International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V71I5P225

Volume 71 Issue 5, 240-250, May 2023

Original Article

Effective K-Way Partitioning of VLSI Circuits with Hetero-Homo Status Based Models using Evolutionary Computation

P. Rajeswari[1], Theodore S Chandra[2], Manoj Kumar Singh[3]

[1]*Department of ETE, Dayananda Sagar College of Engineering, Karnataka, India.*
[2]*Department of ECE, Dayananda Sagar University, Karnataka, India.*
[3]*Manuro Tech Research Pvt Ltd, Karnataka, India.*

[1]*Corresponding Author: prajeswarisugans@gmail.com*

Received: 10 February 2023 Revised: 29 April 2023 Accepted: 06 May 2023 Published: 25 May 2023

Abstract *- In this paper, the circuit-level partitioning problem in VLSI has been considered. The concern objectives of partitioning have been considered in terms of minimizing the number of interconnections between partitions as well as satisfying the desired area of each partition. The need for k-way partitioning has also been satisfied without further computational cost. The problem has been solved through the heuristic approach. Based on the natural process, the new approach in the standard Genetic Algorithm has been included to define the selection process of parents. The proposed model presented the concept of hetero-homo status-based group formation to define parent selection. This model eliminates the biased nature of the standard tournament selection process. Based on the natural extinction process, an extinction operator has also been introduced. The proposed model has shown the relative benefit of the extinction operator against the standard genetic algorithm. The presented hetero-homo status-based group mechanism, along with the extinction operator, has shown benefits in terms of faster convergence with better solution exploration. Comparison of performances of proposed hetero-homo group model in association with extinction operator has shown against the dynamic weighted particle swarm optimization as well as different variation of standard genetic algorithm, and supremacy observed.*

Keywords *- VLSI circuit partitioning, VLSI physical design, K-way partitioning, Genetic algorithm, Extinction operator.*

1. Introduction

The Electronic Design Automation (EDA) industry develops software to support engineers in creating new integrated-circuit (IC) designs. Due to the high complexity of modern designs, EDA touches almost every aspect of the IC design flow, from high-level system design to fabrication. EDA addresses designers' needs at multiple levels of the electronic system hierarchy, including integrated circuits (ICs), multi-chip modules (MCMs), and printed circuit boards (PCBs). EDA tools have always been geared toward automating the entire design process and linking the design steps into a complete design flow. However, such integration is challenging since some design steps need additional degrees of freedom, and scalability requires tackling some steps independently.

On the other hand, the continued decrease of transistor and wire dimensions has blurred the boundaries and abstractions that separate successive design steps – physical effects such as signal delays and coupling capacitances need to be accurately accounted for earlier in the design cycle. Thus, the design process moves from a sequence of atomic (independent) steps toward a deeper level of integration.

During physical design, all design components are instantiated with their geometric representations. In other words, all macros, cells, gates, transistors, etc., with fixed shapes and sizes per fabrication layer, are assigned spatial locations (placement) and have appropriate routing connections (routing) completed in metal layers. Physical design directly impacts circuit performance, area, reliability, power, and manufacturing yield. Examples of these impacts are given as (i) Performance: Long routes have significantly longer signal delays. (ii) Area: placing connected modules far apart results in larger and slower chips. (iii) Reliability: a large number of vias can significantly reduce the reliability of the circuit. (iv) Power: transistors with smaller gate lengths achieve greater switching speeds at the cost of higher leakage current and manufacturing variability; larger transistors and longer wires result in greater dynamic power dissipation. (v) Yield: wires routed too close together may decrease yield due to electrical shorts occurring during

manufacturing, but spreading gates too far apart may also undermine yield due to longer wires and a higher probability of opens. Due to its high complexity, physical design is split into several key steps. (i) Partitioning: breaks up a circuit into smaller which can each be designed or analyzed individually. (ii) Floorplanning determines the shapes and arrangement of subcircuits or modules and the locations of external ports and IP or macroblocks. (iii) Power and ground routing, often intrinsic to floorplanning, distributes power (VDD) and ground (GND) nets throughout the chip. (iv) Placement finds the spatial locations of all cells within each block. (v)Clock network synthesis determines the buffering, gating (e.g., for power management) and routing of the clock signal to meet prescribed skew and delay requirements. (vi) Global routing allocates routing resources that are used for connections; example resources include routing tracks (vii) Detailed routing assigns routes to specific metal layers and routing tracks within the global routine ment and routing techniques. (viii)Timing closure optimizes circuit performance through specialized placement and routing techniques.

The efficient designing of any complex system necessitates decomposing the same into a set of smaller subsystems. Subsequently, each subsystem can be designed independently and simultaneously to speed up the design process. The process of decomposition is called partitioning. Partitioning efficiency can be enhanced within three broad parameters. First of all, the system must be decomposed carefully so that the original functionality of the system remains intact. Secondly, an interface specification is generated during the decomposition, which is used to connect all the subsystems. The system decomposition should ensure the minimization of the interface interconnections between any two subsystems. Finally, the decomposition process should be simple and efficient so that the time required for the decomposition is a small fraction of the total design time.

Further partitioning may be required in the events where the size of a subsystem remains too large to be designed efficiently. Thus, partitioning can be used in a hierarchical manner until each subsystem created has a manageable size. Partitioning is a general technique and finds application in diverse areas. For example, in algorithm design, the divide and conquer approach is routinely used to partition complex problems into smaller and simpler problems. The increasing popularity of parallel computation techniques brings in its fold promises in terms of the provision of innovative tools for the solution of complex problems by combining partitioning and parallel processing techniques. Partitioning plays a key role in designing a computer system in general, particularly VLSI chips. A computing system is comprised of tens of millions of transistors. It is partitioned into several smaller modules/blocks to facilitate the design process. Each block has terminals located at the periphery that are used to connect the blocks. The connection is specified by a netlist, which is a collection of nets. A net is a set of terminals which have Tobe made electrically equivalent. A VLSI system is partitioned at several levels due to its complexity. At the highest level, it is partitioned into a set of subsystems whereby each subsystem can be designed and fabricated independently on a single PCB.

High-performance systems use MCMs instead of PCBs. At this level, the criterion for partitioning is functionality, and each PCB serves a specific task within a system.

Consequently, a system consists of I/O (input /output) boards, memory boards, a motherboard (which hosts the microprocessor and its associated circuitry), and networking boards. Partitioning a system into PCBs enhances the design efficiency of individual PCBs. Due to a clear definition of the interface specified by the netlist between the subsystems, all the PCBs can be designed simultaneously. Hence, significantly speeding up the design process. If the circuit assigned to a PCB remains too large to be fabricated as a single unit, it is further partitioned into subcircuits such that each sub-circuit can be fabricated as a VLSI chip. However, the layout process can be simplified and expedited by partitioning the circuit assigned to a chip into even smaller sub-circuits. The partitioning process of a process into PCBs and PCB into VLSI chips is physical in nature. That is, this partitioning is mandated by the limitations of the fabrication process.

In contrast, the partitioning of the circuit on a chip is carried out to reduce the computational complexity arising due to the sheer number of components on the chip. The partitioning of a system into a group of PCBs is called the system-level partitioning. The partitioning of a PCB into chips is called board-level partitioning, while partitioning a chip into smaller subcircuits is called chip-level partitioning.

A popular approach to decreasing modern integrated circuits' design complexity is partitioning them into smaller *modules*. These modules can range from a small set of electrical components to fully functional integrated circuits (ICs). The partition divides the circuit into several subcircuits (partitions or blocks) while minimizing the number of connections between partitions, subject to design constraints such as maximum partition sizes and maximum path delay. Suppose each block is implemented independently, i.e., without considering other partitions. In that case, connections between these partitions may negatively affect the overall design performance, such as increased circuit delay or decreased reliability.

Moreover, a large number of connections between partitions may introduce inter-block dependencies that hamper design productivity. Therefore, the primary goal of partitioning is to divide the circuit such that the number of

connections between subcircuits is minimized. Each partition must also meet all design constraints. A *cell* is any logical or functional unit built from components. A *partition* or *block* is a grouped collection of components and cells. The *k-way partitioning* problem seeks to divide a circuit into *k* partitions. It is possible to show that the partitioning problem can be abstracted using a graph representation, where nodes represent cells, and edges represent connections between cells.

A better circuit partition will reduce connection among subcircuits and result in a better routing area of the layout. The challenge is that the circuit partitioning problem belongs to the class of well-known NP-hard optimization problems, so solving them with optimal, worst-case polynomial-time algorithms is unlikely. Many heuristics have been developed for these problems, the quality of which can be assessed based on (1) runtime and (2) solution quality, measured by sub-optimality (different from optimal solutions). Heuristic algorithms can be classified as Deterministic: all decisions made by the algorithm are repeatable, i.e., not random. Stochastic: some decisions made by the algorithm are made randomly, e.g., using a pseudo-random number generator. Thus two independent runs of the algorithm will produce two different solutions with high probability. In terms of structure, a heuristic algorithm can be Constructive: the heuristic starts with an initial, incomplete(partial) solution and adds components until a complete solution is obtained. Iterative: the heuristic starts with a complete solution and repeatedly improves the current solution until a preset termination criterion is reached.

2. Related Work

Researchers have proposed their theories to partition circuits in the past. The work of [1,24] proposed a hardware genetic algorithm by developing GA and a local search processor that uses some external memory to overcome the problem of local maxima/minima. [1] has expressed the probability of chromosome selection as a function of both the best and worst chromosomes. [2] has proposed two GA, one based on 0-1 encoding and the other based on integer encoding. Work done in [3] developed an adaptive strategy for partitioning circuits in which population size, crossover rate, and mutation rate are modified during the execution in order to enhance performance. The partitioning problem can be viewed as a graph partitioning problem where each module (gates etc.) is taken as vertices, and the connection between them represents the edges between the nodes [4]. The basic foundation of the algorithm is to represent each vertex in the graph as a location that can represent a logic gate and a connection is represented by an edge [5]. The memetic algorithm [6, 7] is a combination of an Evolutionary Algorithm (EA) and Local Search (LS). The EAs are used for finding the global optimum. The LS used here will aid the EA in convergence speed. [8] has proposed a clustering method which can reduce the size of large-scale partitioning problems without losing partitioning solution qualities. The performance of the proposed clustering algorithm is evaluated on a standard set of partitioning benchmarks-ISPD98 benchmark suite.multiobjective hypergraph-partitioning algorithms have been proposed in [9] based on the multilevel paradigm, which can produce solutions in which both the cut and the maximum subdomain degree are simultaneously minimized. A Discrete Particle Swarm Optimization (DPSO) algorithm has been proposed in [10,23]for the optimization of VLSI interconnection (netlist) bipartition.Memetic Algorithm (MA) is an evolutionary algorithm that includes one or more local search phases within its evolutionary cycle. MA has applied in [11] to some sort of local search for optimization of VLSI partitioning. Various methods proposed in passed have been presented in [12]. The technique has been proposed in [26] using the adjacency matrix of a graph for the layer-assignment problem. [14] have presented a swarm-based heuristic approach for solving balanced min cut circuit partitioning, the very first step of VLSI physical design automation. [15] have studied the various heuristic approaches towards circuit partitioning problems. They have also given a comparative analysis of different algorithms which have been proposed. 3D integrated circuits (3D-ICs) are an emerging technology with lots of potential. 3D-ICs enjoy small footprint areas and vertical interconnections between different dies, allowing shorter wire lengths among gates. Hence, they exhibit both lesser interconnect delays and power consumption. The design flow of 3D integrated circuits consists of many steps, the first of which is the 3D Partitioning and Layer Assignment. This step has significant importance as its outcome will influence the performance of subsequent steps. Like other partitioning problems, this one is also an NP-hard. Factors such as layer assignment, location of I/O terminals, TSV minimization, and area balancing always define partitioning quality. Tabu Search and Simulated Annealing have been employed in [16] to achieve these objectives. [17] has presented an overview of getting a minimum cut using a Discrete Particle Swarm Optimization (DPSO) algorithm and a swarm-based heuristic approach called the Discrete Fire Fly Algorithm(DFFA). [27] has proposed a partitioning algorithm for a 3D floorplan. The proposed method combined a cost-based heuristic and force-directed algorithm, which places the nodes considering attractive and repulsive forces to solve the long net problem. Hypergraph partitioning has a wide range of important applications, such as VLSI design or scientific computing. With a focus on solution quality, [19] has proposed a multilevel memetic algorithm to tackle the problem. Key components of contribution were new effective multilevel recombination and mutation operations that provide a large amount of diversity. In [20], a new graph partitioning problem is introduced and transform the problem into a Depth-bounded Leveled Graph Partitioning (DLGP) problem, which is solved optimally using a dynamic programming algorithm. As an example application has shown that DLGP can

effectively generate timing-correct circuit solutions for Single Flux Quantum (SFQ) logic, which is a magnetic-pulse-based, gate-level pipelined superconductive computing fabric. Combined with the idea of the improved KL algorithm and the multilevel partitioning algorithm, a multilevel circuit partitioning algorithm based on the improved KL algorithm has been proposed in [21]. The algorithm is based on the theory of the balanced bipartition theory of graphs, which has improved by giving a reasonable initial partition.

3. Problem Formulation

The partitioning problem of the VLSI circuit can be transformed in the domain of graph theory. A hypergraph $G = (V, E)$ representing a partitioning problem can be constructed as follows. Let $V = \{v1, v2, v3, \dots . vn\}$ be a set of vertices and $E = \{e_1, e_2, e_3, \dots \dots . . e_m\}$ be a set of hyperedges. Each vertex represents a component. This is a hyperedge joining the vertices whenever the components corresponding to these vertices are to be connected. Thus each hyperedge is a subset of the vertex set, i.e.,$e_i \subseteq V, i = 1,2, \dots m$. In other words, each net is represented by a hyperedge. The area of each component is denoted as $a(v_i), 1 \leq i \leq n$.Modelling the partitioning problem into hypergraphs allows us to represent the circuit partitioning problem completely as a hypergraph partitioning problem. The partitioning problem is to partition V into $V_1, V_2, \dots \dots V_k$, in such a manner that it satisfied the following conditions.

$$V_i \cap V_j = \emptyset, i \neq j \text{ and } \bigcup_{i=1}^{k} V_i = V$$

Partition is also referred to as a *cut.* The cost of partition is called the cutsize, which is the number of hyperedges crossing the cut. LetC_{ij} be the cutsize between partitions V_i and V_j. Each partition has an area $Area(V_i) = \sum_{v \in V_i} a(v)$ and a terminal count $Count(V_i)$ The maximum and the minimum areas that a partition V_i can occupy, are denoted as A_i^{max} andA_i^{min} respectively. The maximum number of terminals that a partition V_i can have is denoted as T_i . Let $P = \{p_1, p_2, \dots . p_m\}$ be a set of hyperpaths. Let $H(p_i)$ be the number of times a hyperpath p_i is cut, and let $K_{min}$and K_{max}represent the minimum and the maximum number of partitions allowed for a given subcircuit.

The constraints and the objective functions for the partitioning algorithms vary for each level of partitioning and each of the different design styles used. This makes it very difficult to state a general partitioning problem which is applicable to all levels of partitioning or all design styles used. Hence in this section, we will list all the constraints and the objective functions and the level to which they are applicable. The partitioning problem at any level or design style deals with one or more parameters like interconnections between partitions; delay due to partitioning; the number of terminals, the area of each partition, and the number of partitions. In this work, parameters, the interconnection between partitions and desired area of partitions have been considered in the objective function. Reducing the interconnections not only reduces the delay but also reduces the interface between the partitions, making it easier for independent design and fabrication. A large number of interconnections increase the design area as well as complicate the task of the placement and routing algorithms.

$$Obj_1 = \sum_{i=1}^{k} \sum_{j=1}^{k} C_{ij} \ , \ (i \neq j) \quad (1)$$

$$Obj_2 = \sum_{i=1}^{k} |Obtained\ Area(V_i) - Desired\ Area(V_i)| \quad (2)$$

The complete objective function is defined as:

$$Obj = Min(w1 \times Obj1 + w2 * Obj2) \quad (3)$$

Where w1 and w2 are the weight factors and satisfy the relation: $w1 + w2 = 1$;

4. Proposed Method

In this paper there are four different forms of variation in SGA have been proposed. In the first modification form, there was the inclusion of extinct and diversification operators with SGA. In the second form associated with SGA, hetero-homo status group-based parent selection has been combined with tournament selection. In the third form, extinction and diversification operator has been included with IH2 SGATS [25]. In the fourth form, the extrinsic model of hetero-homo status group-based parent selection has been applied in association with extinction and diversification. The function block diagram of SGA has shown in Fig.1.

Initially, a population of a defined size has been generated through the uniform random process. Each member carries a random permutation of integer number from 1 to mm (where mm is the total number of modules in the considered circuit); the fitness of members in the population has been obtained. To obtain the offspring first, the two parents have to be selected from the current population. The selection of two parents has been defined through the fitness-based tournament selection. In this process, n members from the population have considered randomly, and among n members, the fittest member has considered as a parent. The advantage of this process is that the is a fair chance of all those members having relative fitness higher than the weakest (n-1) members. Through two-point crossover, offspring have been created, and the mutation operator provides a random change from one module to another one. The process of crossover and mutation may make the offspring infeasible. The transformation from an infeasible to a feasible solution has been done by recorrect operator. The working principle of the recorrector operator has shown in the section. The current

parent and offspring population form a combined pool from where score based tournament selection process is applied to define the next generation population. In the score-based tournament, each member faces a number of random opponents, and depending upon their fitness, the score of the tournament decides. The final score of a member is the total obtained score against their opponents. Members with higher value of tournament scores are considered members of the possible next-generation population. Over this population, the extinction and diversification process has applied. The details of the extinction and diversification process have been discussed in the section. The diversification process's outcome has been considered the final next-generation population. Depending on the terminating criteria, either this next generation will become the current population or the fittest member has been considered the final solution.

5. Status-Based Sub-Group Formation

From the parent population, for each member, the fitness value is estimated. The relative fitness, which represents the individual fitness w.r.t others in the population, has been estimated by Eq.4

$$R_{fit(i)} = 1 - \frac{O_{max} - O_i}{O_{max} - O_{min}} \quad (4)$$

Where O_{max} and O_{min} are the maximum and minimum values of the objective function. The higher value of relative fitness has a better quality of solution compared to lower values. Depending upon the values of relative fitness, there are four divisions made elite ($0.75 < Rf < 1$), good ($0.5 < Rf \leq 0.75$), average ($0.25 < Rf \leq 0.5$) and poor group ($0.0 < Rf \leq 0.25$). The individual member has assigned to a particular group depending upon their relative fitness.

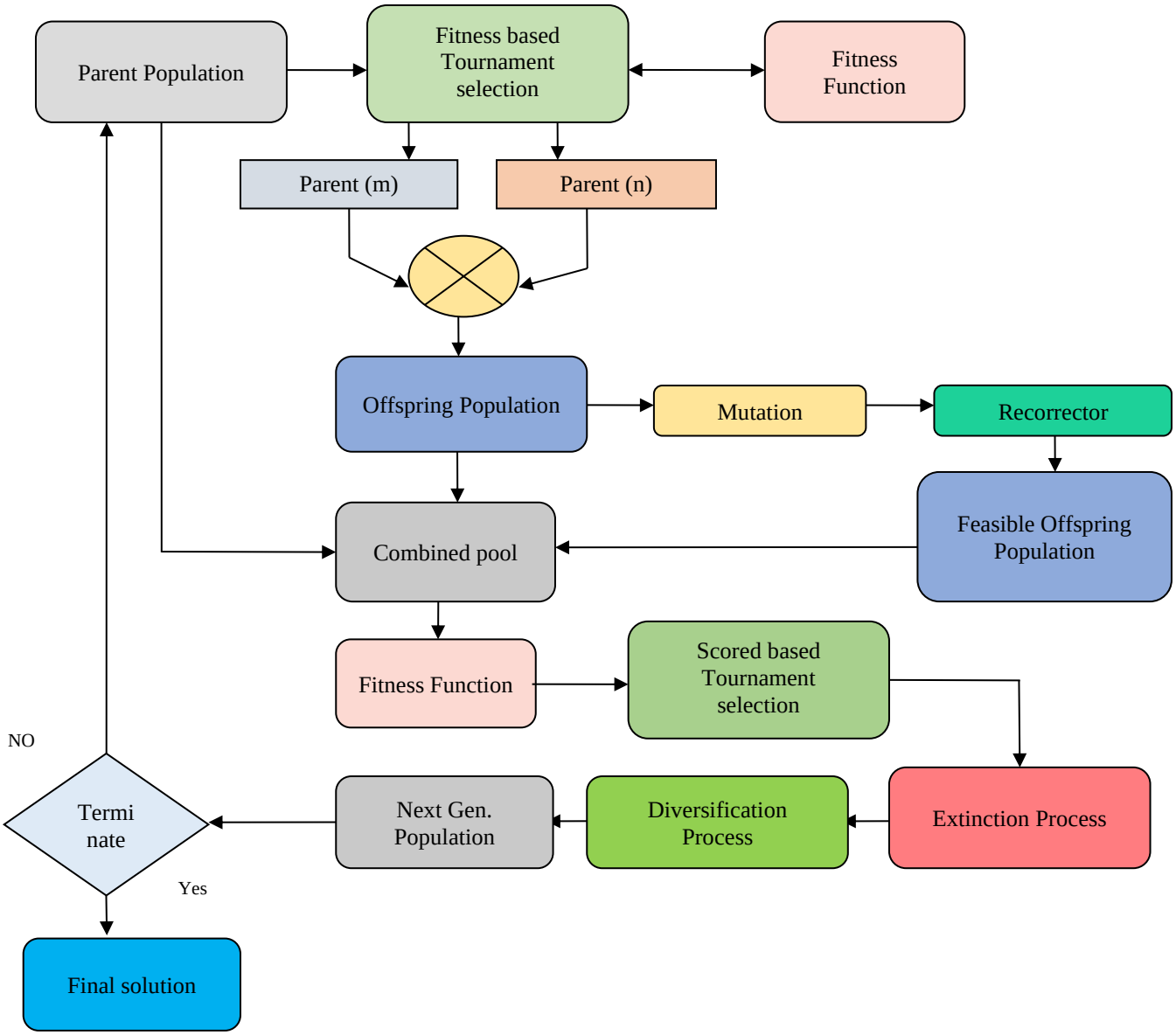

Fig. 1 Function block diagram of SGA.[25]

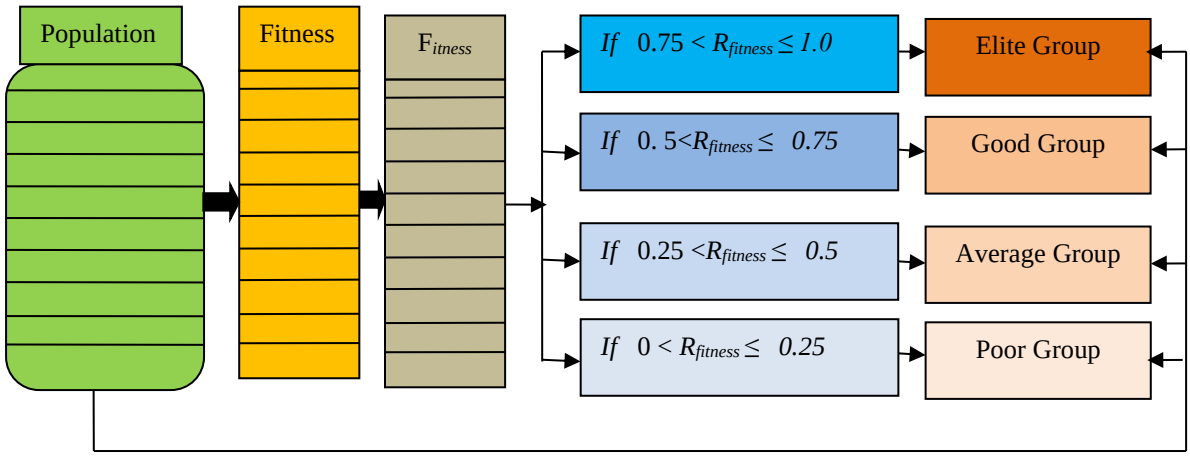

Fig. 2 Status-based sub-group formation

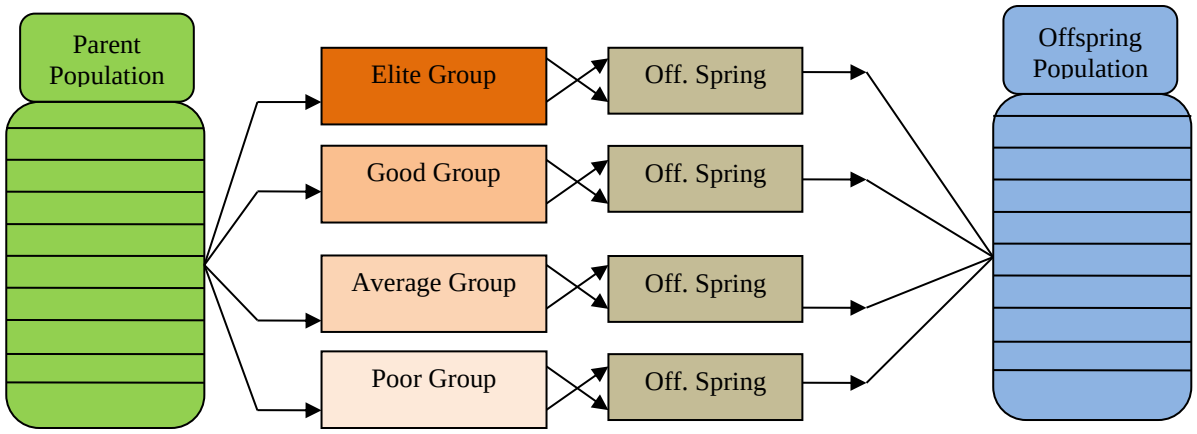

Fig. 3 Homogeneous status-based offspring creation

The relative fitness estimation is important because it provides the relative status of individual members at any time, and the definition of different groups is possible; otherwise, using fitness value, it is very difficult to differentiate the different groups, particularly when convergence is near. The process has shown in Fig.2.

5.1. Homogeneous Status-Based Offspring Creation

The creation of offspring in the homogeneous environment has shown in Fig.3. To create the offspring, and parents were selected randomly from the same sub-group. In this process, both parents come from the same fitness class. The two members of parents were selected through the uniform distribution process.

This process does follow the natural instinct of selecting the mating partner which carries a similar fitness class. Each group creates the same number of offspring as the size of the group population.

5.2. Heterogeneous Status-Based Offspring Creation

From the parent population, based on fitness, the formation of different groups takes place. The selection of two parents for the creation of two offspring has been provided through the selection of the two members from two different groups. Two different groups have significant differences in their relative fitness, which form a heterogeneous environment. The selection of any two groups takes place under uniform random distribution. The selected two parents produce the offspring through the defined process of creation. This heterogeneous process ensures the corresponding events of a natural system where rather than selecting the mating partner from the same class of fitness, the external class of fitness got the preferences like in the case of humans where members belong to poor and rich backgrounds, different cast, a different religion, etc. This heterogeneous environment provides the opportunity to explore the better solution and helps maintain diversity. The process flow has shown in Fig.4.

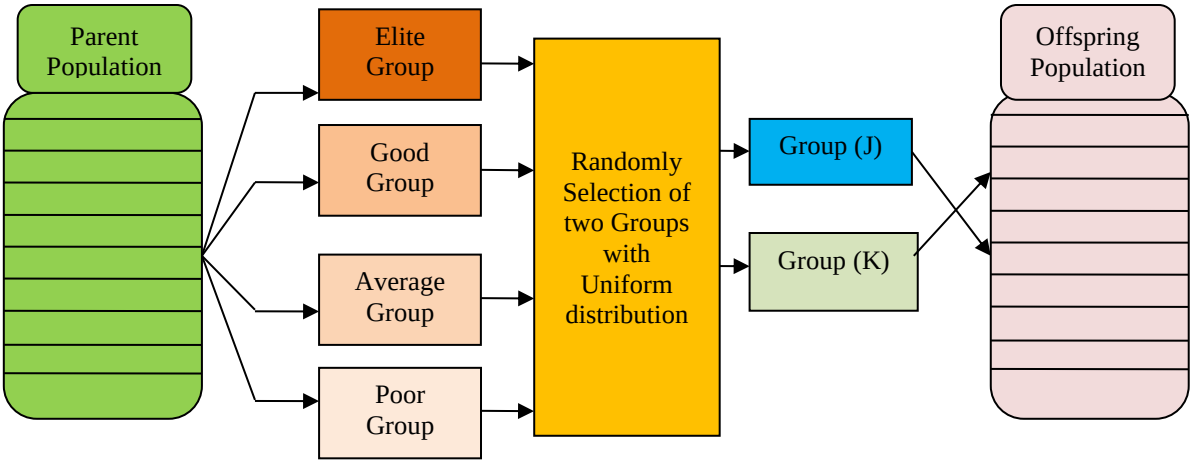

Fig. 4 Heterogeneous status group-based offspring generation

5.3. Extrinsic Hetero-Homo Status-Based SGA including Extinction (EH^{2}E-SGATS)

This model is based on the behaviour of patterns by individuals in the big social culture in the selection of their mating partner. The selection of their mating partner does not follow the tournament selection, which has the principle of random opponent selection. In most successful species like humans, rather than tournament selection, the status-based selection of mating partner has given the preference. It has been observed that there is a good chance that the partner has been selected from the different status groups. The working flow of EH2E-SGATS has shown in Fig.5. Initially, based on the fitness values, a status-based different sub-group has formed. Two parents were selected through the uniform random process under the homogeneous and heterogeneous status groups. The selection of two parents have considered in each process, and later offspring were produced through the E-SGATS process. Once the size of the offsprings population size is equal to the current population size, they combine with the parent population to decide the next-generation population, which has to pass through the extinction and diversification process to define the final next-generation population. Based on the termination criteria, either the next iteration will start, or the process will be terminated. The whole working process flow has shown in Fig.5.

6. Experimental Results

To provide the experimental confirmation of the proposed model, different varieties of combinational circuits have been considered. Different complexity was involved from partitioning the circuit from 2 to 5 levels containing the equal distribution of each partitioned area of total area and custom-defined partitioned area for each partition. Over each circuit, the 6 different algorithms DYPSO), SGATS, E-SGATS, IH2-SGATS, IH2E-SGATS and EH2E-SGATS have been applied for 10 independent trials; the performance of each algorithm has been estimated in terms of the mean value of satisfying percentage desired area of partitions, the need of a total number of interconnections among all available partitioning and objective function value. The standard deviation of objective function values has also been estimated to obtain the algorithm's robustness. Along with numeric comparison among the different algorithms, the convergence graphs have also been analysed to get an idea about the speed of finding the solution. Circuit partitioning has also been presented for the best-achieved partition under different trials, assigning the same colour for all modules that come under the same partition. The whole process has been simulated in a MATLAB environment. The circuit is represented as a graph where nodes represent the circuit module, and the connection between nodes is defined according to connections among the circuit modules.

The population for each algorithm has been maintained as 50, and the allowed number of iterations under a trial is 100. For DYPSO, inertia weight has reduced from 1.2 to 0.1, while cognition and constant social values have been kept at 0.5. The extrinsic factor of 0.72 also has been applied. In all forms of SGATS, the size of the tournament has been kept at 10% of the population size. Two-point crossover operator has been applied, and mutation probability has been maintained as 0.1. In the process of extinction, there was a random selection of numbers through uniform distribution in the range of [0 Mx], which will be extinct at the current iteration. The maximum value limit is 40% of the population size.

Circuit information:
Number of modules =21;
Module Area: [2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,1, 2, 2,1, 2, 2, 2, 2];
Required no of partition: 4;
Allocated area per partition: an equal area for each partition = 39/4 = 9.75;

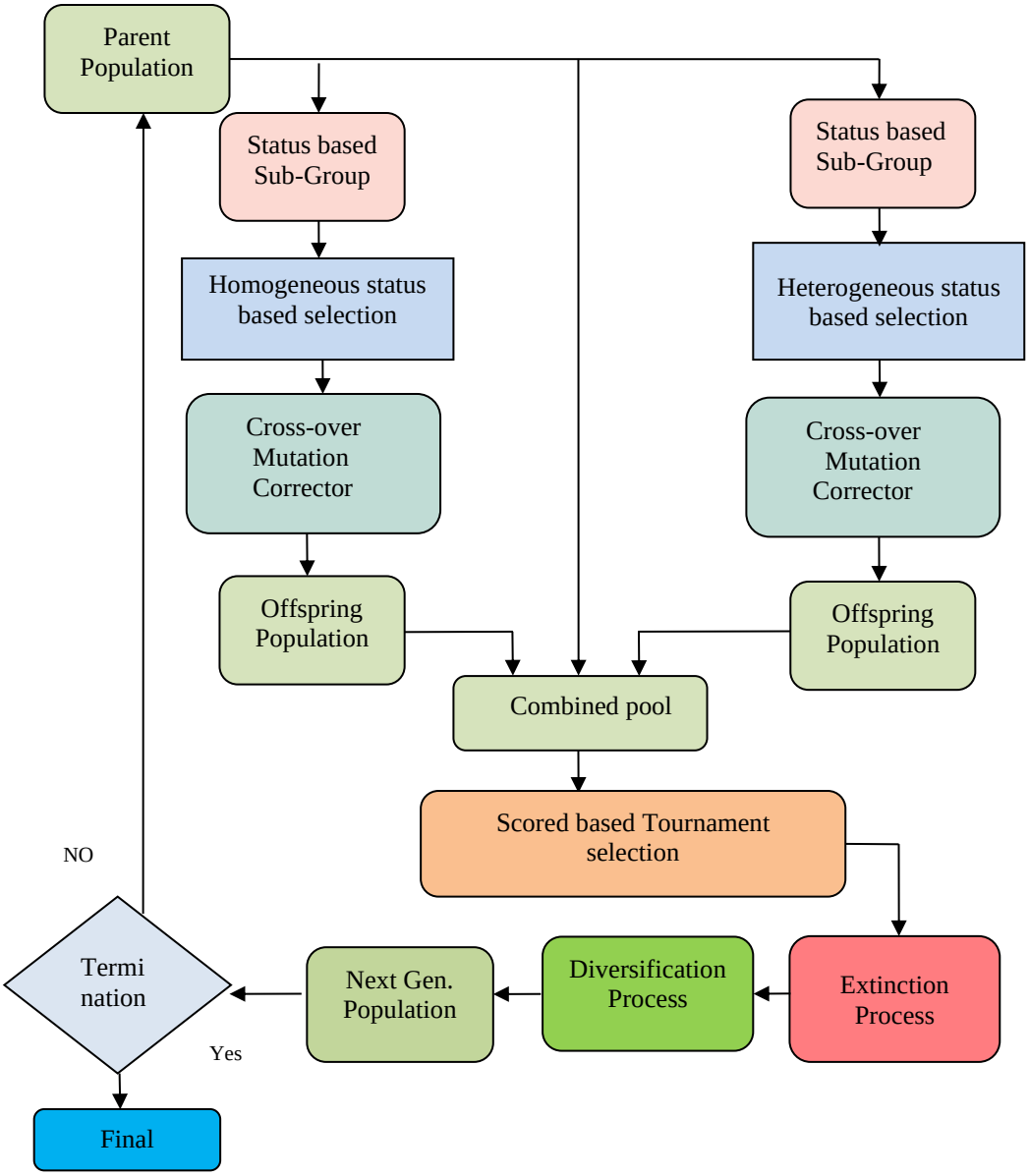

Fig. 5 Working process flow for EH^{2}E-SGATS

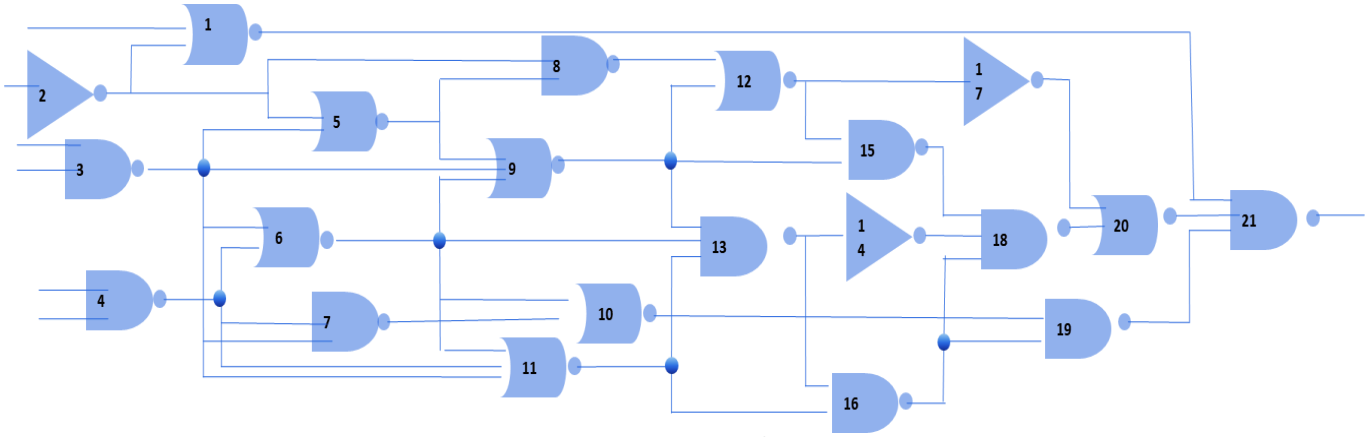

Fig. 6 Module connection for CKT1

Table 1. Mean Performances obtained by the different algorithms over the partitioning of CKT1

CKT 1	Partitions Area constraint satisfaction (%)	Total no. of connections among partitions	Objective Function value
DYPSO	0.9410	33.1000	35.4000 (1.7920)
SGATS	0.9513	26.5000	28.4000 (1.2867)
E-SGATS	0.9462	27.1000	29.2000 (1.2517)
I-SGATS	0.9564	26.5000	28.2000 (0.9487)
IH^{2}E-SGATS	0.9615	26.9000	28.4000 (1.1972)
EH^{2}E-SGATS	0.9564	26.3000	28.0000 (0.8498)

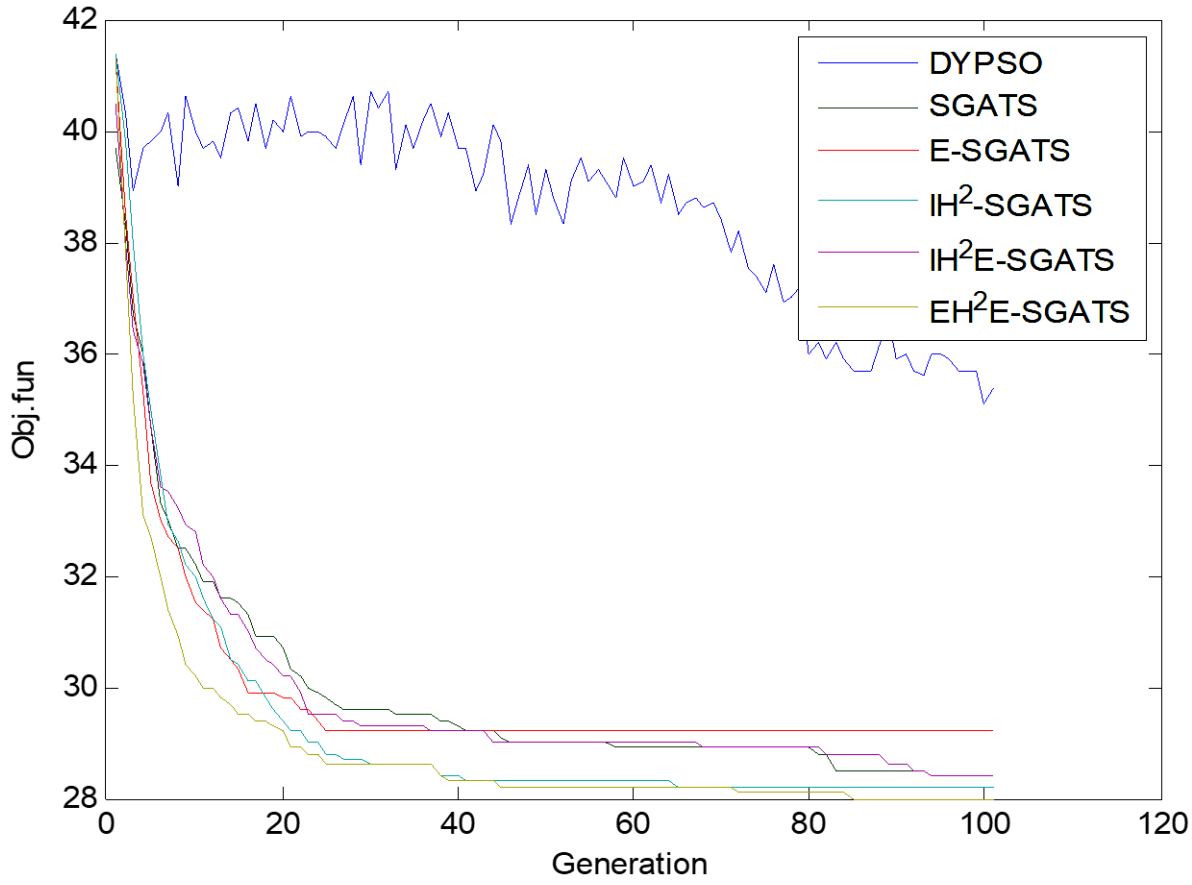

Fig. 7 CKT1 partitioning mean convergence by different algorithms

Table 2. CKT1 partition information

Partition	Partition Area	Total Area violation	No.of Connection between Partitions
P1{1 2 5 8 17 20}	10	1.5000	26.0000
P2{3 4 6 7 11}	10		
P3 {9 10 12 13 15}	10		
P4{14 16 18 19 21}	9		

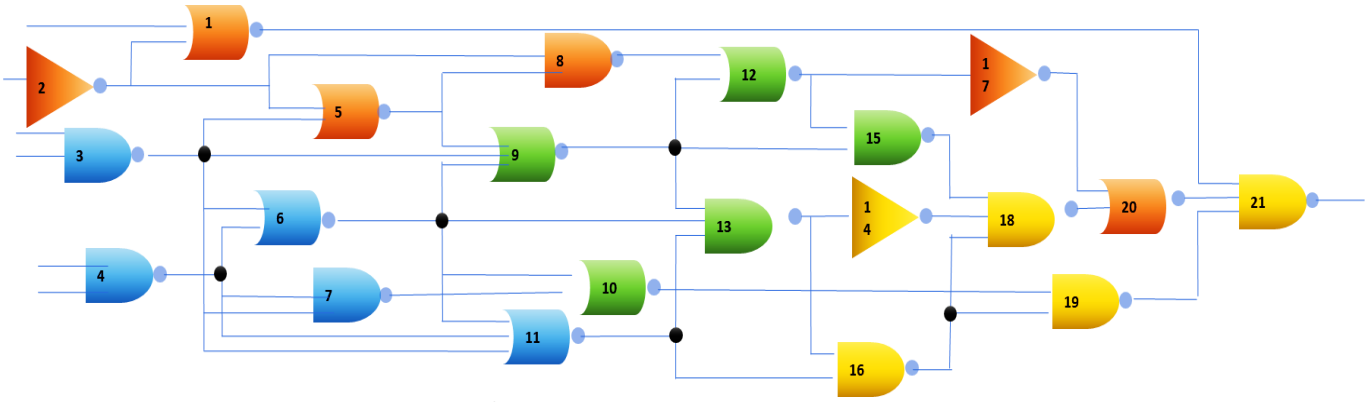

Fig. 8 Four-level partitioning of CKT1. [all modules having the same color belongs to the same partition]

The partition circuit for CKT1 has shown in Fig.6. The observed mean objective function value, along with area satisfaction and required interconnections, is shown in Table 2. It can be observed that the performance of DYPSO was inferior, while EH2E-SGATS delivered the minimum value of the objective function and the minimum amount of standard deviation. The convergence characteristics for all algorithm has shown in Fig.7, and there is fastest and most optimal convergence has been seen with EH^{2}E-SGATS. The obtained best partitions result among 10 trials is shown in Table.2, and Fig.8 shows the partitioned circuit where the same colour of module defines the module's belonging in the same partition.

7. Conclusion

The problem of solving the VLSI circuit partitioning using the heuristic approach having the background of the genetic algorithm has been presented in this work. The complexity of achieving the minimum number of interconnections and satisfying the area constraint has been met simultaneously by defining the problem as multi-objective functions. The proposed form of parent selection for offspring creation which has the basis of status-based hetero and homo inference has shown remarkable benefits compared to the conventional form of selection. The proposed form of selection has also shown that even the parents carried the fitness of poor class, there were fitter offspring generated compared to the higher fitness of the parent. Such possibilities were not possible with conventional tournament selection. Including extinct processes has made solution convergence faster and helped explore the high fitness of solutions. The performance point of DYPSO has shown the issue of slow or very low-quality solutions, while SGATS has performance was acceptable. The inclusion of the extinct operator always has shown betterment, while hetero-homo status-based group model associated with extinction outperformed all others.

References

[1] Y. Yodtean, and P. Chantngarm, "Hybrid Algorithm for Bisection Circuit Partitioning," *Proceedings of IEEE Region 10 Conference*, vol.4, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[2] G.-F. Nan, and M.-Q. Li, "Two Novel Encoding Strategies Based Genetic Algorithms for Circuit Partitioning," *Proceedings of the 3rd IEEE International Conference on Machine Learning*, vol. 4, pp. 2182–2188, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[3] G. C. Sipakoulis, I. Karafyllidis, and A. Thanailkis, "Genetic Partition and Placement for VLSI Circuits," *Proceedings of the 6th IEEE Conference on Electronics Circuits and Systems*, vol. 3, pp. 1647–1650, 1999. [CrossRef] [Google Scholar] [Publisher Link]

[4] C. J. Augeri, and H. H. Ali, "New Graph-Based Algorithms for Partitioning VLSI Circuits," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. V-521–V-524, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[5] A. Cincotti, V. Cuttelo, and M. Pavone, "Graph Partitioning Genetic Algorithms With OPDX," *Proceedings of IEEE World Congress on Computational Intelligence*, pp. 402–406, 2002. [CrossRef] [Publisher Link]

[6] I. Hameem Shanavas, and R. K. Gnanamurthy, "Evolutionary Algorithmical Approach on VLSI Floor Planning Problem," *International Journal of Computer Theory and Engineering*, vol.1, no. 4, pp. 461–464, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[7] N. Krasnogor, and J. Smith, "A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues," *IEEE Transactionson Evolutionary Computation*, vol. 9, no. 5, pp. 474–488, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[8] Jianhua Li, and L. Behjat, "A Connectivity Based Clustering Algorithm with Application to VLSI Circuit Partitioning," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 5, pp. 384 - 388, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[9] N. Selvakkumaran, and G. Karypis, "Multiobjective Hypergraph-Partitioning Algorithms for Cut and Maximum Subdomain-Degree Minimization ," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 504 – 517, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[10] Shao-Jun Peng, Guo-Long Chen, and Wen-Zhong Guo," A Discrete PSO for Partitioning in VLSI Circuit," *2009 International Conference on Computational Intelligence and Software Engineering*, pp. 1- 4, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[11] I. Hameem Shanavas et al., "A Novel Approach to Find the Best Fit for VLSI Partitioning - Physical Design," *Advances in Recent Technologies in Communication and Computing (ARTCOM)*, pp. 330 – 332, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[12] Roman Bazylevych, and Lubov Bazylevych," The Methodology and Algorithms for Solving the Very Large-Scale Physical Design Automation Problems: Partitioning, Packaging, Placement and Routing," *2013 2nd Mediterranean Conference on Embedded Computing (MECO)*, pp. 1 - 2, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[13] Aarti Patel, and Prashant K.Shah, "Semi-Custom Design of Functional Unit Block Using Data Path Methodology in Data Cache Unit," *SSRG International Journal of VLSI & Signal Processing*, vol. 4, no. 2, pp. 34-38, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[14] Pradip Kumar Sharma, and Maninder Kaur, "A Discrete Firefly Algorithm for VLSI Circuit Partitioning," *Electronics and Communication Systems (ICECS)*, pp. 1-4, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[15] Bhargab Sinha et al., "Heuristics in Physical Design Partitioning: A Review," *International Conference on Innovations in Information, Embedded and Communication Systems*, pp. 1-5, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[16] Sadiq M. Sait et al., "Design Partitioning and Layer Assignment for 3D Integrated Circuits Using Tabu Search and Simulated Annealing," *Journal of Applied Research and Technology*, vol. 14, no. 1, pp. 67-76, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[17] Prabhakar R, Sreenivasa Murhthy K E, and Soundara Rajan K, "Incremental Detailed Placement for VLSI Design," *SSRG International Journal of VLSI & Signal Processing*, vol. 2, no. 3, pp. 7-12, 2015. [CrossRef] [Publisher Link]

[18] Linquan Lyu, and Takeshi Yoshimura, "A Force Directed Partitioning Algorithm for 3D Floorplanning," *2017 IEEE 12th International Conference on ASIC (ASICON)* [CrossRef] [Google Scholar] [Publisher Link]

[19] Robin Andre, Sebastian Schlag, and Christian Schulz, "Memetic Multilevel Hypergraph Partitioning," *Arxiv:1710.01968v2 [Cs.DS]*, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[20] Ghasem Pasandi, and Massoud Pedram, "A Graph Partitioning Algorithm with Application in Synthesizing Single Flux Quantum Logic Circuits," *Arxiv.Org > Cs > Arxiv:1810.00134*, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[21] Xia Lei et al., "A New Multilevel Circuit Partitioning Algorithm Based on the Improved KL Algorithm," *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (Bigdatasecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS)*. [CrossRef] [Google Scholar] [Publisher Link]

[22] P. Rajeswari et al., "An Investigation on Basic Concepts of Particle Swarm Optimization Algorithm for VLSI Circuits," *International Journal of Engineering and Technique*, vol. 4, no. 1, pp. 144-149, 2018. [Publisher Link]

[23] P. Rajeswari, Theodore S. Chandra, and Amith Kiran Kumar, "Synthesis of VLSI Structural Cell Partitioning Using Genetic Algorithm" *Springer Nature Proceedings*, vol. 1270, pp. 279-287, 2020. [CrossRef] [Publisher Link]

[24] P. Rajeswari, and Theodore S Chandra, "Partitioning of VLSI Circuits on the Basis of Standard Genetic Algorithm and Comparative Analysis of Partitioning Algorithms," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 9, no. 12, pp. 126-133, 2022. [CrossRef] [Publisher Link]

[25] Ajoy Kumar Khan et al., "A New Efficient Layer Assignment Algorithm for Partitioning in 3D VLSI Physical Design," *2013 1st International Conference on Emerging Trends and Applications in Computer Science*, pp. 203 - 207, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[26] P. Rajeswari, and S Theodore Chandra, "A Survey on An Optimal Solution for VLSI Circuit Partitioning in Physical Design Using DPSO & DFFA Algorithms," *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, pp. 868-872, 2017. [CrossRef] [Google Scholar] [Publisher Link]