

# On solving Mincut Balanced Circuit Partitioning Problem for Digital Circuit Layout using Evolutionary Approach with Solution Archive

Maninder Kaur  
Assistant Professor, SMCA  
Thapar University, Patiala

Kawaljeet Singh  
Director, University Computer Center  
Punjabi University, Patiala

## ABSTRACT

The interest in finding an optimal partition in the area of VLSI has been a hot issue in recent years. Circuit Partitioning Problem is one of the most studied NP complete problems notable for its broad spectrum of applicability in digital circuit layout. The balanced constraint is an important constraint that obtains an area balanced layout without compromising the mincut objective. This paper proposes a non revisited algorithm based evolutionary approach ((NRECP) for balanced circuit min cut Partitioning in VLSI physical design automation which uses binary tries to efficiently store all evaluated solutions during the heuristic search and to effectively transform the solutions into unconsidered candidate solutions in case of solution revisit.

## General Terms

Evolutionary Approach, Hypergraph Partitioning, Revisits.

## Keywords

Binary trie, Cyclic crossover, NP-complete.

## 1. INTRODUCTION

Circuit partitioning problem is used in many areas of VLSI layout and design, such as floorplanning, placement and multiple-chip/multiple-FPGA partitioning. The min-cut balanced bipartition problem was shown to be NP-complete [13]. Because of its applicability in many areas, many heuristic algorithms have been devised for its solution. Few well-known heuristics are Kernighan and Lin type (K&L) iterative improvement methods [1,3], simulated annealing approaches [11], and analytical methods for the ratio-cut objective [7]. In the literature for circuit partitioning problem, min cut partitioning using unit area is more prominent and the implementation of a partitioning algorithm is much simpler with unit areas[5]. VLSI circuit balanced partitioning is a combinatorial optimization problem where every node has varying node weight with weight typically representing cell area. This paper deals with the problem of solving min cut partitioning with non-unit areas of circuit elements

Given a hypergraph representation of circuit, the balanced circuit partitioning problem divides the nodes of a hypergraph into partitions of approximately equal weight satisfying balance constraints while minimizing the number of hyperedges across the partitions.

Evolutionary algorithm [12] is the popular class of heuristic algorithms for solving optimization problems. Evolutionary algorithms are able to find good approximate solutions within a huge search space in relatively short computation times but sometimes lead to problem of local convergence. To tackle

with this problem attention was paid to various hybrid evolutionary approaches [2,6,9,10,12,14] with local improvement for solving circuit partitioning problem in the recent years. This paper is based on the idea of tackling the problem of local convergence by avoiding the revisit of evaluated solutions. The proposed evolutionary approach takes the help of binary trie [4,13] to store the solutions encoded in the form of binary strings. In case of solution revisit, the algorithm further transforms the solutions into yet unconsidered candidate solutions.

## 2. PROBLEM FORMULATION

Given a hypergraph representation of circuit  $G = (V, E)$  with  $V = \{v_1, v_2, v_3, \dots, v_n\}$  as set of  $n$  vertices representing cells of circuit, and  $E = \{n_1, n_2, n_3, \dots, n_e\}$  as set of  $e$  hyperedges representing nets in the circuit. Let  $a(v_i)$  with  $i \in [1, n]$  denotes the area of  $v_i^{th}$  cell. The balanced min cut circuit bipartitioning consists of dividing the circuit into two partitions  $V_1, V_2$  while minimizing the number of cuts across the partitions is stated as

$$\sum_{i=1}^j \sum_{i=1}^k c_{ij} \quad (i \neq j) \text{ is minimized}$$

Where  $c_{ij}$  represents the crossing edge from node to  $i$  node  $j$  crossing a partition.

$$r - \delta \leq \frac{|V_1|}{|V_1| + |V_2|} \leq r + \delta$$

Where  $|V_1|$  and  $|V_2|$  denotes the size of partitions  $V_1$  and  $V_2$  respectively such that

$$|V_j| = \sum_{v_i \in V_j} a(v_i) \quad \text{for } j=1, 2.$$

balance factor,  $r=0.5$  and  $\delta$  denotes the tolerance limit.

The min cut problem is NP complete, it follows that general partitioning problem is also NP complete [8]. The circuit bipartitioning optimization is focused on finding an acceptable solution cut-set cost. The cut-set cost is the number of inter-partition connects, which if not selected carefully, will immensely degrade the overall solution quality

## 3. THE NRECP ALGORITHM

The work proposes a Non Revisited Evolutionary Approach for Circuit Partitioning (NRECP) algorithm where the archive is consulted each time after a new solution is generated by crossover. The following describes the various components of the algorithm.

Let K be the number of sub circuits into which the circuit with graph G is divided, and let n (n=|P|) be the number of logic gates of the original circuit; then each solution is represented by an array S of n elements as S= $p_1, p_2, p_3, \dots, p_n$  with  $p_i \in [0, K-1]$  where the  $p_i$  element in array S represents the subcircuit to which the logic gate i belongs to. In this proposed algorithm the value of K is 2. The circuit graph is traversed in a depth-first way for ordering of vertices.

Pseudocode of the algorithm

Step 1: Randomly generate an initial population P with set of feasible solutions (reading .are files).

Step 2: Insert the population in the binary tries

Step 3: Read the input file (.net files) and convert it into netlist format. Calculate the fitness value of each solution in the population using the net cut evaluation mechanism. For a net cut evaluation a multiword mask of size of the chromosome is pre computed for each net. If a cell is connected to net, the corresponding bit position is set.

$$M_{ij} = \begin{cases} 1 & \text{if } C_j \in N_i \\ 0 & \text{otherwise} \end{cases}$$

Where  $C_j$  is the  $j^{th}$  cell in order.

$M_{ij}$  is the mask for net  $N_i$  and is the  $j^{th}$  bit position of  $M_i$

The value of  $CM_i$  and  $\overline{CM}_i$  is evaluated. If both values are nonzero then net is present in both partitions, hence a cut. Otherwise no cut.

Step 4: After evaluating all the population with the fitness function, the individuals of the next generation will be chosen by a proportional criterion, called roulette proportional criterion, which will guarantee that the best individuals of the current generation have more possibilities of passing to the next one.

Step 5: The cyclic crossover operation is applied at random points on the selected individuals to generate two offsprings.

Step 6: Check the feasibility of new solutions by examining whether the solutions satisfy the balance constraints. If not then repeatedly mutate the solution by inverting the bit positions at random point until getting feasible solution. In NRECP algorithm, the operators are based on the moves of gates between neighborhood partitions. The variants are deterministic, pseudo-random and random.

Step 7: Check the presence of the newly generated offsprings in solution archive. If any of the newly generated solution already present in archive then solution archive generates a new unvisited solution by level order traversal of binary trie structure and also insert that that solution into solution archive. Otherwise insert the new offspring into archive.

Step 8: If the new solution is given by archive then check the feasibility of new solutions. If satisfying the balance constraint then accept the solution. Otherwise again retrieve the new solution from the archive. This process is repeated until the solution satisfy the balance constraint

Step 9: The algorithm is repeated for some set of generations.

The trie structure has been pruned to cut out the number of comparisons for searching and transforming the solution, which further keep a check on maximum comparisons for searches.

**4. SIMULATION RESULTS**

The algorithm is tested on the ISPD '98 IBM benchmark [14] by reading .net files for interconnections and .are files for weights of vertices and converting them into netlist format. The work is carried out by writing code in MATLAB (version 9) and running on an Intel Core i5 (2.60 GHz) machine with 4 GB memory, using balance factor, r as 0.5, crossover rate of 0.75 and population Size 10.

Table 1 gives runtimes and average solution qualities for 10 runs of proposed NRECP algorithm and hMetis on the ISPD '98 IBM benchmark suite with partitioning tolerances of 2 and 10%. Both the minimum and average cuts over 10 runs of each algorithm are reported. Smaller net cut is better. The size of benchmarks range from vertices 12752 in ibm01 to vertices 71076 in ibm12. Both the minimum and average cuts over 10 runs of each algorithm are reported. The CPU column gives the average time (in seconds) required for a single run of each algorithm.

By curve fitting this data, it is found empirically that the runtime and memory usage of both partitioners grows nearly linearly with the size of the benchmark. The data is expressed as ratio of average cut and average CPU time(in sec) Figure 1 and 2 shows the performance of the NRECP by comparing ratios with partitioning tolerances of 2% (each partition must consist of between 49% and 51% of the total area) and 10% (each partition must have between 45% and 55% of the total vertex area) allowing up to 2% deviation and 10% deviation from exact bisection respectively.

Both Figure 1 and 2 depicts that NRECP generally gives better average cut /average CPU time ratio for smaller circuits in comparison with hMetis, producing better partitioning results but the performance gradually decreased as the circuits grow in size.. Both hMetis and NRECP produces better solutions when the tolerance is higher. For number of vertices less than 50,000 the NRECP approach gives good results but the performance gradually decreases for larger circuits. As the number of vertices increases it directly affects the depth of the binary trie. The reason for decrease in performance lies in increase in time consumed by the algorithm in searching the trie structure.

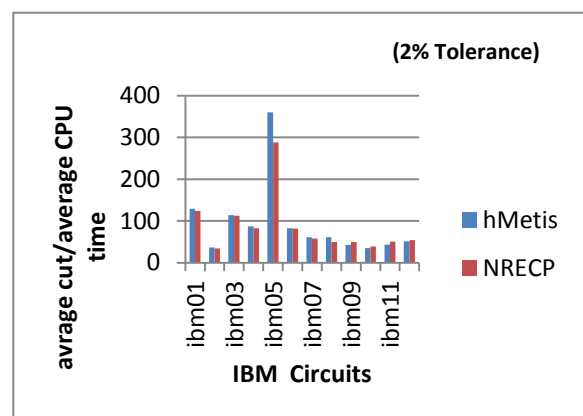


Fig 1: Shows the performance of NRECP and hMetis with 2% Tolerance

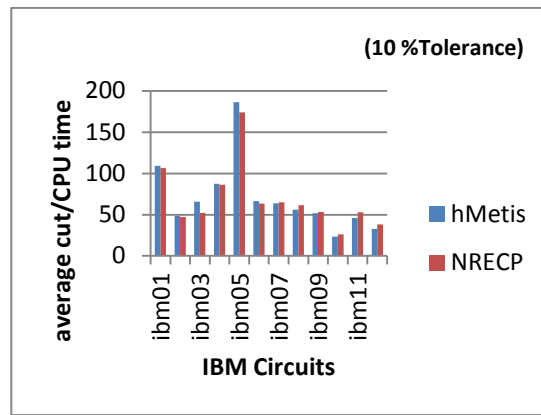


Fig 2: Shows the performance of NRECP and hMetis with 10% Tolerance

Table 1. Comparison of hMetis and NRECP based on Minimum, average, CPU Time (in sec) with 2% and 10% deviation from exact bisection respectively

Bench ark circuits	2% balance tolerance						10% balance tolerance								
	# of Nodes	# of Nets	hMetis			NRECP			hMetis			NRECP			
			Min	Avg	CPU	Min	Avg	CPU	Min	Avg	CPU	Min	Avg	CPU	
ibm01	12752	14111	188	297	2.3	169	223	1.8	188	262	2.4	176	213	2	
ibm02	19601	19584	113	200	5.5	107	166	4.87	121	228	4.7	111	199	4.22	
ibm03	23136	27401	427	629	5.5	418	600	5.34	234	341	5.2	220	298	5.7	
ibm04	27507	31970	458	582	6.7	440	503	6.12	444	525	6.0	400	431	5	
ibm05	29347	28446	1745	3490	9.7	1603	2560	8.88	1744	1828	9.8	1600	1358	7.8	
ibm06	32498	34826	498	836	10.1	491	799	9.76	491	685	10.3	455	603	9.5	
ibm07	45926	48117	868	1074	17.6	880	1072	18.5	818	1030	16.1	800	950	14.65	
ibm08	51309	50513	1272	1426	23.4	1190	1386	27.95	1178	1343	24.0	1050	2145	34.98	
ibm09	53395	60902	572	754	17.8	580	901	18.25	573	780	15.1	550	822	15.37	
ibm10	69429	75196	629	797	22.8	650	789	20.33	286	515	22.1	240	720	27.68	
ibm11	70558	81454	801	1202	27.6	793	1589	31.45	756	1107	24.0	650	1399	26.35	
ibm12	71076	77240	1297	1740	34.0	1315	2208	40.89	472	965	29.5	450	940	24.67	

## 5. CONCLUSIONS

The work presents an approach for balanced circuit partitioning problem combining evolutionary computation with solution archive which further eliminates the redundant solutions, hence avoiding the algorithm getting trapped in local convergence. The mechanism for pruning and generation of feasible solutions is embedded into the algorithm to reduce the number of searching comparisons in the solution archive. The NRECP approach gives good results for vertices less than 50,000 but the performance gradually decreases for larger circuits. The approach can be further enhanced by reducing the number of search comparisons in binary trie by pruning tries further based on the fitness values for individual solutions.

## 6. REFERENCES

- [1] B. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning of electrical circuits," Bell Syst. Tech. J., pp. 291–307, Feb.1970
- [2] C. J. Alpert, L. W. Hagen and A. B. Kahng, "A Hybrid Multilevel/Genetic Approach for Circuit Partitioning," ASPDAC, 1996, pp. 298-301
- [3] C.M. Fiduccia and R. M. Mattheyses, "A linear time heuristic for improving network partitions," in Proc. ACM/IEEE Design Automation Conf., 1982,pp. 175-181
- [4] G.R. Raidl and B. Hu. Enhancing genetic algorithms by a trie-based complete solution archive. In Evolutionary

- Computation in Combinatorial Optimization Evolutionary Comp. 2010, volume 6022 of LNCS, pages 239-251. Springer, 2010.
- [5] J. Cong, L. Hagen, and A. Kahng, “Net partitions yield better module partitions,” in Proc. 29th ACM/IEEE Design Automation Conf., 1992, pp. 47–52.
- [6] J-P. Kim, Y-H. Kim, and B-R. Moon. A hybrid genetic approach for circuit bipartitioning. In Proc. Genetic and Evolutionary Computation Conference (GECCO), volume 3102 of LNCS, pages 1054–1064. Springer, 2004.
- [7] L. Hagen and A. B. Kahng, “Fast spectral methods for ratio cut partitioning and clustering,” in Proc. IEEE Int. Conf. Computer-Aided Design, Nov. 1991, pp. 10–13.
- [8] M. R. Garey and D. S. Johnson, .Computers and Intractability: A Guide to the Theory of NP Completeness,. San Francisco, CA: Freeman, 1979
- [9] Mehdi Farshbaf, Mohammad-Reza Feizi Derakhshi, Arash Roshanpoor, Multi-objective optimization using hybrid genetic algorithm and cellular learning automata applying to graph partitioning problem. HIS2011:722 727
- [10] R. Chandrasekharam, S. Subhramanian and S. Chaudhury, Genetic algorithm for node partitioning problem and applications in VLSI design, IEE Proc. E (Comput. Digital Techniques) 140(5) (1993) 255-260.
- [11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” Sci., pp. 671–680, May 1983.
- [12] Sadiq M. Sait, Aiman H. El-Maleh, Raslan H. Al-Abaji: Evolutionary algorithms for VLSI multi-objective netlist partitioning. Eng. Appl. of AI 19(3): 257-268 (2006)
- [13] S. Y. Yuen and C. K. Chow. A non-revisiting genetic algorithm. In IEEE Congress on Evolutionary Computation (CEC 2007), pages 4583{4590. IEEE Press, 2007
- [14] C.J. Alpert, The ISPD98Circuit Benchmark suite, in Proc. ACM Int. Symp. on Physical Design, 80-85, April 1998. [.http://vlsicad.ucsd.edu/UCLAWeb/cheese /ispd98. html](http://vlsicad.ucsd.edu/UCLAWeb/cheese /ispd98. html)