International Journal on Emerging Technologies **10**(2): 67-70(2019)

ISSN No. (Print) : 0975-8364
ISSN No. (Online) : 2249-3255

A Study on Specific Computational Algorithms for VLSI Cell Partitioning Problems

R. Manikandan[1], R. Parameshwaran[2], J. Prassanna[3] and K.R. Sekar[1]
[1]School of Computing, SASTRA Deemed University, India
[2]Research Scholar, Department of ECE, National Institute of Technology-Tiruchirappalli, India
[3]School of Computing sciences and engineering,VIT University, Chennai Campus, Tamil Nadu, India

(Corresponding author: R. Manikandan)
(Received 13 April 2019, Revised 05 July 2019 Accepted 11 July 2019)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: VLSI Cell partitioning problem is the one of the important discussion in VLSI design. It consists of numerous stages in VLSI design, like-wise logic design and physical design. VLSI Cell partitioning problem is a well known NP hard problem. The Genetic Algorithm has been implemented to solve several complex computationally intensive problems (NP hard problems) because earlier conventional methods are not capable to execute the required breakthrough in terms of complexity, time and cost. A netlist is a description of the connectivity of an electronic circuit. Partitioning the netlist with minimum cut size and swapping is essential so that the floor planning, placement, routing and other functions can be carried out without error. This study analyses the literature on the netlist partitioning using various algorithms like genetic algorithm, Tabu search, simulated annealing, Bee colony, and Ant colony. Parameters such as area, power, delay and temperature were focused during the study.

Keywords: VLSI Cell partitioning, Ant colony, Particle Swarm Optimization, Tabu search, Genetic Algorithm, Support Vector Machine

I. INTRODUCTION

The VLSI industry has been growing vastly obeying Moore's law to an extent where numerous processors can be implemented on a single chip today. Yet, the exciting development is unable to keep up with the demands and challenges faced by the growth of device capacity and the expansion of a new user population in terms of VLSI design, especially in achieving scalability across the entire flow, without sacrificing the quality. There is an impending need to produce an efficient placement solution that can address high quality placement within a shorter time. Additionally, faster placement and routing tools that are flexible and robust to handle modifications in VLSI design styles and design objectives are required to handle complicated tasks. In the fiercely competitive VLSI field, the demand for devices to be faster, smaller and sophisticated with various updated technologies is ever increasing. In this paper various net list partitioning methods have been discussed. These are analyzed not only with reverence to area, delay and power optimizations, but also to prove efficiency of methods for temperature constraint. Mainly three iterative approaches are being most widely used, i.e., Genetic Algorithm, Tabu Search and Simulation annealing are discussed to solve the multi objective problems [1]. In most technology, i.e., 3D-ICs, which are being used widely, simulation annealing process proved to be efficient compared to Tabu search method [2]. Machine learning can be available option for detecting hotspots in a physical design. By solving the critical problem of class imbalance through generation of multiple hotspot clips from a single hotspot in the design, we are able to generate a large balanced dataset and thereby build models with a very high classification ac-curacy. The study mainly deals with four standard machine learning algorithms, viz. KNN - k-Nearest Neighbors, SVM - Support Vector Machine, DT - Decision Tree and RF - Random Forest. Once the configuration parameters of these algorithms are optimized and a sufficiently large training dataset is available ,all of them work very well, with RF showing the highest performance across different dataset sizes, clip sizes and number of features. However, considering the simplicity of the KNN algorithm, it shows very good results, with an F1-score only a few % below the RF models [16]. This study analyzed the literature to understand the Simulation annealing methods for overcoming timing analysis. In this paper, various algorithms are discussed with their contribution to the cell partitioning. In the optimization technique, performance can be improved either in terms of area, speed or power. Many hybrid versions are discussed, which improves the overall performance compared to the pure algorithms.

II. ALGORITHM FOR VLSI CELL PARTIONING PROBLEMS

A. Simulated Annealing

Simulation annealing [2], used for solving optimization problems, it is a common heuristic and also one of the well-developed iterative methods. The heuristic, the hill climbing capability, can be achieved by accepting impair cost. Although the probability of accommodating inferior solutions is huge in the initial stages, as the search progresses, only slighter impairs are accepted, until only high-quality solutions are accepted. In order to imitate the annealing process, a large amount of flexibility is allowed in the neighborhood creation at higher values, i.e., many 'uphill' moves are allowed at higher values of temperature. The search progresses with lower value of temperature parameter. Less uphill moves are accepted as the temperature is lowered. In actuality, at absolute zero, the simulated annealing algorithm turns greedy, allowing only downhill moves.

The iterative improvement scheme starts with several given state and examines a local neighborhood of the state for better solutions. A local neighborhood of

a state S, denoted by X(S), is the set of all states which can be reached from S by creating small change to S.
From analyzing various circuits, Simulation annealing produces high quality outcome [2]. The average percentage of the area deviation compared to the area of a layer was around 2.4% in addition to minimizing the total number required TSVs. At higher temperature, Simulation annealing outperforms the Tabu search.

B. Tabu Search

Tabu search (TS) is one of the trendy iterative heuristics to the circuit partitioning problem. It is a neighborhood search method which employs "wise" search and flexible reminiscence approach to turn possible solutions away from the worst solution. To de-emphasize the use of random choice and to be able to speed up the hunt technique, moves are decided intelligently on the premise of excellent admissible actions, which are decided on downhill and uphill movements. One of the benefits of Tabu search is to restrict the search space and avoid local premier.
Tabu search begins from a preliminary reasonable solution and extends its exploration by making a series of random movements or perturbations and maintains a Tabu search listing, which capture the attributes of some of the preceding actions. With each new release, a subset of neighbor solutions is generated by creating a wide variety of moves and the fine circulate (the pass that resulted within the high-quality solution) is widespread, and not within the Tabu search list. If the stated move is in the Tabu search list, it shows the most effective solution, i.e., it leads to an answer that is better than the satisfactory solution determined till date (aspiration criterion). For this reason, the aspiration criterion can override the Tabu search listing restrictions. The function of the move that is kept in Tabu search listing is the indices of the cells involved inside the interchange. The dimensions of Tabu search listing is calculated by the circuit size, i.e., 10% of the total range of cells.
Aspiration criterion. The aspiration criterion performs a critical function that helps in producing desirable performance, which could be crucial to the fulfillment of the Tabu search set of rules. To increase the ability of the set of rules, while keeping the basic capabilities that permit the algorithm to escape neighborhood top-rated and avoid cyclic behavior, aspiration is used to briefly launch an answer from its Tabu search. Distinctive applications use several aspiration criterions.
According to the first actual aspiration rule (ASP1),if the price related to a Tabu search solution is much less than the aspiration price related to the value of the present day answer, then the Tabu search status of the Tabu search solution is briefly ignored. This indicates that although the Tabu search answer is not eliminated from the Tabu search listing, its reputation is overridden and passed onto the Tabu search answer.
The second aspiration rule (ASP2) includes putting off a categorized as Tabu search when the move yields a solution higher than the first-class acquired to date. The number of blocks growth induces higher impact of the aspiration stage. The Tabu search regulations and aspiration degree criterion of Tabu search plays a twin role in constraining and guiding the hunt process.
By comparing the two heuristic, i.e., Tabu Search and Simulation annealing, in terms of search spaces, Tabu search exhibited better performance. With respect to good sub-spaces, i.e., evaluating solution with high membership values, Tabu Search proves to be more efficient. Further, it shows better TSV (Through Silicon Vias) count than Simulation Annealing.
Tabu Search outperforms Genetic Algorithm in terms of execution time and final solution costs [1].The difference between the two became higher when circuit complexity increased. The superiority of Tabu Search can be attributed to its directed search approach as well as to its higher greediness tendency to obtain a good solution compared to Genetic Algorithm.

C. Genetic Algorithm

Genetic algorithms have been reported for placement of components, which are assigned to physical locations on chip. Cohoon [3], Shahookar [4], and Sait [5] have demonstrated that genetic algorithms to produce better results than simulated annealing. Genetic algorithm provides timing and power driven partitioning. Being an iterative procedure, a genetic algorithm maintains a population of individuals [7] and iteration is referred to as a generation. For solving the problem, these individuals act as candidate solutions and there health is calculated. Crossover and mutation are applied in which new offspring are produced. The generated offspring inherit a little quality from its mother and father.
The mutation operator is used to introduce new random records inside the population. It is also applied after the crossover operator. It facilitates the production of a few versions within the answers so that the search no longer gets trapped in the local minima. An illustration of mutation operation is the exchanging of two randomly decided genes of a chromosome. But mutation is functional with a low rate so that GA (Genetic Algorithm) no longer develops into reminiscence-much less search method. In this method, two mutation versions are used; the first one uses random choice of a mobile cell and swaps its partition. The second one randomly decides on one cell from every partition and then swaps them. For addressing a multi-objective optimization problem, for minimizing three mutually conflicting goals, and for comparing the fitness of a solution, fuzzy membership functions and fuzzy regulations are used. The fitness price of a chromosome is its membership value, which is calculated inside the fuzzy set of an ideal answer. Individuals are selected based entirely on the elitism random selection (ernd), where the great Np=2 chromosomes are selected and the final Np=2 are decided randomly. Primarily based on experimental effects, this scheme gives better results than other schemes, as it provides stability between greediness and randomness. Circuit partitioning is one of the key areas in chip designing. The algorithm can partition large circuit into a number of smaller circuits that reduces the complexity of the VLSI circuit design [8].
For getting near optimal result, Genetic Algorithm can be used; however, its run time has been found to increase quickly as the circuit size is increased. This execution time issues can be condensed by applying various local search optimization methods in most stages of every generation[1][7].Also it is proved that by using various scheme of Genetic Algorithm, such as the steady-state GA, it can increase the speed by 20% and 3% increase in circuit size compared to the generational GA[9-10].
In the case of partitioning instances overall size ranges, GA based partitioning are consistently better than those obtained by Ant Colony Optimization (ACO) based partitioning [11].
D. Ant Colony Optimization ACO is a probabilistic technique used for solving complex computational problems and for searching optimal path through graph.

In 1992, Marco Dorigo proposed the ACO, which replicates the performance of ants seeking a path between the colony and the source of food.
In the real world, ants of same species roam randomly for finding food for the colony. On finding food, it returns to its nest by laying a pheromone trail to its path in order to recognize the path back to the location. Further ants that find the path would stop roaming and follow the trail. Thus, the shortest path to food location from nest is identified. As time passes, pheromone trail evaporates reducing the chance of its discoverability to other ants. A smallest path, by comparison, get used over more frequently, therefore, the pheromone density become superior on shorter path than in the longer one.
The performance of ACO and genetic algorithm are recorded in circuit partitioning instances (net list) as given in the MARCO GSRC VLSI CAD Book shelf-website [12].It is seen that the genetic algorithm based partition are better than the ACO based partitions for all partitioning instances overall size ranges. Both are successfully applied to the non-polynomial hard problems but genetic algorithm outperforms ACO.
When genetic algorithm is combined with ACO for local search it may reduce the runtime [12].

E. BEE Colony Optimization (BCO)

BCO [13] is a swarm based meta heuristic algorithm, and it was first identified by Sato and Haigwara in 1997.It is considered as an emerging technique in the area of optimization as it provides immense problem solving scope for combinatorial and NP hard problems. The BCO is used to generate the multi agent system (colony of artificial bees) to resolve difficult combinatorial optimization problems.

Behavior of honey bee swarm. BCO is a smallest model of forage selection that leads to the emergence of honey bee swarm. It consists of three necessary components, namely, food sources, employed foragers, and unemployed foragers.

-Food sources

The importance of a food sources depends on its proximity to the nest, its richness, concentration of its energy, and the ease of extracting of this energy.

-Employed foragers

They are associated with a particular food sources which exploit or also referred to as employed. It carries the information about the distance and direction from the nest and shows this information with a certain probability.

-Unemployed foragers

There are two categories of unemployed foragers: scouts, searching the environment surrounding the nest for new food sources and onlookers, waiting in the nest and establishing a food source through the information communal by employed foragers. The mean number of scouts averaged over conditions is about 5-10%.

The BCO algorithm in general shows behavior of real bee colonies. Bees are able to calculate their present location from their past trajectory continuously. They can return to their starting point by choosing the direct route rather than retracing their outbound trajectory. This algorithm limitation is less adaptive in nature.

F. Particle Swarm Optimization (PSO)

PSO is an optimization algorithm based on population. Kennedy and Eberhart proposed PSO in 1995.It is expected to evolve by cooperation and competition amongst the individuals themselves through generation instead of using genetic operators.

Each individual with a velocity in PSO flies in the explore space is adjusted to its own experience of flying. Every element in the PSO will be treated as a volume-less particle in the d-dimension search space. In the problem space, each individual watches the assignment of its coordinates. This is associated with the best solution (fitness) that is achieved so far and is called as pbest. The universal version of the PSO is the total best value, and its location is called as gbest. At each point, the PSO algorithm consists of velocity changes of each particle towards its pbest and gbest locations. Velocity changes are recorded at each time step of each particle towards its pbest and gbest locations.
In paper [7], the partitioning of the circuit is denoted by graph and then examined for the run-time of task graphs in Genetic Algorithm, which is almost equal to those of PSO, because run-time is mostly dedicated to fitness function of evolutionary. However, Genetic algorithm takes more time than PSO in small task graphs, i.e., for a fixed number of number of generations in an exceptional case, and PSO takes more time in 992/1083 nodes/edges task graph. Further [6] also discusses discrete PSO which outperforms genetic algorithm in all cases.

G. Markov Renewal Process Model

Markov renewal reward process is an extremely useful tool for proficient cell partitioning; therefore, an exploration of this approach was conducted. While analyzing the Markov Renewal Reward method for VLSI circuit partitioning, a challenge was made to investigate the probability distribution of the accumulated reward in a Markov renewal process and to obtain the accumulated reward that is directly influenced by random process that can be modeled by continuous-time Markov chain. In doing so, two novel approaches were employed [17-18]. The main focus of this study is on influential the optimal criterion for the total expected discounted cost as it is initiate to be more appropriate and essential, particularly in biomedical and consistency studies. Alternatively, one could think on a long-run average cost per unit time, which is more appropriate when many state transitions occur within a relatively short time, just as in the stochastic control problem in telecommunication applications. The inter arrival- time of MARP can be extended to a minimum delay of VLSI Circuit partitioning problems. The formulae states in MARP formulism are generally less transparent and less explicit. It is expressed in terms of the Markov renewal process and finite number of states in each VLSI cell partitioning that can be constructed. The solution of the MARP gives the asymptotic behavior and generalized Markov renewal reward equation on VLSI Cell partitioning with max-position level. Finally, the max-position level in each VLSI Cell partitioning is attained.

H. Four Machine Learning Algorithm For VLSI Cell Partitioning Problems

The study of machine learning can be a viable option for detecting hotspots in a physical design particularly in VLSI Cell partitioning problems. By solving the critical problem of class imbalance through generation of multiple hotspot clips from a single hotspot in the design, we are able to generate a large balanced dataset and thereby build models with a very high classification ac-curacy. The study mainly focused on four standard machine learning algorithms, viz. KNN - k-Nearest Neighbors, SVM - Support Vector Machine, DT - Decision Tree and RF - Random Forest. Once the configuration parameters of these algorithms are optimized and a sufficiently large training dataset is available, all of them work very well, with RF showing the highest performance across different dataset sizes. However, considering the simplicity of the KNN algorithm, it shows very good results compared with RF models. By utilizing a large balanced dataset, we have studied the effect of various parameters on the model accuracy. These parameters include - the diameter of the layout clips, the length of the density vector and the size of the training dataset. Based on our study, a larger training dataset size and a longer density vector, both improve the model prediction accuracy. Thus, a longer density vector with more feature scan compensate for a smaller training dataset size and vice versa. This complementary behavior is significant for efficient utilization of computational resources. A more comprehensive study is needed to fully understand all the effects of clip size on prediction accuracy with respect to cell partitioning problems [16].

III. CONCLUSION

In this study, various algorithms have been discussed along with their advantages and drawbacks with respect to area, power and delay optimization. The analysis of the study shows that the hybrid versions can perform better than the pure versions. Compared to the genetic algorithm and simulation annealing, Tabu search, ACO and BCO could show better speed. Among four machine learning algorithms SVM algorithms is shown better result for cell partitioning problems. Future research of these techniques is to proposed techniques extended to enhance their functionality and performance. The several possibilities that can be investigated for extending these basic techniques is reducing delay in the placement phase in clustering models, and constructing the scalable parallel partitioning algorithm for two-dimensional distributions in VLSI cell placement. VLSI Cell partitioning problems will be addressed by the deep learning algorithm to achieve efficient results.

Conflict of Interest: Nil

REFERENCES

[1]. Sait, S.M., El-Maleh, A.H. and Al-Abaji, R.H. (2006). Evolutionary algorithms for VLSI multi-objective netlist partitioning. *Engineering applications of artificial intelligence*, **19**(3): 257-268.

[2]. Sadiq, M. Sait, Feras Chikh, Oughali and Mohammed, Al-Abaji, (2016). "Original design partitioning and layer assignment for 3D integrated circuits using tabu search and simulated annealing", *Journal of Applied Research and Technology,* vol. **14**, pp. 67-76.

[3]. Cohoon, J.P. and Paris W.D, (1987). "Genetic Placement", IEEE Trans. on Computer-Aided Design of Integrated Circuits, Vol. **CAD-6**, issue 6, pp. 956-964.

[4]. Shahookar, K. and Mazumder, P. (1990). "A genetic approach to standard cell placement using meta-genetic parameter optimization", *IEEE Trans. on Computer-Aided Design of Integrated Circuits*, Vol. **9**, issue 5, pp. 500-511.

[5]. Sait, S.M., Youssef, H., Nassar, K and Benton, M.S., (1995). "Timing driven genetic algorithm for standard-cell placement", in *Proc.14th Phoenix Conf. on Computers and Communications,* pp. 403-409.

[6]. Holland, J.H. (1975). "Adaptation in natural and artificial systems: an introductory analysis with applications to biology", Control and Articial Intelligence. University of Michigan Press.

[7]. Amin Farmahini-Farahani, Mehdi Kamal, Sied Mehdi Fakhraie and Saeed Safari, (2007). "HW/SW partitioning using discrete particle swarm" in GLSVLSI Proceedings of the 17th ACM Great Lakes Symposium on VLSI, pp. 359-364.

[8]. Saini, P. and Kaur, M. (2013). "Study of the Circuit Partitioning Using Genetic Algorithm". *International Journal of Emerging Technology and Advanced Engineering*, vol. **3**, issue 4.

[9]. Davis L. (ed.), (1991). Handbook of Genetic Algorithms, Van Nostrand Reinhold.

[10]. Yoshida, N., Moriki, T. and Yasuoka, T., (1997). "GAP: Generic VLSI processor for Genetic Algorithms", Proc. 2nd Int'l ICSC Symp. on Soft Computing, pp. 341-345.

[11]. Gill, S.S., Chandel, R. and Chandel, A. (2009). "Comparative study of Ant Colony and Genetic Algorithms for VLSI circuit partitioning". *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. **3**, issue 4.

[12]. http://www.gigascale.org/bookshelf

[13]. Chikhalikar, A. and Darade, A. (2013). "Swarm intelligence techniques: Comparative study of ACO and BCO" CSI, Mumbai, pp. 1-9.

[14]. Kaur, Maninder and Singh, Kawaljeet (2012). "An Algorithm for Min Cut Partitioning for Digital Circuit Layout based on Evolutionary Approach". *IRACST – Engineering Science and Technology: An International Journal*, Vol. **2**, issue 4.

[15]. Fiduccia, C.M. and Mattheyses, R.M. (1982). "A linear-time heuristic for improving network partitions," *Proc. Design Automation Conf.,* pp. 175-181.

[16]. Verma, P., Pack, R. and Madhavan, S. (2018). "A Systematic Study of Hotspot Detection in Physical Designs using Machine Learning" 19th Int'l Symposium on Quality Electronic Design, pp 302-307.

[17]. Manikandan, R. and Swaminathan, P. (2013). "Analysis of Markov renewal reward process on VLSI cell partitioning", *Applied Mathematics Science*, Vol. **7**, no. 66, pp. 3281-3289.

[18]. Manikandan, R., Swaminathan, P. and Vaithiyanathan, V. (2013). "Study on Markov alternative renewal reward process for VLSI cell partitioning", *Int. J. Math. Anal.*, Vol. **7**, no. 40, pp. 1949-1960.

How to cite this article: Manikandan, R., Parameshwaran, R., Prassanna, J. and Sekar, K.R. (2019). A Study on Specific Computational Algorithms for VLSI Cell Partitioning Problems. *International Journal on Emerging Technologies*, **10**(2): 67-70.