Comparative Study of Various Evolutionary Approaches for Digital Circuit Layout based on Graph Partitioning Technique

Maninder Kaur
Assistant Professor
SMCA,Thapar University,Patiala

Kawaljeet Singh
Director,University Computer Center
Punjabi University,Patiala

ABSTRACT
The paper presents the comparison of various evolutionary approaches for digital circuit layout based on graph partitioning technique. The comparison of the algorithms is done by taking number of iterations, memory requirements and minimum cuts during computation. The algorithms are verified by comparison with existing approach. The experimental work uses UCLA spp-circuit benchmark series and results are statistically analyzed using ANOVA

1. Keywords
Circuit Layout, Hypothesis, Partitioning, DNA Computing.

2. INTRODUCTION
Graph layout problems are a particular class of combinatorial optimization problems whose goal is to find a linear layout of an input graph in such way that a certain objective function is optimized. A linear layout is a classification of the vertices of a graph with distinct integers. A large amount of relevant problems indecent domains can be formulated as graph layout problems. These include optimization of networks for parallel computer architectures, numerical analysis, VLSI circuit design, information retrieval, graph theory, computational biology, scheduling and archaeology. Moreover, the minimal values of some layout costs are also related to interesting graph theoretic invariants of graphs. Most interesting graph layout problems are NP-hard and their decisional versions NP-complete, but, for the majority of their applications, the feasible solutions with an almost optimal cost are sufficient. As a consequence, approximation algorithms and effective heuristics are preferred in practice.

Circuit partitioning is the task of dividing a circuit into smaller parts .A chip may contain several million transistors. The layout of the entire circuit cannot be handled due to the limitations of memory space as well as computation power available. As a result, layout area is normally partitioned by grouping the components into blocks (sub circuits/modules). A more important use of circuit partitioning, is to divide up a circuit in hierarchical manner into parts with divide and conquer algorithms for floorplanning, placement, and other layout problems[1-6]. Here, cost measures to be minimized during partitioning may vary, but mainly they are similar to the connection cost measures for general partitioning problems. The actual partitioning process considers many factors such as: number of the blocks, size of blocks and number of interconnections between the block – the netlist. The result of partitioning is a set of blocks along with the interconnections required by blocks.

In this work, comparative analysis of five proposed evolutionary approaches namely Hybrid Artificial Bee Colony Optimization with Simulated Annealing for Circuit Partitioning (HABCSACP)[7], Non Revisited Evolutionary Approach for Circuit Partitioning (NRECP) algorithm[8], Extended-Non Revisited Evolutionary Approach for Circuit Partitioning algorithm (E-NRECP)[9] , Soft Computing Algorithm for Partitioning (SCAP) Approach[10], DNA Based Approach for Circuit Partitioning (DBACP)[11], for the optimization of VLSI netlist bi-Partitioning is carried out. These approaches are based on soft computing, DNA computing, simulated annealing, and artificial bee colony and trie data structures. For simulation work a set of UCLA benchmark SPP series is used to evaluate the efficiency of the algorithms.

3. COMPARATIVE ANALYSIS OF VARIOUS PROPOSED ALGORITHMIC APPROACHES
The performance of the proposed algorithms is tested on spp- circuit series of UCLA small circuit partitioning instances generated by the top-down partitioning-based placement process employed by the UCLA Capo placer (http://vlsicad.ucsd.edu / GSRC /bookshelf/Slots/Partitioning/ smallPP.html). These circuits are given in multiple number of partitioning instance groups in each size range. The circuit net lists are in the nodes/nets/wts format. The average results from the proposed algorithms have been compared with those obtained by the UCLA branch and bound partitioner.

Table 1 gives runtimes and average solution qualities for multiple instances partitioning groups in each size range of proposed NRECP,HABCSACP, E-NRECP Algorithm and UCLA Branch & Bound on the SPP-benchmark suite with 10 % partitioning tolerance. The average cuts over instances partitioning groups in each size range of the proposed algorithm are reported. Smaller net cut is better. The sizes of benchmarks range from 10 vertices (in spp_N10 series) to 60 vertices (in spp_N60 series). The CPU column gives the average time (in seconds) required for a single run of each algorithm.

The results of DBACP approach are compared with that of SCAP algorithm which uses the same method of solution encoding over a set of spp Benchmark circuits. The tabulated results of comparison are given in Table 2. The parameter values used for SCAP are population size 10, crossover probability 0.6, mutation probability 0.02, and number of generations 50 .Nodes are assigned actual cell areas. Solutions are constrained to be within 10 % of bisection i.e. 45% to 55% of total partitioned area. The algorithms were simulated using 1,2,3,4 processors and the results presented the average cut and average runtime. Data is expressed as average cut (average CPU time) which have been obtained on multiple number of partitioning instance groups in each size range.

As seen from Table 3, average results obtained by DBACP based partitioner are consistently better than those

obtained by SCAP algorithm. With the increase in the size of problem instance the DBACP execution time increases in comparison to SCAP algorithm. The DBACP algorithm gives excellent quality of solution at the cost of running time of algorithm.

Various proposed approaches are compared in terms of solution quality, running time and the memory utilized by the underlying approach. The comparative analysis of various proposed algorithmic approaches in terms of average min cut and average CPU time over a set of benchmark circuits with node size varying from 10 to 30 nodes are shown in Figure 1 and Figure 2 respectively.

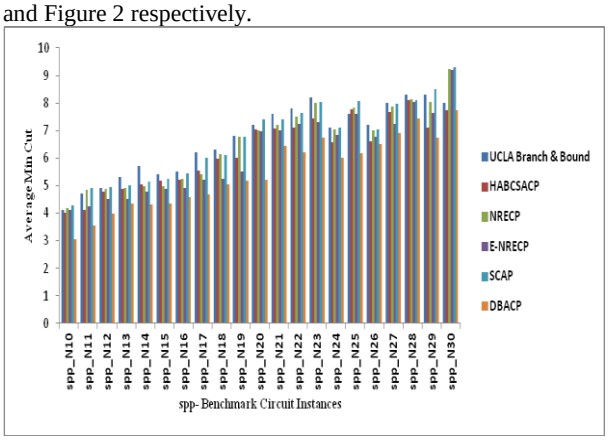

Figure 1: Average min cut of various proposed algorithmic approaches over a set of spp –benchmark circuit instances

The E-NRECP approach gives better performance than NRECP in terms of both average mincut and running time of algorithm. In addition to this, the trie data structure in the NRECP approach consumes more space as in comparison with that of E-NRECP approach. The reason of less memory utilized by the efficient trie of E-NRECP approach is that the trie stores only a set of feasible solutions whereas the trie in NRECP approach stores all possible permutations of the solution bits. The number of search comparisons is further improved by pruning the trie in both the approaches

The SCAP approach uses the same encoding technique for solutions as that of DBACP and is implemented in parallel computing environment

This approach does not provide better net cut and running time in comparison to DBACP in parallel computing environment. The approach does not provide satisfactory results even in case of uniprocessing in comparison to other proposed approaches

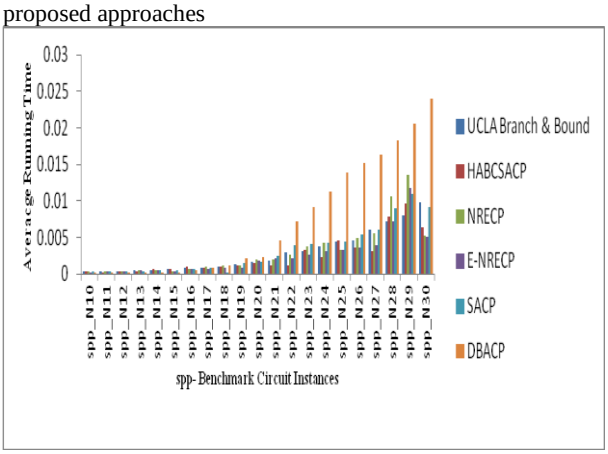

Figure 2: Average CPU time elapsed of various proposed algorithmic approaches over a set of spp –benchmark circuit instances

The HABCSACP approach provides consistent results in terms of both average cut and average run time in comparison to rest of proposed approaches.

Figure 2 concludes the comparative analysis of running time of various algorithmic approaches for the set of circuits ranging from 10 to 30 gates. There is an increase in the running time of DBACP algorithm as circuit size goes beyond 18 gates. When the comparative analysis is extended to circuits with 60 gates, it was concluded that the running time of the NRECP algorithm elevated for number of vertices greater than 54 nodes while the algorithms HABCP,E-NRECP and SCAP have comparative values of running time as that of UCLA Branch & Bound partitioner.

Table 1: Results containing average cut and average runtime of HABCSACP, NRECP, E-NRECP Algorithm and UCLA Branch & Bound partitioner on the small partitioning instances with 10% deviation from exact bisection.

Circuit Series	No. of Nodes	No. of Files	UCLA Branch & Bound Partitioner		HABCSACP		NRECP		E-NRECP	
			Average cut	Average Runtime	Average cut	Average Runtime	Average cut	Average Runtime	Average cut	Average Runtime
spp_N10	10	483	4.1	0.000350	4.031	0.0003	4.166	0.00028	4.1293	0.00025
spp_N15	15	184	5.4	0.000630	5.17	0.00059	4.989	0.0004	4.887	0.000362
spp_N20	20	121	7.2	0.001600	7.05	0.00152	7.01	0.0019	6.988	0.001888
spp_N25	25	107	7.6	0.004380	7.78	0.00459	7.83	0.00321	7.59	0.00331
spp_N30	30	52	8.0	0.009840	7.75	0.00632	9.23	0.00518	9.2116	0.005022
spp_N35	35	31	10.4	0.060920	10.5	0.05999	10.17	0.05761	9.94	0.05699
spp_N40	40	41	8.4	0.217860	8.4	0.18672	8.12	0.1501	7.63	0.1522
spp_N45	45	28	11.2	0.684830	10.75	0.65	10.52	0.6091	10.2	0.6082
spp_N50	50	24	10.5	3.939020	10.25	3.7852	10.36	4.7859	10.125	4.76691
spp_N55	55	20	11.7	23.64880	11.6	23.24723	13.01	30.9481	12.69	30.7899
spp_N60	60	9	11.6	31.75144	10.8	30.7864	12.35	45.0098	12.02	45.812

Table 2: Comparison of Branch - Bound, SCAP and DBACP based on average cut, average CPU Time (in sec) with 10% deviation from exact bisection respectively.

Circuit Series	No. of Nodes	No. of Files	UCLA Branch& Bound Partitioner		SCAP		DBACP	
			Average cut	Average Runtime	Average cut	Average Runtime	Average cut	Average Runtime
spp_N10	10	483	4.10	0.000350	4.293	0.000310	3.06	0.0001120
spp_N15	15	184	5.40	0.000630	5.23	0.000462	4.36	0.0001192
spp_N20	20	121	7.20	0.001600	7.39	0.001638	5.204	0.0023280
spp_N25	25	107	7.60	0.004380	8.07	0.004410	6.18	0.0133710
spp_N30	30	52	8.00	0.009840	9.289	0.009085	7.75	0.0239083

Table 3: Comparison of reported DBACP results with those produced by SCAP methods for different processors. Nodes were assigned actual cell areas. Solutions are constrained to be within 10 % of bisection.

Processor	1		2		3		4	
Circuit	SCAP	DBACP	SCAP	DBACP	SCAP	DBACP	SCAP	DBACP
N10	4.293(0.00031)	3.06(0.000112)	3.889(0.000121)	3.06(0.000102)	3.40056(0.0109)	3.06(0.000092)	3.2034(0.00103)	3.06(0.000088)
N15	5.23(0.000462)	4.36(0.000112)	5.128(0.000317)	4.36(0.000118)	4.9818(0.00278)	4.36(0.000101)	4.567(0.00019)	4.36(0.000091)
N20	7.39(0.001638)	5.204(0.00228)	6.16(0.001232)	5.204(0.001623)	6.009(0.001219)	5.204(0.00127)	5.89(0.001205)	5.204(0.00055)
N25	8.07(0.00441)	6.18(0.013371)	7.94(0.00334)	6.18(0.010234)	7.88(0.003112)	6.18(0.009232)	7.62(0.002112)	6.18(0.002306)
N30	9.29(0.009085)	7.75(0.0239083)	8.806(0.007902)	7.75(0.010915)	8.256(0.006805)	7.75(0.009649)	7.942(0.004964)	7.75(0.004818)

4. STATISTICAL ANALYSIS

The following null hypotheses have been investigated,

a) H0: there is no variation in the average min cut size, i.e. average min cut size of all algorithms is same, i.e. $\mu 1 = \mu 2$.

b) H1: there is no variation in the average runtime, i.e. average runtime of all algorithms is same, i.e. $\mu 11 = \mu 22$.

The single value ANOVA test was applied on tabulated values at 5% and 1% level of significance along with the p-values to confirm the results.

The statistical conclusions show that there is a wide variation in the quality of solution and average running time among all proposed algorithms except for circuits with 60 nodes. For the set of circuits with 60 nodes the mean of the average min cut values of the three algorithms is same.

5. CONCLUSION

Due to increase in the complexity of the digital electronic circuit, there is a massive requirement of CAD tools to automate the design process of VLSI systems. The field of CAD design tools focuses on shortening design time, improve product quality, and reduce product costs. Despite significant research efforts in this field, the CAD tools still lag behind the technological advances in fabrication. This calls for development of efficient heuristics for physical design automation. The main goal of this work was to introduce new approaches to tackle the circuit partitioning problem in circuit layout as a means to obtain near optimal solutions with reasonable running time. The solution to this optimization problem for the circuit layout focuses on finding a feasible solution which minimizes the net cut across partitions satisfying the balance constraints which further greatly affect the overall design of circuit reflecting circuit performance.

6. REFERENCES

[1] Caldwell,A. E., Kahng, A. B. and Markov, I. L. (2000), Improved Algorithms for Hypergraph Bipartitioning. *In Proceedings of the ASP-DAC 2000, Asia and South Pacific Design Automation Conference*, Yokohama, Japan ,pp. 661-666.

[2] Okumoto, K., Fukunaga, T., Nagamochi, H.(2010), Divide-and-Conquer Algorithms for Partitioning Hypergraphs and Submodular Systems. *Algorithmica*. Volume 62, pp. 787-806.

[3] Merz P, Freisleben B (2000) Fitness Landscapes, Memetic Algorithms, and Greedy Operators for Graph Bipartitioning, Journal of Evolutionary Computation, 8(1): 61–91.

[4] Bui TN, Moon BR (1996) Genetic Algorithm and Graph Partitioning. IEEE Transactions on Computers, 45(7): 841–855.
[5] S.Areibi, ―An Integrated Genetic Algorithm With Dynamic Hill Climbing for VLSI Circuit Partitioning‖, in Genetic and Evolutionary Computation Conference (GECCO-2000), IEEE, Las Vegas, Nevada, July 2000.
[6] Areibi, "Memetic Algorithms for VLSI Physical Design: Implementation Issues", Genetic and Evolutionary computation Conference, San Fransisco, California, pp140-145, July,2001.
[7] Maninder Kaur, Kawaljeet Singh, An Algorithm for Min Cut Partitioning for Digital Circuit Layout based on Evolutionary Approach IRACST – Engineering Science and Technology: An International Journal (ESTIJ), ISSN: 2250-3498, Vol.2, No. 4, August 2012
[8] Maninder Kaur, Kawaljeet Singh, On solving Mincut Balanced Circuit Partitioning Problem for Digital Circuit Layout using Evolutionary Approach with Solution Archive, International Journal of Computer Applications (0975 – 8887) Volume 62– No.14, January 2013
[9] Maninder Kaur, Kawaljeet Singh, An Intelligent Solution Archive Based Evolutionary Approach for Digital Circuit Layout International Journal of Computational Intelligence Research. ISSN 0974-1259 Vol. X, No.X (2013), pp. XXX–XXX .
[10] Maninder Kaur, Kawaljeet Singh, Soft Computing Approach for Digital Circuit Layout based on Graph Partitioning. International Journal of Computer Applications (0975 – 8887) Volume 15– No.1, February 2011
[11] Maninder Kaur, Kawaljeet Singh, Digital Circuit Layout based on Graph Partitioning Technique using DNA Computing, International Journal of Computer Applications (0975 – 8887) Volume 15– No.1, June 2013