International Journal of Mechanical and Production
Engineering Research and Development (IJMPERD)
ISSN (P): 2249-6890; ISSN (E): 2249-8001
Vol. 8, Issue 1, Feb 2018, 945-956
© TJPRC Pvt. Ltd.

A BROAD REVIEW ON VARIOUS VLSI CAD ALGORITHMS FOR CIRCUIT PARTITIONING PROBLEMS

R. MANIKANDAN, K. R. SEKAR & K. HARIHARAN

School of computing, Sastra Deemed University, India

ABSTRACT

Circuit partitioning is the first and the most important step in the designing of VLSI circuits. Owing to the rapidly increasing size of the designs, partitioning tools are becoming more important for the future. The partitioning algorithms are of two types, namely, constructive algorithms and iterative algorithms. In constructive algorithms, partition sets are formed with the help of algorithms; whereas, in case of iterative algorithms, new improved partition sets are formed at each iteration stepwith the modified netlist. A variety of heuristic algorithms have been developed to solve the problem of mincut which is NP-complete. With the main objective of minimizing the cutsize, numerous algorithms have been proposed for circuit partition which includes genetic and evolutionary algorithms, probability-based algorithms, clustering algorithms, and nature-based heuristics. The main intention of this paper is to provide a concise review of the VLSI CAD algorithms adopted for designing VLSI circuits. From the numerous partitioning methods available in the literature, a subjective selection has been made.

KEYWORDS: VLSI Circuit Partitioning,Probability Based Algorithm,Clustering Algorithm, Ant Colony Optimization, Firefly Approach & Ccomputer Aided Design

Received: Jan 03, 2018; **Accepted**: Jan 23, 2018; **Published**: Jan 31, 2018; **Paper Id.**: IJMPERDFEB2018115

INTRODUCTION

Partitioning refers to the process of breaking down a circuit into smaller sub-circuits and helps to speed up the design process. The sub-circuits are designed independently while keeping the functionality of the circuit intact. Partitioning is usually done to minimize the number of interconnections (Dutt & Deng, 2002); thereby, minimizing the complexities in the design process and also to reduce the space complexities. As circuit partitioning is the most critical step in the physical design of a VLSI circuit. The main objective of circuit partitioning is to minimize the number of cuts and to achieve the global optimum. The present paper reviews the various CAD algorithm approaches and the existing CAD techniques used for better circuit partitioning. The paper analyzes various heuristics applied to solve the partitioning problem. Some of them include genetic algorithm, ant colony optimization, simulated annealing and their variations.

Kernighan and Lin proposed a bipartitioning algorithm in 1970. This algorithm uses an iterative approach where the most suitable pair of components is swapped at every iteration; thereby, ensuring local optimization. A few years later, Fiduccia and Mattheyses (1982)designed a more complex iterative algorithm where the most suitable vertex of the partition is chosen for swapping and the process is repeated until no further vertex from other partition is available. Apart from these two common heuristics, a number of heuristics have been proposed to solve the problem of partitioning which is analyzed in this paper.

PARTITIONING TECHNIQUES

Multiway Circuit Partitioning Algorithm

Traditional two-way algorithms cannot be used to satisfy the requirements that optimize on various objective functions. As the requirements and problems vary, many objective functions are derived which cannot be solved by the two-way partitioning methods. Hence, the need for multi-way partitioning becomes evident. Cong, Labio and Shivakumar (1996) proposed a paradigm for multiway circuit partitioning based on dual net transformation. The dual transformation algorithm was applied to k-FM and the new algorithm k-dualFM was found to reduce the net cutsize by 20-31%. Furthermore, the same algorithm was applied to k-maximum fanout-free cone (MFFC)-FM algorithm and it was found that k-DualMFFC-FM reduced the net cutsize by 15 to 26%. The dual transformation algorithm was also compared with EIG1 and Paraboli wherein it was found to have reduced the net cutsize by 56%. The k-FM module partitioning algorithm has been converted into k-way net partitioning algorithm for solving the k-way module contention problem. Tan et al. (1997) presented a similar multi-way partitioning algorithm by including the potential gain function into the net cost function. This function removes the constraint placed on moving a single cell, to achieve a balanced partitioning. This new algorithm was found to be more remarkable than the existing K-dual algorithm. Alternatively, Dasdan and Aykanat (1997) proposed two new approaches for the multi-way partitioning algorithm as all theKL-based partitioning algorithms restricts the movement of cells in each pass. The first approach included more phases in each pass toallow one move for each cell. And the second approach was to include mobility for each cell.

However, Gong and Lim (1998) found that the traditional iterative multiway partitioning algorithms were unsuitable for large circuits. They proposed a hill climbing method known as pair wise cell movement method to overcome the limitations of K-FM methods. This method distributes the clusters evenly and minimize the number of connections across multiple cutlines.

Traditional FM-based iterative methods are based on local optima and the selection of next cell to be moved is based on the cell gain. In order to overcome these two problems and to explore a broader solution space, multiway hypergraph partitioning algorithm was proposed by Zhao, Tao, and Zhao(2002). In this algorithm, the next cell to be moved is selected based on both the cell gain and the net gains of all the cells incident to the cell. This algorithm performed better compared to the previous multi partitioning algorithm proposed by Dasdan and Aykanat (1997) in terms of cutsize and runtime.

Later in 2013, Sikand, Gill, Chandel and Chandel implemented the multi-way partitioning algorithm with some additional modifications and advocated the same for solving partitioning problems. The algorithm was subjected to iterations and clustering to deal with the problems of local minima. An important addition made to the existing multiway partitioning algorithm is the addition of pads to the output netlist files to make them as standalone files. Iterations and clustering were used to obtain multi-way partitioning from the two initial partitions. This algorithm additionally uses the top-down clustering technique and primal-dual technique to enhance the partitioning results. They can also handle more than one objective function which is not possible by the core FM algorithm.

Multilevel Circuit Partitioning

As the size and complexity of the problem increase, it becomes impossible for the clustering algorithm to provide excellent results in single application. Hence, multilevel partitioning algorithm was used by Alpert, Huang, and Kahng(1998). The algorithm first clusters the instances recursively until its size is smaller than the given threshold. The partitioning algorithm is then applied to uncluster the instances. It was found that the multilevel partitioning algorithm yielded better results when compared to the previous algorithms, with less CPU time. Karypis, Aggarwal, Kumar and Shekhar (1999) presented a hypergraph partitioning algorithm based on multilevel paradigm in order to produce high-quality partitioning in less amount of time. Partitioning introduces local and global interconnect delays which affects the performance of the partitioning algorithm. In order to overcome this, hierarchical performance driven multilevel partitioning was introduced by Cong, Lim, and Wu(2000). Their algorithm provided a competitive cutsize and delay minimization compared to the previous multi level algorithms.

Complex unstructured relationships are best explained by hypergraphs. It was found that though the multilevel partitioning techniques obtain subgraphs, they cannot satisfy the global objective function. In order to overcome this disadvantage, Karypis and Kumar (2000) presented a new partitioning algorithm known as the k-way multilevel hypergraph partitioning algorithm. Their algorithm was found to outperform the existing K-PM/LR algorithm in terms of speed and hyperedge cut. The partitioning objective function and partitioning balancing constraint are satisfied with the help of the greedy refinement algorithm. However, this algorithm was found to be trapped in local minima.

Selvakkumaran and Karypis (2006) designed a new multiobjective hypergraph partitioning algorithmthat not only emphasizes on minimizing the cut but also minimizes the maximum sub-domain degree. Although the previous multilevel algorithms succeeded in minimizing the number of cuts, these cuts were not uniformly distributed throughout the different subdomains. By implementing the present algorithm, the maximum subdomain degree was found to be lowered by 5-36% when compared to previous algorithms. The computational complexity of the algorithm was also found to be relatively low. Aykanat, Cambazoglu and Uçar (2008) found out that hypergraph partitioning algorithm performed better when they were implemented using the direct k-way algorithm rather than with the recursive bisection-based partitioning algorithm. They found out that multi-level k-way hypergraph partitioning algorithm performed better with multiple constraints and fixed vertices. Lotfifar and Johnson (2015) proposed a multilevel hypergraph partitioning algorithm that performs in a sequential manner. Using this algorithm unimportant hyperedges were removed to make better clustering decisions. The number of transistors needed for a VLSI design is increasing exponentially. The partitioning algorithms also have linear worst case complexity. In order to increase the performance of existing subgraph partition, Jain and Kamalapur (2016) introduced a new hypergraph partitioning algorithm to identify dense subgraphs while addressing the minimum cut cost, which the previous algorithms failed to address. Owing to multilevel recursive coarsening, the existing graph partitioning algorithms cannot guarantee minimum cut and follow global view; whereas, the proposed algorithm was found to yield better partitions with minimum cut cost.

Genetic and Evolutionary Algorithm

Baruch, Cret and Pusztai (1999) compared the genetic algorithm with simulated annealing partitioning approach and found that genetic algorithm took less execution time compared to that of simulated annealing while generating similar results. Also, in case of parallel processing where the data has to be distributed across the memory of the parallel machine, graph partitioning techniques do not prove to be effective where the number of cut edges is minimized. Hence, evolutionary algorithms were combined with multilevel heuristic algorithm by Soper, Walshaw and Cross (2004) to obtain high quality partitions. Deep, Singh, Singh and Singh (2009) realizing the potential of genetic algorithms and because the existing algorithms were not able to perform in terms of complexity, time and cost, used the genetic algorithms for circuit partitioning. It was also found that the existing conventional circuit partitioning algorithms were unable to solve NP hard problems. Hence, Gupta, Garg and Gupta (2012) used genetic algorithm to solve circuit partitioning.

Roy and Sarma (2012) adopted the genetic algorithm for circuit partitioning. The solution to circuit partitioning must be the global optimum. Their method involved finding the fitness value for each solution and discarding the solution which has low fitness value. Though the genetic algorithms require huge memory,they take comparatively less time to compute the number of cuts and sub-circuits. However, the genetic algorithms are not free from issues. Genetic algorithms do not deal directly with the optimization problem; rather they work with the codes generated to represent the problem. Therefore, coding the problem under study to represent the parameters for the algorithm is an important issue. As genetic algorithms deal with an initial set of solution, it is important to create the initial population with all possible solutions. Further, the quality of the initial population is unknown to further improve the solution. Hence, it is of prime importance to design an interface between the algorithm and the problem environment. Selection of genetic operators is another issue to be dealt with the use of genetic algorithm.

Eberhart and Shi (1998) compared the evolutionary paradigms, genetic algorithms and Particle Swarm Optimization algorithm (PSO). They compared the parameters used in the algorithms such as elitist strategy of genetic algorithm, use-better parameter of PSO, inertia weight, variance and assignment of velocity on a parameter-by-parameter basis. The performance of one algorithm can be improved by incorporating the features of the other algorithm. Areibi, Moussa and Abdullah (2001) compared the genetic algorithms with heuristic search techniques. According to their findings, iterative methods that are based on modules yielded better partition results. The genetic algorithm which has been modified to incorporate the local search is known as the Memetic algorithm. It was found that the solutions obtained by memetic algorithms were far better than the results obtained from genetic algorithms. Subbaraj, Sivasundari and Kumar (2007) used the memetic algorithm (combination of local search and global search) to achieve minimum cut as well as enhance the other important parameters of circuit partitioning such as power, delay and area.

Singh, Kalpna and Mishra (2013) performed hybrid implementation to achieve circuit minimization with the help of PSO algorithm and Genetic algorithm (GA). Both the algorithms were combined to obtain their individual features to solve the problem of circuit partitioning. Hence, it was found that hybrid PSO and GA is better than any other technique in achieving the effective circuit partitioning. Similarly, Shanavas and Gnanamurthy (2014) used the hybrid evolutionary algorithm for finding an optimal solution for circuit partitioning, floor planning, routing and placement. Their hybrid evolutionary algorithm combined simulated annealing with the genetic algorithm to achieve circuit minimization. Sangwan, Verma and Kumar (2014) developed a circuit partitioning approach with the help of evolutionary techniques. Genetic algorithms help in balancing the partitions and distributing the connections. However, the run time of the algorithm increases quickly as the size of the problem increases due to a large number of initial populations. Hence, local search optimizationhas been used at various stages in each iteration step.

Probability Based Algorithm

Dutt and Deng (1996) adopted a min-cut two-way partitioning of the circuit to reduce the number of components and the interconnections in a large circuit and to facilitate parallel simulation of circuits. The circuit is partitioned into two subsets in such a way that the number of nets connecting nodes in different subsets is minimized. Their approach (PROP) which was based on probability and conditional probability theory was found to outperform various other iterative improvement methods and clustering based algorithms. Node probabilities and node gains are recomputed at each iteration step to obtain better performance than previous deterministic methods like FM (Fiduccia & Mattheyses, 1982)and LA (Look Ahead algorithm)(Krishnamurthy, 1984). However, it was found that PROP, when used in conjunction with the clustering based algorithms, yielded high quality results. The run times of PROP was found to be more favorable than other iterative and clustering algorithms.

In 2000, Dutt and Dengimprovised the PROP algorithm and presented another algorithm known as SHRINK-PROP along with PROP algorithm. Improvement iterative algorithms such as FM and LA are mainly concerned about the immediate impact of moving a node. Therefore, the present algorithms were modified to include probabilistic gains (PROP) which can capture the future implications of moving a node in the current time. This algorithm was further extended to increase the probability of removing recently perturbed nets (nets whose nodes have been removed for the first time). The extended algorithm was known as the SHRINK-PROP algorithm. SHRINK-PROP algorithm helps in increasing the number of nets removed from the cutset in any pass. These algorithms were designed to predict the likelihood of future events much better than the FM and LA algorithms. Further, the mincut improvement of PROP and SHRINK-PROP over FM was found to be 24% and 31% respectively.

Clustering Algorithm

Iterative partitioning algorithms such as FM and LA are move-based and are known as local improvement algorithms. Dutt and Deng (1997) proposed new iterative methods to move clusters that straddle the two subsets of a partition into one of the subsets. Their approach can explore a wider solution space and is less dependent on the initial partition. The method was experimented on ACM/SIGDA benchmark circuits and revealed up to 70% improvement over FM in cutsize; 25% per circuit improvement; and 35% overall improvement. Their method which was purely based on clustering was found to be significantly better both in terms of cutset quality and speed. Clustering based mechanisms were incorporated into the traditional FM based algorithms to enhance the performance of the iterative algorithms. The k-means algorithm are center-based and its performance depends on the initialization of the centers. Hence, Zhang, Hsu and Dayal (1999) proposed a new algorithm known as K-Harmonic Means algorithm (KHM). KHM are also center-based which makes use of the harmonic averages between each data point and the centre for its performance function.

Krishna and Murty (1999) introduced a new algorithm known as genetic k-means algorithm which is a hybrid genetic algorithm. The earlier genetic algorithms make use of expensive crossover operators to generate child chromosomes from the parent chromosomes. In order to overcome this expense, the genetic algorithm has been hybridized to include k-means clustering algorithm. The resultant algorithm was known as Genetic K-means Algorithm (GKA). It was observed that GKA performs at a faster rate than other evolutionary algorithms used for clustering. Similarly, Maulik and Bandyopadhyay (2000)reported that GA-clustering algorithm was found to be superior to the traditional k-means algorithm. The clustering algorithms that utilized the searching capability of genetic algorithms were found to be more efficient to optimize the clusters. Due to the problem of initialization suffered by the k-means algorithm, Kao, Zahara and Kao (2008) proposed a hybrid technique which combines k-means algorithm, Nelder-Mead simple search and PSO algorithm known as K-NM-PSO search, which effectively finds the global optima. Similarly, Yang, Sun and Zhang (2009) proposed another hybrid approach that combines the K Harmonic Means algorithm with the PSO and is known as PSOKHM. This hybrid algorithm overcomes the problem of local optima and the slow convergence of PSO algorithm. Niknam, Amiri, Olamaei and Arefi (2009) proposed another hybrid algorithm by combining PSO and SA (PSO-SA) for clustering to overcome the local minima and initial cluster center problem of k-means algorithms. It was found that PSO-SA converges more quickly than k-means, PSO and SA algorithms. Wang and Cai (2009) proposed a new hybrid algorithm known as hybrid multi-swarm particle swarm optimization problem (HMPSO) to solve constrained optimization problems. The algorithm was found to be efficient to solve 24 benchmark test functions that were collected during the 2006 IEEE Congress on Evolutionary Computation. Peng, Chen and Guo (2010) proposed a new algorithm known as multi-objective discrete PSO algorithm (DPSO) to solve the problem of circuit partitioning. It was observed that the proposed algorithm yielded better results than the genetic algorithm and Tabu search algorithms for bipartition problem.

Most of the clustering algorithms are trapped in local optimum and are sensitive to outliers and initialization. In order to overcome these two issues, Zadegan, Mirzaie and Sadoughi (2013) proposed a ranked k-medoids algorithm where the initialization does not lead the algorithm to reach local optimum. Further, k-medoids algorithm was found to be the most suitable clustering algorithm for large data sets. Manikandan and Leela (2014) used clustering algorithms to provide effective circuit partitioningand are also time-saving. The clustering algorithms adopted by them include K-Mean, Y-Mean and K-Medoid. Among the means used for clustering, Fuzzy-c mean was found to give accurate results when compared to k-mean and y-mean algorithms but have increased run time. However, the Y-Mean algorithm was found to have less run time due to trained sets.

Simulated Annealing

Parallel computers are increasingly being used for the design of VLSI applications to speed up the process of designing. Hence, there arises the need for developing parallel algorithms for circuit testing, synthesis, logic minimization, and simulation. Gil, Ortega, Montoya,and Baños (2002) proposed a hybrid algorithm where the Tabu local search was incorporated into the simulated annealing algorithm. The resulting algorithm was called as Mixed Simulated Annealing and Tabu Search algorithm (MSATS). This algorithm is used for testing the circuits after partitioning; hence, it has been placed in the parallel test pattern generator. It was found that MSATS outperforms the Simulated Annealing and Tabu Search algorithms. Hybrid algorithms were always found to be more effective than the individual algorithms when applied to the same cost function. However, in case of Tabu search, moves or iterations that do not produce any improvements are also included in the search which may lead to cycling problems (visiting the same node repeatedly). In order to avoid the cycling issues, Tabu list has been used to explore the search space and to avoid the previously visited nodes. Effective Tabu list is maintained in the form of a first-in-first-out circular array. Kolar, Puksec and Branica (2004) used the simulated annealing procedure for the two-way partitioning of a circuit. Bhattacharya, Ghatak, Ghosh and Das (2014) also used thesimulated annealing approach for VLSI circuit partitioning. While designing a complex VLSI system, the cost of inter-module connectivity and the relevant circuital delay are the two important parameters to be considered. Simulated annealing approach is a probability based approach where a random entity from the solution space of the problem is selected as the initial state and the iterations are carried out. Guo, Liu, Chen and Peng (2014) proposed an effective hybrid algorithm which was based on Discrete Particle Swarm Optimization (DPSO) and a local search strategy called MDPSO-LS to solve the problem of partitioning by minimizing the cutsize and circuit delay. FM strategy was applied to improve the cutsize of each particle and the local search strategy was used to overcome the circuit delay. Furthermore, Markov chains were used to prove the global convergence of MDPSO-LS algorithm. MDPSO-LS algorithm was found to be highly feasible and efficient when compared with other partitioning algorithms.

Ant Colony Optimization

Dorigo, Maniezzo and Colorni (1996) defined a new paradigm known as Ant System whose main characteristics are positive feedback, distributed computation and the use of constructive greedy heuristic. Owing to the widespread usage of nanotechnology in fabrication, reliability analysis has become inevitable in designing combinational circuits. Wang, Gong and Kastner (2006) used the ant colony optimization technique for optimizing the latency, area, power, and other performance metrics of digital systems. ACO provided robust optimal solutions when compared with the simulated annealing approach. However, the authors suggested a hybrid approach which combines both ACO and SA to yield better results. Lee, Su, Chuang and Liu(2008) incorporated local search, ant colony optimization, to the genetic algorithm to enhance its performance and for multiple sequence alignment which is a NP-complete problem. ACO helps in overcoming the problem of local optima; therefore, the proposed algorithm (GA-ACO) was found to perform superior than all the existing algorithms. Jose, Nirmal Kumar, Hussain and Shanker (2014) devised a framework that used Ant Colony Optimisation (ACO) and primary input primary output fan in fan out partitioning algorithm to design fault tolerant VLSI systems. ACO is the proven meta-heuristic algorithm that provides an optimal solution for both single-objective and multi-objective problems. According to the authors, reliability and design for testability (DFT) issues are the two important concerns to be considered while designing a complexVLSI system. Therefore, they developed fault tolerant testable designs to help in the partitioning of VLSI circuits.

Firefly Approach

The firefly algorithm which is a novel meta-heuristic used for circuit partitioning was proposed by Xin-She Yang, inspired by the behavior of fireflies. Apostolopoulos and Vlachos (2010) used the nature-inspired heuristic, FireFly approach to solve the economic emissions load dispatch problem. Senthilnath, Omkar and Mani (2011) compared the FireFly algorithm with other two nature based heuristics, namely, Artificial Bee Colony (ABC) and PSO. Their results suggest that FireFly algorithm is an efficient, reliable and robust method to obtain optimized clusters. Sharma and Kaur (2014) used a FireFly based approach for partitioning the VLSI circuits. The discrete firefly algorithm is a swarm based heuristic algorithm that helps in solving the min cut circuit partitioning. The algorithm proposed in their study is known as Discrete FireFly Algorithm for Circuit Partitioning (DFACP). Using this algorithm, the initial set of array values (gates/nodes) are assigned to a single partition and the rest are assigned to another partition; thereby satisfying the balancing constraint. In case of the continuous optimization problem, distances between two nodes are calculated as Euclidean distance. In circuit partitioning, the distance between two fireflies is taken as the number of different arcs between them. Marichelvam, Prabaharan and Yang (2014) used the discrete FireFly algorithm to solve hybrid flowshop scheduling problems which are NP-hard. The algorithm was found to outperform many heuristic and metaheuristic algorithms.

CONCLUSIONS

This paper successfully presented some of the commonly adopted CAD Algorithms for Circuit Partitioning problems.. All the CAD algorithms were tested on benchmark circuits and were found to yield better results. The paper described the efficient approaches of partitioning with the help of genetic algorithms, evolutionary algorithms, hybrid algorithms, clustering algorithms and probabilistic algorithms. While genetic algorithms help in multi way partitioning of many types of VLSI circuits, hybrid algorithms which utilize PSO and GA were found to be better than any other optimization technique for minimizing the number of cuts in the partition. Probabilistic algorithms were found to outperform any other iterative improvement methods and clustering methods with favorable run times. In general, partitioning algorithms were used for large dataset and hierarchical clustering algorithms were used for small datasets. The present paper has thus presented the reviews on various techniques followed for circuit partitioning. The techniques applied to solve different NP-hard and NP-complete problems were also presented.

REFERENCES

1. *Alpert, C. J., Huang, J. H., & Kahng, A. B. (1998). Multilevel circuit partitioning. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 17(8), 655-667.*
2. *Apostolopoulos, T., & Vlachos, A. (2010). Application of the firefly algorithm for solving the economic emissions load dispatch problem. International Journal of Combinatorics, 2011.*
3. *Areibi, S., Moussa, M., & Abdullah, H. (2001). A comparison of genetic/memetic algorithms and other heuristic search techniques. In International Conference On Artificial Intelligence (IC-AI).*
4. *Aykanat, C., Cambazoglu, B. B., & Uçar, B. (2008). Multi-level direct k-way hypergraph partitioning with multiple constraints and fixed vertices. Journal of Parallel and Distributed Computing, 68(5), 609-625.*
5. *Baruch, Z. O. L. T. A. N., Cret, O., & Pusztai, K. A. L. M. A. N. (1999). Genetic algorithm for circuit partitioning. In Fifth Int. Conf. on Engineering of Modern Electric Systems: Section Computer Science and Control Systems (pp. 19-23).*
6. *Bhattacharya, A., Ghatak, S., Ghosh, S., & Das, R. (2014). Simulated annealing approach onto VLSI circuit partitioning.*
7. *Cong, J., Labio, W. J., & Shivakumar, N. (1996). Multiway VLSI circuit partitioning based on dual net representation. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 15(4), 396-409.*
8. *Ujwala A. Kshirsagar (Belorkar) & Ashish E. Bhande, VLSI Implementation of Back Propagated Neural Network for Signal Processing, International Journal of Electrical and Electronics Engineering Research (IJEEER), Volume 4, Issue 2, March - April 2014, pp. 131-140*
9. *Cong, J., Lim, S. K., & Wu, C. (2000, June). Performance driven multi-level and multiway partitioning with retiming. In Proceedings of the 37th Annual Design Automation Conference (pp. 274-279). ACM.*
10. *Dasdan, A., & Aykanat, C. (1997). Two novel multiway circuit partitioning algorithms using relaxed locking. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 16(2), 169-178.*
11. *Deep, A., Singh, B., Singh, A., & Singh, J. (2009). A Simple Efficient Circuit Partitioning by Genetic Algorithm.*
12. *Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26(1), 29-41.*
13. *Dutt, S., & Deng, W. (1996, June). A probability-based approach to VLSI circuit partitioning. In Proceedings of the 33rd annual Design Automation Conference (pp. 100-105). ACM.*
14. *Dutt, S., & Deng, W. (1997, January). VLSI circuit partitioning by cluster-removal using iterative improvement techniques. In Proceedings of the 1996 IEEE/ACM international conference on Computer-aided design (pp. 194-200). IEEE Computer*

Society.

15. *Dutt, S., & Deng, W. (2000). Probability-based approaches to VLSI circuit partitioning. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 19(5), 534-549.*

16. *Dutt, S., & Deng, W. (2002). Cluster-aware iterative improvement techniques for partitioning large VLSI circuits. ACM Transactions on Design Automation of Electronic Systems (TODAES), 7(1), 91-121.*

17. *Eberhart, R. C., & Shi, Y. (1998, March). Comparison between genetic algorithms and particle swarm optimization. In International conference on evolutionary programming (pp. 611-616). Springer, Berlin, Heidelberg.*

18. *Gil, C., Ortega, J., Montoya, M. G., & Baños, R. (2002). A mixed heuristic for circuit partitioning. Computational Optimization and Applications, 23(3), 321-340.*

19. *Gong, J., & Lim, S. K. (1998, November). Multiway partitioning with pairwise movement. In Computer-aided Design, 1998. ICCAD 98. Digest of Technical Papers. 1998 IEEE/ACM International Conference On (pp. 512-516). IEEE.*

20. *Guo, W., Liu, G., Chen, G., & Peng, S. (2014). A hybrid multi-objective PSO algorithm with local search strategy for VLSI partitioning. Frontiers of Computer Science, 8(2), 203-216.*

21. *Gupta, N., Garg, D., & Gupta, S. (2012). Genetic algorithms based partitioning of vlsi circuit Systems. In IJCA Proceedings on National Conference on Future Aspects of Artificial Intelligence in Industrial Automation 2012 NCFAAIIA (Vol. 2, pp. 15-19).*

22. *Jain, C., & Kamalapur, S. (2016). Hypergraph Partitioning Algorithm.*

23. *Jose, D., Nirmal Kumar, P., Hussain, A., & Shanker, P. (2014). VLSI Circuit Partitioning Using Ant Colony Optimisation to Yield Fault Tolerant Testable Systems. Arabian Journal for Science & Engineering (Springer Science & Business Media BV), 39(12).*

24. *Kao, Y. T., Zahara, E., & Kao, I. W. (2008). A hybridized approach to data clustering. Expert Systems with Applications, 34(3), 1754-1762.*

25. *Karypis, G., & Kumar, V. (2000). Multilevel k-way hypergraph partitioning. VLSI design, 11(3), 285-300.*

26. *Karypis, G., Aggarwal, R., Kumar, V., & Shekhar, S. (1999). Multilevel hypergraph partitioning: applications in VLSI domain. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 7(1), 69-79.*

27. *Kolar, D., Puksec, J. D., & Branica, I. (2004, May). VLSI circuit partition using simulated annealing algorithm. In Electrotechnical Conference, 2004. MELECON 2004. Proceedings of the 12th IEEE Mediterranean (Vol. 1, pp. 205-208). IEEE.*

28. *Krishna, K., & Murty, M. N. (1999). Genetic K-means algorithm. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 29(3), 433-439.*

29. *Krishnamurthy, B. (1984). An improved min-cut algonthm for partitioning vlsi networks. IEEE Transactions on Computers, (5), 438-446.*

30. *Lee, Z. J., Su, S. F., Chuang, C. C., & Liu, K. H. (2008). Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment. Applied Soft Computing, 8(1), 55-78.*

31. *Lotfifar, F., & Johnson, M. (2015, August). A multi–level hypergraph partitioning algorithm using rough set clustering. In European Conference on Parallel Processing (pp. 159-170). Springer, Berlin, Heidelberg.*

32. Manikandan, R., & Leela, V. (2014). Effective Clustering Algorithms for VLSI Circuit Partitioning Problems. Contemporary Engineering Sciences, 7(19), 923-929.

33. Marichelvam, M. K., Prabaharan, T., & Yang, X. S. (2014). A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems. IEEE transactions on evolutionary computation, 18(2), 301-305.

34. Maulik, U., & Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. Pattern recognition, 33(9), 1455-1465.

35. Niknam, T., Amiri, B., Olamaei, J., & Arefi, A. (2009). An efficient hybrid evolutionary optimization algorithm based on PSO and SA for clustering. Journal of Zhejiang University-SCIENCE A, 10(4), 512-519.

36. Peng, S. J., Chen, G. L., & Guo, W. Z. (2010). A multi-objective algorithm based on discrete PSO for VLSI partitioning problem. In Quantitative Logic and Soft Computing 2010 (pp. 651-660). Springer, Berlin, Heidelberg.

37. Roy, S., & Sarma, S. S. (2012). Improvement Of The Quality Of Vlsi Circuit Partitioning Problem Using Genetic Algorithm‖. Journal of Global Research in Computer Science, 3(12).

38. Sangwan, D., Verma, S., & Kumar, R. (2014, November). An Efficient Approach to VLSI Circuit Partitioning Using Evolutionary Algorithms. In Computational Intelligence and Communication Networks (CICN), 2014 International Conference on (pp. 925-929). IEEE.

39. Selvakkumaran, N., & Karypis, G. (2006). Multiobjective hypergraph-partitioning algorithms for cut and maximum subdomain-degree minimization. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 25(3), 504-517.

40. Senthilnath, J., Omkar, S. N., & Mani, V. (2011). Clustering using firefly algorithm: performance study. Swarm and Evolutionary Computation, 1(3), 164-171.

41. Shanavas, I. H., & Gnanamurthy, R. K. (2014). Optimal solution for VLSI physical design automation using hybrid genetic algorithm. Mathematical Problems in Engineering, 2014.

42. Sharma, P. K., & Kaur, M. (2014, February). A discrete firefly algorithm for VLSI circuit partitioning. In Electronics and Communication Systems (ICECS), 2014 International Conference on (pp. 1-4). IEEE.

43. Sikand, K. S., Gill, S. S., Chandel, R., & Chandel, A. (2013). Implementation of Multi-Way Partitioning Algorithm. International Journal of Computer and Communication Engineering, 2(1), 28.

44. Singh, R., Kalpna, K., & Mishra, D. K. (2013). Hybrid Optimization Technique for Circuit Partitioning Using PSO and Genetic Algorithm. International Journal of Emerging Trends in Electrical and Electronics (IJETEE–ISSN: 2320-9569), 4(2), 69-71.

45. Soper, A. J., Walshaw, C., & Cross, M. (2004). A combined evolutionary search and multilevel optimisation approach to graph-partitioning. Journal of Global Optimization, 29(2), 225-241.

46. Subbaraj, P., Sivasundari, K., & Kumar, P. S. (2007). An effective memetic algorithm for VLSI partitioning problem.

47. Tan, X., Tong, J., Tan, P., Park, N., & Lombardi, F. (1997, October). An efficient multi-way algorithm for balanced partitioning of VLSI circuits. In Computer Design: VLSI in Computers and Processors, 1997. ICCD'97. Proceedings., 1997 IEEE International Conference on (pp. 608-613). IEEE.

48. Wang, G., Gong, W., & Kastner, R. (2006). Application partitioning on programmable platforms using the ant colony optimization. Journal of Embedded Computing, 2(1), 119-136.

49. Wang, Y., & Cai, Z. (2009). A hybrid multi-swarm particle swarm optimization to solve constrained optimization problems. Frontiers of Computer Science in China, 3(1), 38-52.

50. Yang, F., Sun, T., & Zhang, C. (2009). An efficient hybrid data clustering method based on K-harmonic means and Particle Swarm Optimization. Expert Systems with Applications, 36(6), 9847-9852.

51. Zadegan, S. M. R., Mirzaie, M., & Sadoughi, F. (2013). Ranked k-medoids: A fast and accurate rank-based partitioning algorithm for clustering large datasets. Knowledge-Based Systems, 39, 133-143.

52. Zhang, B., Hsu, M., & Dayal, U. (1999). K-harmonic means-a data clustering algorithm. Hewlett-Packard Labs Technical Report HPL-1999-124.

53. Zhao, Z., Tao, L., & Zhao, Y. (2002). An effective algorithm for multiway hypergraph partitioning. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 49(8), 1079-1092.