

A MIMETIC ALGORITHM FOR REFINEMENT OF LOWER BOUND OF NUMBER OF TRACKS IN CHANNEL ROUTING PROBLEM

Debasri Saha¹, Rajat K. Pal² and Samar Sen Sarma³

¹²³*DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF CALCUTTA*

Abstract: Study of algorithms and its design can be progressed in various dimensions. In this paper, we have a definite refinement of lower bound on the number of tracks required to route a channel. The attack is from a complementary viewpoint. Our algorithm succeeds to avoid all kind of approximation. The approach performs exact mapping of the problem into graphical presentation and analyzes the graph taking help of mimetic algorithm, which uses combination of sequential and GA based vertex coloring. Performance of the algorithm depends on how effectively mimetic approach can be applied selecting appropriate values for the parameters to evaluate the graphical presentation of the problem. This viewpoint has immense contribution against sticking at local minima for this optimization problem. The finer result clearly exemplifies instances, which give better or at least the same lower bound in VLSI channel routing problem.

Key words: Manhattan Routing model, Channel routing problem, Constraint graphs, Maximum Independent set, Mimetic algorithm

1. INTRODUCTION

1.1 Channel routing problem

Channel routing problem(CRP) is NP hard in nature [10]. Extensive effort and attention has been attempted to tackle it. With the advancement of VLSI technology, as millions of gates have been accommodated in a tiny chip area, wiring the terminals of logic blocks altogether using minimum possible area has become a tedious task. If electrically equivalent pins are wired using rectangular routing region with terminals only on opposite sides, this strategy is termed as channel routing.

CRP is constrained form of optimization problem, where horizontal span of nets are assigned in horizontal tracks, avoiding conflicts so that track requirement is minimized. As CRP is NP hard [1,2,3], to design an algorithm with much lower complexity, we have taken heuristic support. As practical lower bound deviates much from the trivial one, our algorithm focuses on the computation of nontrivial lower bound on the number of tracks. The evolutionary techniques of mimetic algorithm, which efficiently handles hybrid optimization problems, are effectively incorporated here to find a better non-trivial solution. It generates near-optimal results for a number of well-known benchmark channels in reasonable time.

Here we consider grid based reserved layer Manhattan routing model, which is rectilinear in nature and each layer is restricted to accommodate a certain type (horizontal or vertical) of wire.

1.2 Constraints of CRP and their significance

Routing of wires should satisfy both kind of constraints- Horizontal constraints and Vertical constraints. Two nets n_i and n_j are said to have horizontal constraints, if their horizontal spans have at least one column common. Two nets n_i and n_j are said to have vertical constraints, if there exists a column such that the terminal on the top of the column belongs to net n_i and the terminal on the bottom of the column belongs to net n_j or vice versa.

These constraints can be well visualized by two constraint graphs – HCG(Horizontal Constraint Graph) and VCG(Vertical Constraint Graph)[4].

HCG $G=(V,E)$ is an undirected graph where each vertex $v_i \in V$ represents a net n_i and each edge $(v_i,v_j) \in E$ represents horizontal constraint between net n_i and net n_j . It signifies that if there is an edge between vertices v_i and v_j , then nets n_i and n_j cannot be placed in the same track.

Horizontal constraint can have a complementary representation through HNCG(Horizontal Non-Constraint Graph). HNCG $G=(V,E)$ is an undirected graph where each vertex $v_i \in V$ represents a net n_i and each edge $(v_i,v_j) \in E$ indicates that net n_i and net n_j are horizontal constraint-free i.e. horizontal span of net n_i and n_j have no common column. It implies that if there is an edge between vertices v_i and v_j , then net n_i and n_j can be placed in the same track if only horizontal constraint is taken into account.

VCG $G=(V,E)$ is a directed graph where each vertex $v_i \in V$ represents a net n_i and each directed edge $\langle v_i,v_j \rangle \in E$ represents vertical constraint between net n_i and net n_j such that there exists a column for which the top terminal belongs to net n_i and the bottom terminal belongs to net n_j . Interpretation of VCG is that if there is a directed edge from vertices v_i to v_j , then net n_i must be placed in a track above the track where net n_j is placed. That means it emphasize the ordering of net assignments in the channel.

The maximum number of nets, which crosses a column gives the knowledge of Channel Density (d_{max}). If we neglect vertical constraint, minimum number of track requirement is equal to d_{max} . This information is extracted either from HCG or from HNCG. In case of HCG, computation of Clique Number generates the value of channel density where as if HNCG is considered, we have to calculate Independence Number. Here we introduce the definition of clique number and independence number of a graph.

Definition 1 :Clique Number of a graph is the size of maximal complete sub-graph of the graph.

Definition 2: A set of vertices in a graph is said to be an independent set of vertices or simply independent set if no two vertices in the set are adjacent.

Definition 3: A maximal independent set is an independent set to which no other vertex can be added without destroying its independent property.

Definition 4: The number of vertices in the largest independent set of a graph is called the independence number.

On the other hand, VCG contributes the value of v_{\max} , which is nothing but the length of longest chain in VCG. It indicates that, if we consider only vertical constraints, at least v_{\max} number of track is required.

The rest of the paper is organized as follows. Section2 discusses the motivation of the work. Section3 discusses the proposed algorithm and section4 throws light on the time complexity of the algorithm. Section5 illustrates the execution of the algorithm by an example. Section6 focuses on the definite refinement on minimum number of tracks to route a channel and discusses the empirical observations on some randomly generated instances. Section7 extends our proposed algorithm for two-layer restricted doglegging model. Section8 concludes the paper & discusses scope for future work.

2. MOTIVATION AND CONTRIBUTION

Our work is motivated as we have analyzed a lots of practical instances of channel, which cannot be routed using either d_{\max} or v_{\max} number of tracks. Apparently $\max(d_{\max}, v_{\max})$ is formulated as an estimate of *trivial lower bound*. But simultaneous consideration of both the constraints generates a practical situation where a greater number of tracks are necessary to route a channel. It encourages us to combine the information from two constraint graphs into a single one, so that the resulted composite constraint graph can conjointly helps us to find the *non-trivial lower bound*.

HCG is an interval graph, whereas it's complement graph is a comparability graph [5]. The common feature of them is that they are both perfect in nature. A graph is said to be perfect, if it has no induced sub-graph with odd cycle of length greater than or equal to five.

But VCG can be any directed acyclic graph (if we take only cycle free VCG). If we proceed by extracting constraint based information from VCG and incorporating those into HNCG, it results into a modified HNCG, which may not still remain perfect in nature. Although Clique number or Independence number of perfect graph is polynomial-time computable, the possibility for modified HNCG of being non-perfect restricts us guaranteeing a deterministic polynomial time algorithm for independence number computation.

Success of mimetic algorithm in handling NP hard optimization problems inspired us to introduce it in our problem solving [6,9]. In our paper, mimetic algorithm tries to optimally color the vertices of the composite graph. The result is equivalent to finding maximal independent set of maximum cardinality.

In our previous paper[7], we deliberately kept composite constraint graph(Modified HCG) chordal as clique number of chordal graph is polynomially computable. But to do so, some vertical constraint based information is lost, which is treated as approximation. Hence there the Modified HCG reflects only approximated lower bound, not the exact one. Here we preserve all constraint related information in modified HNCG and this information is processed using GA operators to produce practical lower bound.

3. CONTRIBUTION OF THE ALGORITHM

3.1 Construction of composite graph

We propose a hybrid GA based heuristic algorithm to determine the non-trivial lower bound on the number of tracks required to route a channel in polynomial time. An edge between vertices v_i and v_j in HNCG signifies that, net n_i and n_j have no horizontal overlapping. That doesn't mean those can be placed in the same track, as vertical constraint may impose ordering on their tracks. In VCG, if directed edges $\langle i, j \rangle$ and $\langle j, k \rangle$ are present, that indicates net n_i has to be placed above net n_j and net n_j above net n_k . Hence net n_i has to be placed above net n_k . This transitive closure property is strictly followed by vertical constraints. So net n_i cannot be accommodated with net n_k , even if those are horizontal constraint free. It is focused that none of the constraint graph can alone cover all constraint information. So we extract this vertical constraint based information from VCG and incorporate those into HNCG to highlight all constraint information through a single graph.

We find out all possible directed paths between each pair of source (indegree zero) and sink (Outdegree zero) vertices in VCG, then apply transitive closure property (if $a \rightarrow b$ and $b \rightarrow c$, then $a \rightarrow c$) to construct an edge list E , which contains edges between all pair of vertices having a directed path between them in VCG, but without any directed edge between them. The directed edges already present in VCG reflects direct vertical constraint, hence those are automatically covered by horizontal constraint consideration. Hence E contains only those edges, which reflect indirect or derived vertical constraints.

Each edge (v_i, v_j) of edge list E , if present in HNCG, indicates that the corresponding nets n_i and n_j are not horizontally constrained but only vertically. Our strategy is to delete all such edges from HNCG. The Modified HNCG, thus obtained, is termed as Composite graph as it focuses combined effect of all constraints.

Definition 5: For composite graph $G = (V, E)$, each vertex $v_i \in V$ represents a net n_i and each edge $(v_i, v_j) \in E$ implies corresponding nets n_i and n_j are constraint-free and can be placed in same track.

Conversely we can say, two disconnected vertices v_i and v_j reflects the fact that corresponding nets n_i and n_j are mutually constrained, hence occupy separate tracks. Independence number I , i.e. the maximum number of mutually unconnected vertices of the Composite graph gives an estimate of lower bound (Lbound) of tracks.

3.2 Computation of Independence Number using Mimetic Algorithm

The problem of finding Independence number I of Composite graph is mapped into the problem of proper coloring of vertices, where connected vertices are colored with distinct colors. Our algorithm proceeds with proper coloring of Composite graph satisfying the objective that as many vertices as possible are colored by each color applied. That means, if each color is assigned to as many vertices as possible obeying proper coloring, the maximum number of vertices colored with identical color specifies Independence Number I .

In this context, the order of sequential coloring of vertices is of great significance. The vertices of composite graph are arranged in increasing order of

their degree and considered for proper coloring in this sequence. We stack for use as many colors as the number of vertices in Composite graph. Each color is encoded as an integer. GA works by evolving a population of strings over generations. We use random selection of a color, consider vertices in minimum degree sequence, continue assigning the color till the violation of proper coloring, followed by selection of another color. Fitness value of a string is evaluated as the maximum occurrences of a single color (integer) in the string. GA attempts to optimize this fitness function through effective application of GA parameters Reproduction, Crossover and Mutation [8] with appropriate probability. Reproduction emphasizes survival of highly fit strings. Crossover provides encouraging results against sticking to local optima. Random selection of mutation location also helps to reach global minima.

3.3 Detection of obstruction condition

Let us consider the following two channel specifications:

TOP: 3 1 2 0 2 0
BOTTOM: 0 3 0 1 4 4

TOP: 1 1 4 0 2 0
BOTTOM: 0 3 0 3 4 2

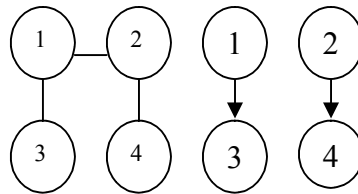
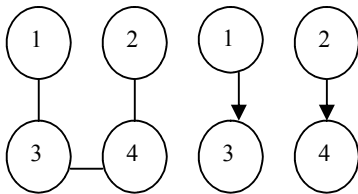


Fig1 HNCG

Fig2 VCG

Fig3 HNCG

Fig4 VCG

In both cases, Composite graph is same as HNCG. Independence number I is 2 but track requirement is 3, as net 1 and net 2 can't be placed in same track for fig1 and net 3 and net 4 can't be placed in same track for fig3. So at least three tracks are required to route the channel.

Lemma1: For a pair of directed paths (chain) from source to sink vertices, with length difference ≤ 1 and at least one with length v_{max} , then if source vertices, or sink vertices, or both pairs are disconnected in HNCG, at least one extra track is the essentially required.

Our proposed algorithm searches for the presence of obstruction condition, if found, at least one extra track is needed. Hence minimum increment in number of track requirement, INCR is 1.

3.4 Algorithms

Algorithm MIMETIC_LBOUND

Input: Channel specification

Output: Lbound, Non-trivial lower bound on the number of tracks.

Step1: Construct HNCG and VCG from channel specification.

Step2: Using the transitive closure property, compute the list of edges, E between all possible pair of vertices having shortest directed path length ≥ 2 between them in VCG.

Step 3: If E is empty, consider HNCG as Composite graph (Modified HNCG).

Go to Step 5.

- Step 4: Delete each edge e of the list E from HNCG, if present in HNCG.
Finally resulted graph is denoted as Composite graph (Modified HNCG).
- Step 5: If the Composite graph is a null graph (having only isolated vertices), then
Lbound = No of vertices in Composite graph,
Else compute independence number, I , of the Composite graph using mimetic algorithm.
- Step 6: Check for the presence of *obstruction* condition
If present, compute increment in lower bound, INCR due to that,
Else INCR = 0.
Finally, Lbound = $I + INCR$.

Following are the steps of mimetic algorithm to compute the independence number of a graph

Mimetic Algorithm I_number

- Input: Composite graph, size of initial population, No of iteration n , Crossover probability $pcross$, Mutation probability $pmutate$.
- Output: I , Independence number of Composite graph.
- Step1: Generate initial population containing valid and unique strings of colors using sequential vertex coloring.
- Step2: Compute maximum fitness value, $max_fitness$, of strings in current population.
- Repeat up to step6 for n times
- Step3: Select strings of high fitness value to generate mating pool.
(Reproduction)
- Repeat step4 for $ncross*$ times
- Step4: Select parents and crossover site; Perform crossover.
Check validity of new strings; if valid, replace previous one by it.
- Repeat step5 for $nmutate*$ times
- Step5: Select string for mutation, site and replacing color; perform mutation.
Check validity of new strings; if valid, replace previous one by it.
- Step6: Compute maximum fitness value, $new_max_fitness$, of the new generation population.
If $new_max_fitness > max_fitness$, $max_fitness \leftarrow new_max_fitness$;
Replace current population with new generation population
- Step7: $I \leftarrow max_fitness$
*(Compute $ncross$ (number of crossover) from $pcross$ and $nmutate$ (number of mutation) from $pmutate$.)

4. COMPLEXITY ANALYSIS OF MIMETIC_LBOUND

Complexity calculation in Mimetic algorithm based design is not straightforward. This paper emphasizes on finding a better non-trivial lower bound than our earlier deterministic algorithm [7]. Let us try to give some highlights of time complexity of our algorithm, Sequential vertex coloring requires $O(n^2)$, where n is the number of nets. The initial population of Genetic algorithm is thus obtained in $O(n^2)$ time complexity.

For mimetic algorithm based heuristic search, we know that it is suitable for MIMD parallel computing and distributed computing environment as these are composed by network of workstations. However we have seen that CPU time required for executing our algorithm using single Pentium4 processor is reasonable for all practical purposes.

5. ILLUSTRATION WITH EXAMPLE

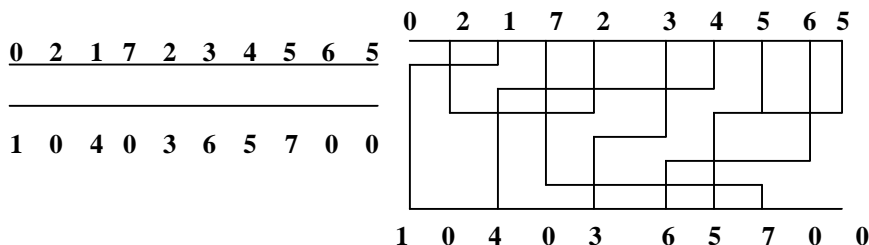


Fig5: Channel Instance and its routing

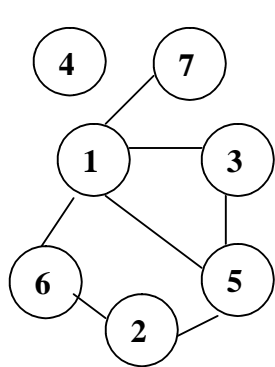


Fig6: HNCG

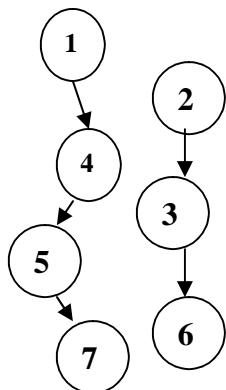


Fig7: VCG

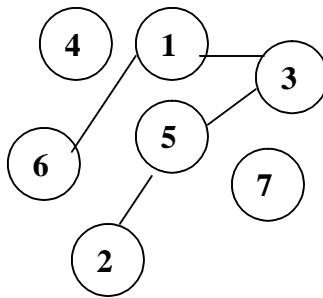


Fig8: Composite graph

Using transitive closure property, the final edge list E is constructed. $E = \{(1,5), (1,7), (4,7), (2,6)\}$ Edges in these list indicates derived or indirect vertical constraints between corresponding nets. Among these edges, $(1,5)$, $(1,7)$ and $(2,6)$ are present in HNCG and those have to be eliminated from HNCG. Deletion of those edges generates Composite graph. The Maximum Independent Set is $\{2,3,4,6,7\}$. Independence number $I=5$. Analyzing VCG, it is revealed that there are 2 directed paths from source to sink vertices with lengths $4(v_{max})$ and 3. Those are $(1 \rightarrow 4 \rightarrow 5 \rightarrow 7)$ and $(2 \rightarrow 3 \rightarrow 6)$. The source vertices 1 and 2 are horizontally constrained and the edge $(1,2)$ is absent in HNCG. Thus *obstruction* condition is satisfied for this channel instance. So $INCR=1$. Hence minimum number of track requirement by our algorithm $MIMETIC_LBOUND$ is $5+1$ or 6. Practical solution shows that, the minimum number of tracks requirement is 6, Hence the result obtained by $MIMETIC_LBOUND$ tallies with practical solution.

6. REFINEMENT OF LOWER BOUND OF NUMBER OF TRACK REQUIREMENT THROUGH OUR ALGORITHM

Theorem: MIMETIC_LBOUND computes exact lower bound on the number of track requirement to route a channel without any approximation, and result is better or at least equal to that found in LOWER_BOUND algorithm.

We demonstrate the refinement in results achieved by MIMETIC_LBOUND in comparison to other algorithms in tabular form.

Table 1: Lower bound using MIMETIC_LBOUND and comparison with other algorithms

| Channel Instance | d_{\max} | v_{\max} | $\text{Max}(d_{\max}, v_{\max})$ | Lbound by our algo | CPU time | Best solution known |
|------------------|------------|------------|----------------------------------|--------------------|----------------|---------------------|
| CH1 | 4 | 4 | 4 | 6 | .002s | 5 |
| CH2 | 3 | 5 | 5 | 6 | .0023s | 5 |
| CH3 | 4 | 4 | 4 | 6 | .0025s | 5 |
| CH4 | 4 | 4 | 4 | 6 | .0024s | 5 |
| CH5 | 5 | 5 | 5 | 7 | .0034s | 6 |
| RKPC1 | 3 | 3 | 3 | 4 | .002s | 4 |
| RKPC6 | 4 | 5 | 5 | 7 | .11s | 7 |
| RKPC8 | 5 | 5 | 5 | 7 | .06s | 7 |
| RKPC9 | 6 | 6 | 6 | 10 | .16s | 10 |
| DDE | 19 | 23 | 23 | 28 | 1min 54.16s | 28 |

The result is achieved implementing MIMETIC_LBOUND in matlab using Pentium4 machine with clock frequency 1.5 GHz. CH1 through CH5 [mentioned in appendix], clearly demonstrate refinement in results. For next four channel instances [10], MIMETIC_LBOUND results tally with previous results. MIMETIC_LBOUND also provide result as good as other conventional algorithm for Deutsch's difficult example (DDE).

Table 2: Suitable values of GA parameters to obtain optimum solution for some channel instances using MIMETIC_LBOUND

| Channel instance | GA related parameters for optimum Lbound | | | |
|------------------|------------------------------------------|-----------------|-----------------------|----------------------|
| | Initial Population | No of iteration | Crossover Probability | Mutation Probability |
| CH1 | 10 | 2 | .4 | .001 |
| CH2 | 10 | 2 | .4 | .001 |
| CH3 | 10 | 2 | .4 | .001 |
| CH4 | 10 | 2 | .4 | .001 |
| CH5 | 20 | 6 | .8 | .001 |
| RKPC1 | 16 | 6 | .8 | .001 |
| RKPC6 | 14 | 4 | .6 | .001 |
| RKPC8 | 12 | 2 | .4 | .001 |
| RKPC9 | 30 | 4 | .8 | .001 |
| DDE | 140 | 12 | .8 | .001 |

Regarding track minimization problem of CRP, our proposed algorithm is able to overcome approximation included in our previous paper and ascertains better results for a number of channel instances. With the help of algorithm LOWER_BOUND of previous paper [7] minimum number of tracks required to route the channel (described in section 5) is 5. This result varies from practical solution as it is an approximated result. But MIMETIC_LBOUND concludes nontrivial lower bound is 6, which tallies with the practical solution. As our

approach preserves all constraint-based information in composite graph, it enhances the accuracy in result.

7. TWO LAYER RESTRICTED DOGLEG ROUTING

For channels with multi-terminal nets, restricted doglegging often remove cycles from VCG and can route such channels. It sometimes produces better routing solution. Our algorithm can invariantly be applied for multi-terminal nets, if horizontal wire segment of such net is splitted into set of two terminal subnets and HCG (or HNCG) and VCG are constructed as follows.

For both HCG $G_H'=(V',E_1')$ and VCG $G_V'=(V', E_2')$, V' is the set of vertices corresponding to two terminal subnets of nets. If e_{i_1} and e_{i_2} are two subnets of net n_i and n_j respectively, then $(e_{i_1}, e_{i_2}) \in E_1'$ when e_{i_1} and e_{i_2} overlaps. HNCG G_{HN}' is obtained by complementing the edges of G_H' . For constructing edges of VCG, if net n_i and n_j both cross through some column c , where l_i and r_i are subnets of net n_i and l_j and r_j are subnets of net n_j to the right and left of column c , then directed edges $\langle l_i, l_j \rangle$, $\langle l_i, r_j \rangle$, $\langle r_i, l_j \rangle$ and $\langle r_i, r_j \rangle$ have to be introduced in VCG. Construction of HNCG and VCG and hence lower bound on number of tracks for channels with multi-terminal nets can be demonstrated by an example.

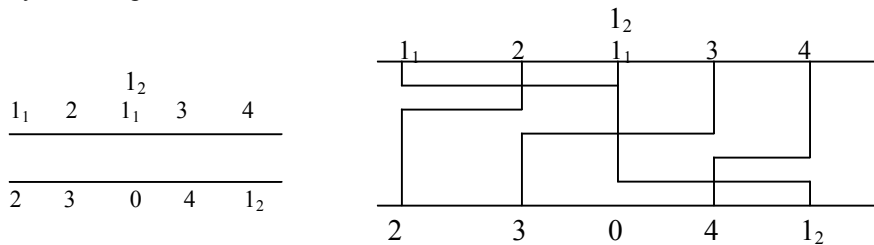


Fig9: Channel instance and its routing

VCG of this channel forms a cycle, so doglegging is applied. 1_1 and 1_2 are two subnets of net 1.

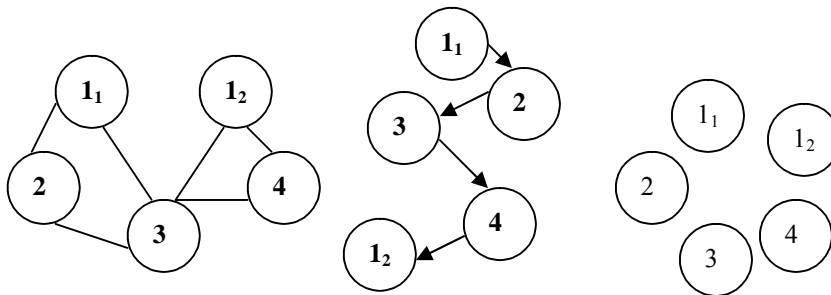


Fig10: HNCG G_{HN}'

Fig11: VCG G_V'

Fig12: Composite Graph

Composite graph is null graph. Hence Independence Number $I=5$ and lower bound on number of tracks requirement is also 5, which tallies with the practical solution for routing the nets, as shown.

8. CONCLUSION

Heuristic algorithm in general, outperforms Approximation algorithm. In this paper, we tried to solve non-trivial lower bound in the restricted two layer (VH) channel routing problem. The algorithm presented is non-deterministic in nature and specifically NP hard. The deterministic version that was presented in the paper [7], approximates the problem and solved it deterministically in $O(n^4)$ time. However the solution is ratio-bound to lower bound solution. We have taken here the exact problem and solved the problem by a mimetic algorithm that gives near-optimal solution. The result is encouraging as it shows a better lower bound on number of tracks in many instances.

The extension of the work in multi-layer environment is our next projected extension of the work.

APPENDIX

| | | | | | | | | | | | | | | | |
|------|----------|---|---|---|----|---|---|---|---|---|----|----|---|----|---|
| CH1: | TOP: | 0 | 2 | 1 | 7 | 2 | 3 | 4 | 5 | 6 | 5 | | | | |
| | BOTTOM:1 | 0 | 4 | 0 | 3 | 6 | 5 | 7 | 0 | 0 | | | | | |
| CH2: | TOP: | 9 | 8 | 7 | 5 | 6 | 1 | 0 | 0 | 2 | 4 | 3 | | | |
| | BOTTOM:0 | 0 | 0 | 9 | 8 | 7 | 6 | 2 | 1 | 4 | 3 | 5 | | | |
| CH3: | TOP: | 0 | 2 | 1 | 9 | 2 | 4 | 0 | 3 | 5 | 6 | 7 | 0 | | |
| | BOTTOM:1 | 0 | 3 | 0 | 4 | 6 | 9 | 5 | 7 | 8 | 0 | 8 | | | |
| CH4: | TOP: | 0 | 4 | 1 | 4 | 2 | 0 | 3 | 5 | 9 | 6 | 7 | 0 | | |
| | BOTTOM:1 | 0 | 3 | 2 | 6 | 9 | 5 | 7 | 0 | 8 | 0 | 8 | | | |
| CH5: | TOP: | 0 | 4 | 1 | 11 | 4 | 2 | 3 | 5 | 6 | 7 | 8 | 0 | 9 | 0 |
| | BOTTOM:1 | 0 | 3 | 0 | 2 | 6 | 5 | 7 | 8 | 9 | 10 | 11 | 0 | 10 | |

REFERENCES

- [1] A.S. LaPaugh, Algorithm for integrated circuit layout: An analytic approach, Ph.D.dissertation,MIT Laboratory for Computer Science, Nov. 1980.
- [2] R.K Pal, S.P. Pal, A. Pal, On the Computational complexity of multiplayer channel routing, Technical Report: TR/IIT/CSE/92/02, Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur 721 302, India,1992
- [3] R.K Pal, S.P. Pal, A.K. Datta, A. Pal, NP- completeness of multi-layer no-dogleg channel routing and an efficient heuristic, Proc. 6th Int. Conf. On VLSI Design, 1993, pp.80-83.
- [4] G.A. Scaper, Multi-layer channel routing, Ph.D. dissertation, Computer Science Department, University of Central Florida, Orlando, Fla., Aug. 1989.
- [5] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980
- [6] Ricardo Blanco-vega, Jose Hernandez Orallo, Analyzing the Trade-off between comprehensibility & accuracy in Mimetic Models, Dept of System Informatics & Computation
- [7] Integration, the VLSI journal 25(1998) pp. 71-84.
- [8] Optimization for Engineering Design, Kalyanmoy Deb.
- [9] Pinaki Mazumder, Elizabeth M. Rudnick, Genetic Algorithms for VLSI Design, Layout & Test automation.
- [10] R.K. Pal, Multi-layer Channel Routing. Narosa Publishing House, India.