A Genetic Algorithm For Over–The–Cell and Channel Area Optimization

B. M. Goni, T. Arslan

Department of Electronics and Electrical Engineering,
The University of Edinburgh,
The King's Buildings,
Edinburgh, EH9 3JL,
United Kingdom.
Baba.Goni@ee.ed.ac.uk

B. Turton

Department of Electrical, Electronic & Systems Engineering,
Cardiff University,
Cardiff, CF2 3TF,
United Kingdom.
Turton@cardiff.ac.uk

Abstract–The paper presents a new genetic algorithm (GA) based router which concurrently optimizes both areas over–the–cell and main channel. The GA is guided through the solution space by a set of heuristics/rules which are executed during the evaluation of prospective solutions. The GA commences by identifying multi–terminal net circuit segments (including side nets) for over–the–cell routing, and places these on tracks over–the–cell according to horizontal constraints. The algorithm proceeds to route the remaining net circuits inside the main channel area subject to horizontal and vertical constraints. The over–the–cell routing procedure within the GA ensures that only two terminal nets are left in the main channel after its action. This reduces the amount of congestion in the main channel area. The algorithm is tested on several internationally well known benchmark examples, and found to produce significantly better routing solutions than previous approaches to the problem to date.

1 Introduction

The number of devices in a chip have increased from about a thousand devices in the early 70's to over tens of million devices now [1]. This has led to a significant increase in the number of interconnections. An efficient routing of these interconnections reduces chip size, and improves the reliability of the chip design. In light of this, routing has received a great deal of attention from researchers [2], [3], [4].

In the physical design process for very large scale integrated (VLSI) circuits the logical structure of a circuit is transformed into its physical layout. Detailed (channel) routing is one of the tasks in this process. A detailed router connects pins of signal nets (wires) in a rectangular region in accordance with a set of routing constraints, such as the number of layers, the minimal space between wires, minimum wire width, number of vias, crosstalk and the net length. The results of this detailed routing has a strong influence on the fabrication yield and production costs of the circuit [1].

The channel routing problem has been found to be NP–complete [5] implying there is no known deterministic algorithm to solve it in polynomial time. Several algorithms have been proposed, for example [2]–[5], to solve this problem. These traditional channel routing algorithms have proved successful in reducing channel height. Any further reductions are only possible if some nets are routed "outside" the channel [6]. Based on this observation, some researchers have proposed utilization of over–the–cell (OTC) area to obtain further reduction in the channel height [3], [6]–[9]. As more metal layers are becoming available for routing in the standard cell design style, routing OTC becomes both practical and important.

The OTC channel routing problem is a generalization of the channel routing problem. It therefore follows that the OTC channel routing problem is also NP–hard [1], [5]. Due to this fact research in this problem has focused on the development of heuristic based techniques [3], [8]. The fundamental idea behind existing OTC routers is to select a subset of nets which are suitable for OTC routing. This is due to the fact that not all nets can be routed OTC. Net segments at the top or bottom of the channel can be routed OTC. A net is called top (bottom) net if all of its terminals are on the top (bottom row). Other nets have segments which can only be partially routed OTC while their remaining segments are routed within the channel. Previous routers select only top or bottom net segments for OTC routing. However, the OTC router developed by the authors in [6] can route both types of nets, by considering both non–bottom/top nets and bottom/top nets for routing OTC. All the OTC routers above route only some multi–terminal net circuit segnments OTC. The routing of all multi–terminal net circuit segments (including side nets) OTC will greatly reduce congestion in the main channel area. The algorithm developed in this paper specifically tackles this type of routing.

Conventional channel routers, including those utilizing OTC routing, use algorithms which are driven by rules/heuristics which perform routing actions based on identification of certain nets. Such techniques are ad-hoc and do not involve systematic search of the complex search space imposed by the routing problem (the complexity increases when over the cell solutions are included in the search). The search space includes a number of solutions, to a given routing problem, which vary in their quality of satisfying the routing criteria. Identification of quality solutions requires effective exploration of the search space, a task which could be carried out by an effective search algorithm.

To our knowledge, non of the combined OTC routers in the literature utilize a systematic search algorithm in the exploration of the routing search space.

This paper describes a new router which utilizes a class of search techniques called Genetic Algorithms (GAs) in exploring the search space formed by both OTC and channel regions. We show that our router produces solutions which significantly improves on current OTC routers in the literature. To our knowledge this is the first GA based router that carries out OTC routing in addition to channel routing.

GAs [14] are a class of stochastic search techniques which have shown significant success in the solution of problems with complex search spaces. These are characterized by multiple solutions, discontinuities [10]–[12], and existence of local minima/maxima, which characterize VLSI design problems [13], [14], specifically channel routing [15]–[21].

The GA based router concurrently optimizes both areas OTC and main channel. The GA is guided through the solution space by a set of heuristics/rules which are executed during the evaluation of prospective solutions. The GA commences by identifying multi–terminal net circuit segments (including side nets) for OTC routing, and places these on tracks OTC according to horizontal constraints. The algorithm proceeds to route the remaining net circuits inside the main channel area subject to horizontal and vertical constraints. The OTC routing procedure within the GA ensures that only two terminal nets are left in the main channel after its action. This reduces the amount of congestion in the main channel area. The algorithm is tested on several internationally well known benchmark examples, and found to produce significantly better routing solutions than previous approaches to the problem to date.

Section 2 gives a brief description of the routing problem. Section 3 presents concepts and definitions on genetic algorithms. The algorithm description is given in Section 4. Experimental results showing the effectiveness of the proposed approach are presented in Section 5. Conclusions are given in Section 6.

2 The Routing Problem Definition

A channel routing problem is traditionally presented by specifying a netlist such as the Yoshimura–Kuh (YK) channel netlist shown in Figure 1 [2]. In some problems, a netlist may contain *side nets*, which are specified in the *left list* and *right list*, corresponding to the nets that exit on the left and right edges, respectively of the channel [25]. The realization of the netlist of Figure 1 is shown in Figure 2 [2], with two rows of terminals along its top and bottom sides. Each terminal is assigned a number from the interval [0, N]. Terminals with the same positive number i have to be connected and form *net i*. Unconnected terminals are designated by zero. Two wiring layers are available, one for horizontal segments and the other for vertical segments. The horizontal segment of each net must be placed on a horizontal track [15]. Wire transitions between one layer to the next are made through vias (contact holes).

The channel router is required to assign each net to a horizontal track subject to vertical constraints and horizontal constraints, in addition to minimizing such horizontal tracks [15]. Also, in each of the channel regions (OTC or inside the channel), the horizontal segment of each net must be placed on a single horizontal track, and may not be segmented over several tracks.

In the over–the–cell channel areas, there are only horizontal constraints and vertical constraints do not exist. We assume that vias are allowed in the over–the–cell areas.

A vertical constraint or violation (VV) [15] is said to exist between two distinct nets if the terminal number of the top connected net is the same with that of the bottom connected net. In such a case the former net is placed above the latter net. Horizontal constraints or violations (HVs) [15] occur between two different nets if the sets of terminal numbers spanned by the two nets intersect. For example if we consider the sets of terminal numbers spanned by nets 1 and 2 in Figure 1, we see that the span of net 1 (2–5) is contained in 2. The two nets intersect and therefore cannot be routed on the same horizontal track.

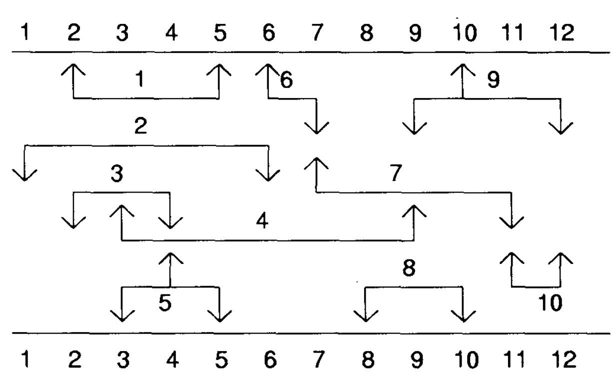

Figure 1: Yoshimura–Kuh Benchmark Channel netlist.

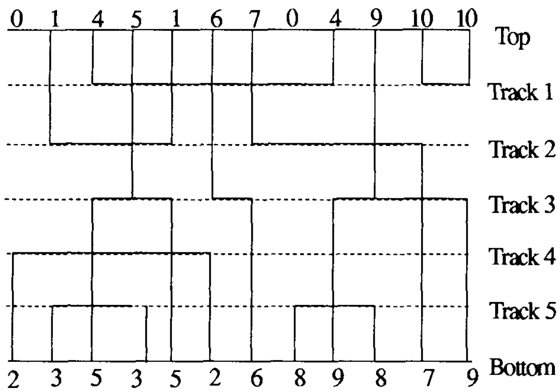

Figure 2: A realization of the netlist in Figure 1.

3 Genetic Algorithms

A genetic algorithm is a search algorithm modelled on the mechanics of natural selection and genetics. Domain knowledge is embedded in the abstract representation of a candidate solution termed an individual. Individuals are grouped into sets called populations. A genetic algorithm, like any other search technique, requires some criterion for the evaluation of the solutions. This criterion, called the fitness function, estimates the quality of the individual. Genetic operators, such as crossover and mutation, are applied to solutions chosen from the population by means of the selection operator. The best parents and the offspring make up the population in subsequent iterations of the algorithm, called generations [21]. There exist many variants upon the basic GA mechanism as defined in [12], [24], but a more or less standard technique is that of Goldberg [10], and can be stated in pseudocode as in Figure 3.

Begin

Generate an initial population of binary/integer coded solutions
Derive a fitness measure for each solution
Repeat until convergence
{
Repeat to create new population
{
Select two parents according to their fitness
Mate parents in order to create two offspring
Mutation
Use offspring to create a new population
}
Derive a fitness measure for each solution
}

End

Figure 3: Genetic Algorithm pseudocode

Before applying a GA to the solution of a particular problem, it has to be determined how the solution is to be encoded as a chromosome, and how that chromosome will subsequently be translated into an appropriate measure of its relative fitness. We shall consider two encoding schemes here; those with binary valued, and those with real valued chromosomes.

A binary chromosome consists of a string of binary digits, as described in [12]. A 1 or 0 in the string may, in a Boolean scheme, correspond to whether some condition is true or false, as in Figure 4. Alternatively, a chromosome may be formed by a string of integer variables, as in Figure 5.

1	0	0	0	1	0	0	1

Figure 4: Each BIT in a binary chromosome may represent some Boolean condition.

7	3	4	9	1	4	8	2

Figure 5: A chromosome may contain integer variables.

In traditional GAs, selection of parents is carried out by roulette wheel parent selection [23]. The aim of parent selection in a genetic algorithm is to give more reproductive chances, on the whole, to those population members that are the most fit. This returns a parent which has been randomly selected with a probability proportional to it's fitness. Therefore,

Prob(individual being selected) = fitness of individual/total population fitness

This algorithm is referred to as roulette wheel selection because it can be viewed as allocating pie–shaped slices on a roulette wheel to population members, with each slice proportional to the member's fitness. Selection of a population member to be a parent can be viewed as a spin of the wheel, with the winning population member being the one in whose slice the roulette spinner ends up [23]. The population member with the greatest fitness is likely to be selected most of the time.

In a conventional genetic algorithm, crossover recombines the genetic material in two parent chromosomes to make two children. John Holland [12] experimented with a crossover operator called *one–point crossover.* This occurs when parts of two parent chromosomes are swapped after a randomly selected point, creating two children. Figure 6 shows one point crossover with binary chromosomes. Variations to one point crossover include two point and N point crossover, with each having certain advantages in solving particular problems.

CROSSOVER POINT

Parent 1

1	1	0	1	0	1	0	1

Parent 2

0	1	1	1	0	0	1	1

Child 1

0	1	1	1	0	1	0	1

Child 2

1	1	0	1	0	0	1	1

Figure 6: Crossover mechanism for binary chromosomes

Mutation is the random change of an allele from one alphabet value to another. For a problem over the binary alphabet, the original allele is exchanged for its complement. The mutation operator offers the opportunity for new genetic material to be introduced into the population. Mutation takes place after crossover a small percentage of the time (mutation rate).

Elitism is a technique by which the fittest individual in the population always gets put through to the next generation in order to overcome the problems caused when the fit individuals are not selected to produce children.

4 Algorithm Description

The algorithm for this work is implemented within the framework of GENESIS (GENEtic Search Implementation System) [23], which is a generic tool for the implementation of genetic systems. Our investigations revealed that with a population of 100 chromosomes, the following GA characteristics are necessary for a performance which provides a good trade off in speed and quality of solutions.

1. Two point crossover [23] and random mutation at rates of 0.6 and 0.001.
2. Chromosome selection procedure using Baker's Stochastic Universal Sampling algorithm [22], [23].
3. Employment of an elitist selection strategy [23].

The chromosome structure for a single net in a routing problem is presented as follows;
[TOP, INT, BOT, ID]

where,

TOP = top OTC track number for the net
INT = internal track number for the net
BOT = bottom OTC track number for the net
ID = net number

For a complete routing structure with several nets the chromosome structure will be a cascade of the structure above. A key component of our implementation is the fitness evaluation stage, where the quality of the different routing solutions in the GA solution space are assessed and rewarded. Given a chromosome structure from the GA solution space (see Figure 7), the fitness function starts by identifying the number of multi–terminal net circuits (including side nets) routed OTC. Further, it proceeds to identify the number of two terminal net circuits routed inside the channel. These are combined to provide a single fitness measure. This process is repeated for each chromosome in the GA population. For a given structure Y, the fitness function F(Y) (which we seek to maximize) can be specified as:

$$F(Y) = OTC(Y) + IR(Y).$$

Where OTC and IR are routing procedures and are described as follows:

A. *Over–the–cell routing (OTC)*: Scans the chromosome for multi–terminal net circuits (OTC nets) routed in top or bottom (or both) OTC areas without HVs, and returns their number. Here, *side nets* are treated in the same way as multi–terminal nets. In the chromosome, side nets with terminals routed in OTC areas without HVs are identified and their numbers returned as well.

B. *Internal routing (IR)*: Performs the scanning of the chromosome for two pin (terminal) net circuits (IR nets) connected to the top and bottom pins inside the main channel, without HVs and VVs and returns their number.

The GA is set to interpret structures with the highest fitness total (summation of values returned by the procedures above) as the best. If both OTC and IR nets are absent in the structure sent by the GA for evaluation, then a fitness value of zero is returned. Such a structure will not proceed to the next generation and hence will not survive (see Figure 7). Therefore, the higher the fitness value associated with a given routing structure the better. There exist other fitness functions for channel routing [15–21], however, the framework defined in our fitness function combines OTC routing and computational simplicity, inspite of carrying out routing both inside the channel and OTC, in achieving better track savings than these functions. This will be shown in the results and discussion section.

Two or more net circuits are placed on the same track if they are without horizontal and vertical violations in the main channel area. For the OTC areas only the elimination of horizontal violations need to be satisfied before net circuits are routed on the same track. This is illustrated using the example of Yoshimura-Kuh channel circuit [2] in Figure 2. The algorithm commences by first identifying and rewarding multi-terminal net circuits that are routed in top or bottom (or both) OTC areas without HVs (nets 1,2,3,4,5,8,9 and 10) as shown in Figure 8. It then identifies and rewards those net circuit segments without HVs and VVs that remain fully or partially within the channel area, (nets 5,6,7 and 9). During this process the algorithm identifies those nets that can be placed on the same track (nets 5,6, and 9) resulting in track savings. It is evident that our algorithm requires only two rows/tracks (see Figure 8) to route this channel circuit, compared to five in [2]. The four net circuits left in the channel are all two terminal nets. Using the chromosome structure mentioned above, the realization of Figure 8 is represented as below:

[1,0,0,1,0,0,1,2,0,0,3,3,2,0,0,4,0,2,2,5,0,2,0,6,0,1,0,7,0,0,2, 8,0,2,1,9,1,0,0,10]

Vertical and horizontal segments of nets are placed on the first and second layers respectively in the channel area (VH model).

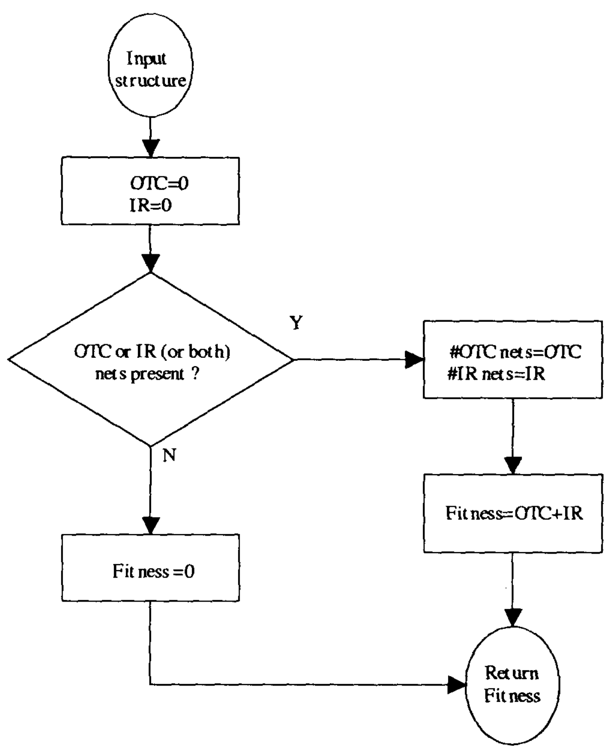

Figure 7: Fitness function flowchart

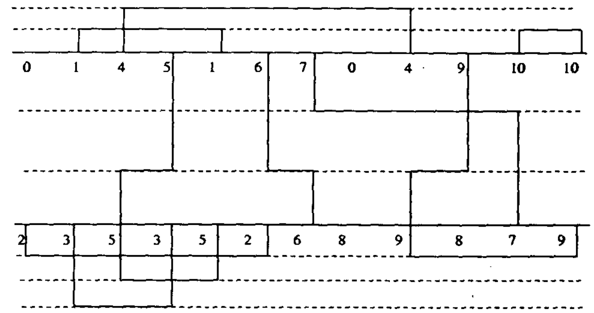

Figure 8: Realization of the netlist of Figure 2 using our algorithm

5 Results and Discussion

Table 1 shows the results (in terms of number of tracks) obtained using our algorithm, and compares this with the best GA based internal channel router [21], and other well known OTC routing algorithms [6], [7], [8]. From the table, our algorithm obtains solutions with significant track savings inside the channel. For example with the benchmark example Ex3a in Table 1 our algorithm obtains a solution using 8 tracks, compared with 12 tracks obtained by the best of the previous algorithms [6], [7]. Since the algorithm in [8] provides the closest match to some of our results in the main channel area, Table 2 compares our algorithm with that in [8] for all areas i.e Top, Bottom and Internal. Overall our algorithm performs better. For example, with Ex3c our algorithm obtains solutions of 12 and 9 tracks for the top and bottom OTC areas respectively, compared to 15 and 16 tracks achieved for the same areas in [8]. Inside the channel our algorithm obtains a solution using 11 tracks compared with 15. Also, with the Ex1 benchmark our algorithm obtains solutions of 4 and 6 tracks for the top and bottom OTC areas respectively, compared to 11 and 7 tracks achieved for the same areas in [8]. Within the main channel area our algorithm obtains a solution using 5 tracks compared with 6. The layout implementation for the Ex1 benchmark obtained using our algorithm is shown in Figure 9.

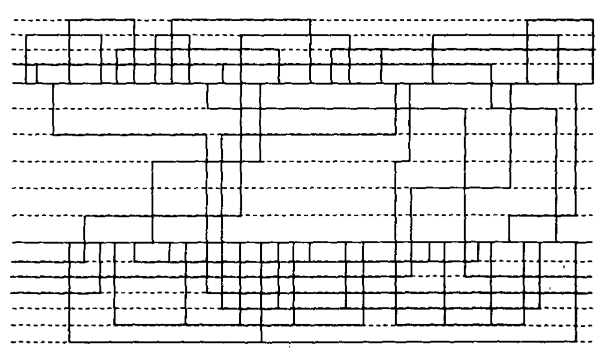

Figure 9: Layout implementation for the Ex1 Benchmark obtained using our algorithm.

Example	*Original Density*	*[21]*	*[6]*	*[7]*	*[8]*	*Our Algorithm*
Ex1	12	12	9	9	6	5
Ex3a	15	15	12	12	15	8
Ex3c	18	18	13	14	15	11
Ex4b	17	17	13	13	6	6
Ex5	20	20	11	11	9	9
De	19	19	16	16	13	13

Table 1: Comparison (in terms of number of tracks) of our algorithm with GA internal router, and some well-known algorithms for OTC channel routing.

		[8]			*Our Algorithm*		
Example	*Density*	*Top*	*Int.*	*Bot.*	*Top.*	*Int.*	*Bot.*
Ex1	12	11	6	7	4	5	6
Ex3a	15	13	15	10	9	8	6
Ex3c	18	15	15	16	12	11	9
Ex4b	17	28	6	24	8	6	11
Ex5	20	24	9	18	10	9	7
De	19	20	13	19	14	13	14

Table 2: Comparison (in terms of number of tracks) of our algorithm with a well-known algorithm for all areas.

6 Conclusions

We have presented a new genetic algorithm (GA) based router which concurrently optimizes both areas OTC and main channel. The GA is guided through the solution space by a set of heuristics/rules which are executed during the evaluation of prospective solutions. The GA commences by identifying multi-terminal net circuit segments (including side nets) for OTC routing, and places these on tracks OTCl according to horizontal constraints. The algorithm proceeds to route the remaining net circuits inside the main channel area subject to horizontal and vertical constraints. The OTCl routing procedure in the GA ensures that only two terminal nets are left within the channel after its action. This reduces the amount of congestion in the channel area. When compared with the best of traditional OTC routers, our algorithm achieved up to 53% reduction in area in the main channel area. In the OTC areas our algorithm utilizes up to 71% less area for routing, when compared with the best of the traditional OTC routers. When compared with best results achieved by a GA channel router, our algorithm achieves up to 55% reduction inside the main channel area

Bibliography

1. SHERWANI, N., *Algorithms for VLSI Physical Design Automation*, Kluwer Academic Publishers, 1995.

2. YOSHIMURA, T., and KUH, E. S., "Efficient algorithms for channel routing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* CAD-1(1):25-35, January, 1982.

3. CONG, J., and LIU, L., "Over-the-cell channel routing," *IEEE Transactions on CAD of Integrated Circuits and Systems*, vol. 9, pp 408-418, April 1990.

4. JOOBBANI, R., *An artificial intelligence approach to VLSI routing*, Boston, MA, Kluwer, 1986.

5. SZYMANSKI, T. G., "Dogleg channel routing is np-complete," *IEEE Transactions on Computer-Aided Design*, CAD-4: 31-41, January 1985.

6. HOLMES, N., SHERWANI, N., SARRAFZADEH, M., "Utilization of vacant terminals for improved over-the-cell channel routing," *IEEE Transactions on CAD of Integrated Circuits and Systems*, 1993.

7. CHANG, L., HSIAO, P., YAN, J., and SHEW, P., "A robust over-the-cell channel router," *IEEE Transactions on CAD of Integrated Circuits and Systems*, vol. 12, no. 10, pp 1592-1599, october 1993.

8. SHIRAISHI, Y., and SAKEMI, Y., "A permeation router," *IEEE Transactions Computer-Aided Design*, Vol. CAD-6, pp. 462-471, May 1987.

9. LIN, M. S., PERNG, H. W., HWANG, C. Y., and LIN, Y. L., "Channel density reduction by routing density reduction by routing over the cells," *Proc. 28th Design Automation Conf.*, pp. 120-125, 1991.

10. GOLDBERG, D. E., *Genetic Algorithms in search, Optimization & Machine Learning, Addison-Wesley*, 1989.

11. GREFENSTETTE, J.J. and BAKER, J.E.,"How genetic algorithms work: A critical look at implicit parallelism," *The proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufman Publishers, San Mateo, California, 1989.

12. HOLLAND, J. H., *Adaptation in natural and Artificial Systems*, MIT Press. 1992.

13. ARSLAN, T., OZDEMIR, E., BRIGHT, M. S. and HORROCKS, D. H., "Genetic synthesis techniques for low-power digital signal processing," *Proceedings of The IEE Colloquium on Digital Synthesis*, London, UK, pp. 7/1-7/5, 1996.

14. BRIGHT, M. S., *Evolutionary Strategies for the High-level Synthesis of VLSI-Based DSP Systems for Low Power*, Ph.D. Thesis, School of Engineering, Cardiff University, 1998.

15. RAHMANI, A. T., and ONO, N., "A genetic algorithm for channel routing problem," *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 494-

498, 1993.

16. PRAHLADA RAO, B. B., PATNAIK, L. M., and HANSDAH, R. C., "A Genetic algorithm for channel routing using inter–cluster mutation," *IEEE*, pp 97–103, 1994.

17. LIENIG, J., and THULASIRAMAN, K., "A new genetic algorithm for the channel routing problem," *7th International Conference on VLSI Design*, pp. 133–136, January 1994.

18. LIENIG, J., and THULASIRAMAN, K., "A genetic algorithm for the channel routing of VLSI circuits," *Evolutionary Computation*, vol. 1, No. 4, pp. 293–311, 1994.

19. GOCKEL, N., PUDELKO, G., DRECHSLER, R., and BECKER, B., "A hybrid genetic algorithm for the routing of VLSI circuits," *Proc. IEEE Int. Symp. Circuits and Systems*, 1996, pp 675–678.

20. LIENIG, J., "A parallel genetic algorithm for performance–driven VLSI routing," *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, April 1997

21. DAVIDENKO, N. D., KUREICHIK, V. M., MIAGKIKH, V. V., "Genetic algorithm for restrictive channel routing problem," *Proceedings of the Seventh International Conference on Genetic Algorithms*, pp. 636–641, 1997.

22. BAKER, J. E., "Reducing bias and inefficiency in the selection algorithm," *Proceedings of the third International Conference on Genetic Algorithms.* Morgan Kaufmann Publishers. 1989.

23. DAVIES, L., *Handbook of genetic algorithms.* Van Nostrand Reinhold. 1991.

24. LIEPINS, G.E and HILLARD, M. R., "Genetic algorithms: Foundations and applications," *Annals of Operations Research.* Oak Ridge National Laboratory, Tennessee. 1989.

25. BRAUN, D., BURNS, J., L., ROMEO, F., SANGIOVANNI–VINCENTELLI, A., MAYARAM, K., DEVADAS, S., MA, H., T., "Techniques for multilayer channel routing," *IEEE Transactions on Computer–Aided Design*, vol. 7, no. 6, June 1988.