he joined the Department of Information Technology, Tejaa Shakthi Institute of Technology for Women, as a Senior Lecturer, and later he joined as an Assistant Professor with the Department of Computer Science and Engineering, Excel Engineering College, affiliated to Anna University. Since June 2016, he has been with the School of Computing - IT, SASTRA Deemed University, where he is an Assistant Professor. His current research interests include network security and data mining. He is the Life Member of Indian Society for Technical Education (ISTE), the Indian Society of Systems for Science and Engineering (ISSE), India. He is an Oracle Certified Associate (OCA).

BELQASEM ALJAFARI (Member, IEEE) received the master's degree in electrical engineering from Northern Illinois University, DeKalb, IL, USA, in 2016, and the Ph.D. degree in electrical engineering from the University of South Florida, Tampa, FL, USA, in 2019. He is currently an Assistant Professor with the Department of Electrical Engineering, Najran University, Saudi Arabia. His research interests include power and renewable energy, energy storage, solar cells, and nano materials.

KETAN KOTECHA is currently an Administrator and a Teacher of deep learning. He has expertise and experience in cutting-edge research and projects in artificial intelligence and deep learning for the last 25 years. He has published more than 100 widely in several excellent peer-reviewed journals on various topics ranging from cutting edge artificial intelligence, education policies, teaching-learning practices, and artificial intelligence for all. He has also published three patents and delivered keynote speeches at various national and international forums, including at the Machine Intelligence Laboratory, Rochester, NY, USA, IIT Bombay under the World Bank Project, and the International Indian Science Festival organized by the Department of Science and Technology, Government of India. His research interests include artificial intelligence, computer algorithms, machine learning, and deep learning. He was a recipient of the two SPARC projects worth INR 166 lakhs from MHRD, Government of India, in artificial intelligence, collaboration with Arizona State University, Tempe, AZ, USA, and The University of Queensland, Australia. He was also a recipient of numerous prestigious awards, such as Erasmus + Faculty Mobility Grant to Poland, the DUO-India Professors Fellowship for research in responsible AI, in collaboration with Brunel University, U.K., the LEAP Grant at Cambridge University, U.K., the UKIERI Grant with Aston University, U.K., and the Grant from the Royal Academy of Engineering, U.K., under Newton Bhabha Fund. He is an Associate Editor of IEEE ACCESS journal.

V. INDRAGANDHI received the B.E. degree in electrical and electronics engineering from Bharathidasan University, in 2004, and the M.E. degree (Hons.) in power electronics and drives and the Ph.D. degree from Anna University, in 2015. She is currently a Professor with the School of Electrical Engineering, Vellore Institute of Technology (VIT), Vellore, Tamil Nadu. She has organized many national and international workshops/symposiums/conferences in collaboration with top leading universities. She has been engaged in research work for the past 15 years in the area of the Internet of Things, smart grid, power electronics, cyber-physical systems, and renewable energy systems. She was awarded the Gold Medal for the achievement of university first rank in her M.E. degree.

SUBRAMANIYASWAMY VAIRAVASUNDARAM received the B.E. degree in computer science and engineering from Bharathidasan University, India, the M.Tech. degree in information technology from the Sathyabama Institute of Science and Technology, India, and the Ph.D. degree from Anna University. After his Ph.D. degree, he continued the extension work with the Department of Science and Technology support as a Young Scientist Award Holder. He is currently a Professor with SASTRA Deemed University, Thanjavur, India. He has 18 years of experience in academia. He is also a Research Supervisor. He is a visiting expert to various universities in India. He has contributed more than 160 articles and chapters for many high-quality Scopus and SCI/SCIE indexed journals and books. His research interests include recommender systems, social networks, the Internet of Things, information security, and big data analytics. He is on the reviewer board of several international journals and has been a program committee member for several international/national conferences and workshops. He also serves as the guest editor for various special issues of reputed international journals.