CHANNEL ROUTING OPTIMIZATION USING A GENETIC ALGORITHM

A thesis submitted in partial fulfillment of the requirements for the degree of

Bachelor of technology

In

Electronics & Instrumentation Engineering

Submitted by

VAIBHAV KUMAR B.K.

Roll No.:10607027

Electronics & Instrumentation Engg.

R. ARAVINDH

Roll No.:10607023

Electronics & Instrumentation Engg.

Under the Guidance of

Prof. D.P.Acharya

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING,

NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA

CHANNEL ROUTING OPTIMIZATION USING A GENETIC ALGORITHM

A thesis submitted in partial fulfillment of the requirements for the degree of

Bachelor of technology

In

Electronics & Instrumentation Engineering

Submitted by

VAIBHAV KUMAR B.K

Roll No.:10607027

Electronics & Instrumentation Engg.

R. ARAVINDH

Roll No.:10607023

Electronics & Instrumentation Engg.

Under the Guidance of

Prof. D.P.Acharya

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING,

NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA

CERTIFICATE

This is to certify that the thesis entitled "**Channel Routing Optimization Using A Genetic Algorithm**" submitted by **VAIBHAV KUMAR B.K.**, Final year student of Electronics & Instrumentation Engineering, Roll No: 10607027 and **R.ARAVINDH**, Final year student of Electronics & Instrumentation Engineering, Roll No: 10607023 in partial fulfillment of the requirements for the award of B.Tech degree at NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other university/institute for the award of any degree or diploma.

Prof. D.P Acharya

Associate Professor

Dept of Electronics and Communication Engg.,

National Institute of Technology Rourkela,

Orissa: 769008

Acknowledgement

We would like to express our deepest of gratitude to our project guide, **Prof. D.P.Acharya, Associate Professor, Department of Electronics & Communication Engineering, NIT Rourkela** for giving us this wonderful opportunity to work under him and to provide us with valuable guidance and the necessary support throughout the course of this project.

We would also like to express our sincere thanks to **Mr. Prakash Kumar Rout (VLSI Lab, NIT Rourkela)** for helping us throughout the development process of this project.

We are also grateful to **Prof G S Rath**, **Prof U C Pati** and **Prof T K Dan**, members of our project evaluation panel for their constructive criticism, advices to change the way of approaching the problem where ever required and constant inspiration.

We would like to express our gratitude towards our **faculty members** for providing a solid background for our studies and research thereafter which helped us a lot to properly shape the problem and provided insights towards the solution. They have been great sources of inspiration to us and we thank them from the bottom of our heart.

Also, we are extremely thankful to our family for the enormous support and encouragement they have always provided us with, which has been our source of strength throughout the life.

Last but not least, we would like to thank our friends who supported us and everyone else who directly or indirectly helped us in our project.

Vaibhav Kumar B.K
EIE
NIT, Rourkela

R.Aravindh
EIE
NIT, Rourkela

CHANNEL ROUTING OPTIMIZATION USING A GENETIC ALGORITHM

Abstract

A modified approach for the application of Genetic Algorithm (GA) to the Channel Routing Problem has been proposed. The code based on the algorithm proposed in [1] has been implemented for the GA procedures of Initial Population Generation, Crossover, Mutation and Selection. A few improvements over the existing work have been made and the results so far obtained have been encouraging. Further experimentation is being done on the algorithm and other ideas generated during the development of the code are being implemented for faster convergence of the algorithm and for generation of more efficient results. Also application of variations of the GA technique like Vector GA and even other computationally intelligent techniques like Particle Swarm Optimization to the channel routing problem is being thought of.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 Background

In the last few decades, with the amazing increase in component density of integrated circuits, Very Large Scale Integrated (VLSI) circuits have become the norm. This has necessitated the construction of new algorithms and methodologies in all phases of VLSI circuit design and realization. As a result, a variety of routing algorithms, mathematical models, heuristics and automation tools have been developed. This paper deals with the optimization of a specific subset in routing known as channel routing. In case of routing, the placement of the active units is already known, as well as the terminals to be interconnected. The objective is to determine if such an interconnection is possible, and if so, an optimal interconnection is to be found. Optimality in VLSI is measured in terms of chip size, wire length, delay minimization etc. which have a direct impact on the manufacturing cost, the IC performance and its power consumption

1.2 Overview of IC Design Process

The physical design process of very large scale integrated (VLSI) circuits aims to transform a high-level system description into a set of mask geometries for fabrication. The usual approach to this involves a sequence of stages (see Figure 1.1).

- **Design Specification:** Starting from a real-world requirement (e.g.-secure communication"), a high-level system description (e.g., the DES data encryption standard) is developed which includes such parameters as architecture, performance, area, power, cost and technology.

- **Functional Design:** The design is transformed into a behavioral specification which captures the system I/O behavior using mathematical equations, timing diagrams, instruction sets and other devices.

- **Logic Design:** The functional design is represented in logical form, typically via Boolean expressions which may be subsequently optimized to reduce the complexity of the system description.

- **Structural Design:** The logic design is represented as a circuit using components from an available library of modules (e.g., NAND and NOR gates, standard cells, or building-block macros); this may also involve technology mapping steps.

- **Physical Design:** The structural design is transformed into the mask geometry for fabrication while adhering to underlying design rules for the chosen technology.

The last stage in this process, physical design, contains our area of interest.

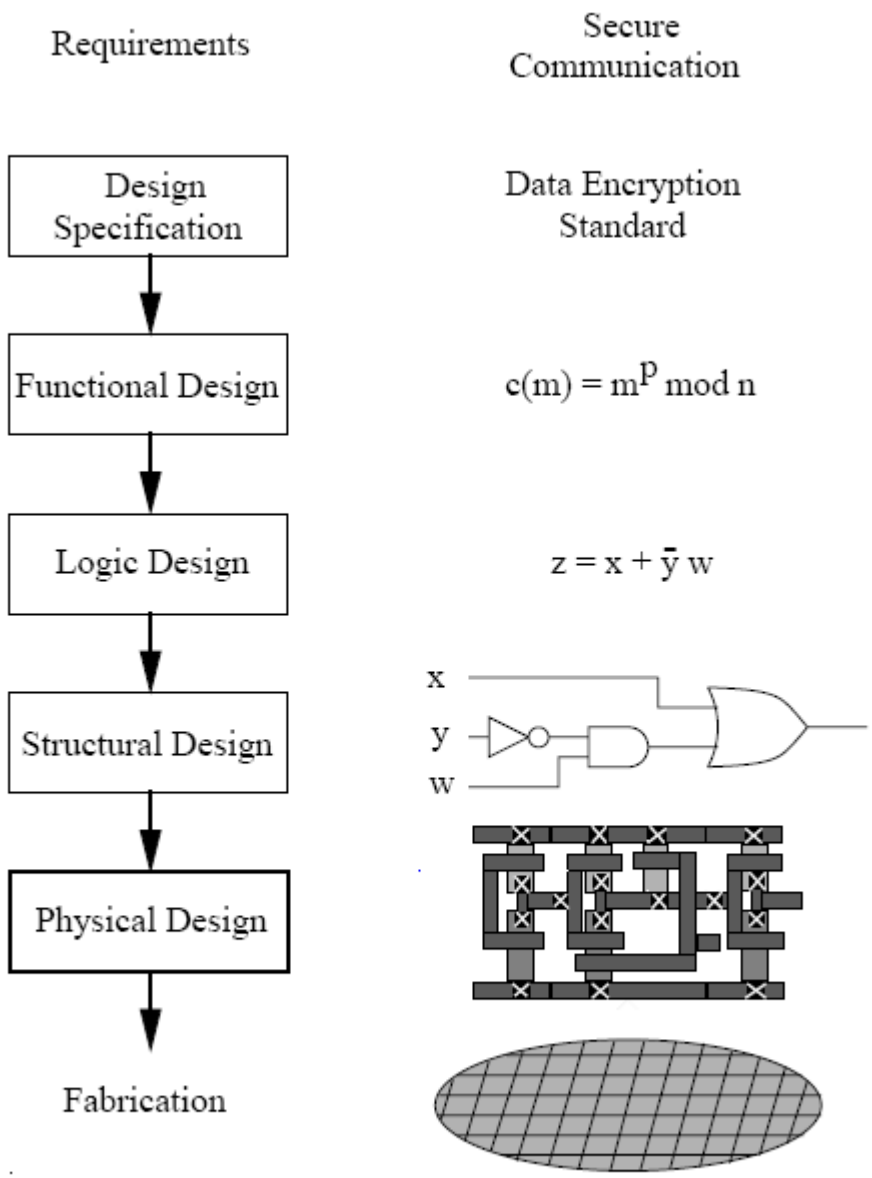


Figure 1.1 The VLSI design process.

Fig: 1.1 The VLSI Design process

Our area of interest, routing, falls under the area of physical design.

1.3 Introduction to Physical Design Process

Owing to its complexity, Physical design process of VLSI circuits is usually separated into four consecutive phases, namely, partitioning, placement, routing and compaction.

- **Partitioning:**

The process of partitioning divides the complex systems of a circuit into subsets which can be designed separately. This is done to minimize the size and thus the complexity of the layout design problem. There are two levels of partitioning: partitioning a system among multiple integrated circuits and partitioning a single integrated circuit design into multiple components. Often, the partitioning stage is implicit; the designer creates the logic for various subcomponents of the system using a library of predefined components which are meant to mimic the physical components.

- **Placement:**

The placement procedure is responsible for the assignment of components generated in partitioning on the area of the chip. Any inferior placement assignment not only affects the chip's performance but might also make it impossible to manufacture by requiring excessive wire length, which maybe beyond available routing resources. Some of the major design metrics in the placement phase include

- ✓ Total wire length
- ✓ Timing
- ✓ Congestion
- ✓ Power

Thus the ideal objective of placement is to position the cells to yield maximum performance using up minimum area. But this is an enormously difficult task as the number of possible placements grows exponentially with the number of placeable objects.

- **Routing:**

The routing phase interconnects specified sets of terminals, i.e., the signal nets of the design, by wiring within routing regions that lie between or over the functional units. (A signal net is a set of the module output terminals and the corresponding module input terminals which need to be connected to each other using routing)

- **Compaction/Spacing**

Compaction or spacing is usually the final step in the physical layout design of VLSI circuits. It is performed to reduce integrated circuit area while eliminating design rule violations. If a design has initial design rule violations, spacing could actually increase the size of the chip. Spacing minimizes the area of an integrated circuit without changing its topology. It is important to keep the topology constant to preserve the timing and performance optimization of the previous phases. Thus, the symbolic layout is transformed to a mask layout with the goal of minimizing the size of the resulting layout.

1.4 Routing-A closer look

VLSI Routing is typically done in two phases called global routing and detailed routing.

- **Global Routing**

Global routing is the decomposition of an integrated circuit interconnection network into net segments, and the assignment of these net segments to regions or channels. The results of the global routing phase will be fed to a detailed router on a channel by channel basis. This divide-and-conquer strategy produces global view solutions while dealing with the complexity of large circuit designs. Usually it is assumed that the positions of the pins of a net have been determined in the placement phase. This phase of routing is especially useful for those technologies, in which the circuit board can be decomposed into smaller parts in a natural way (for example, in the placement phase). In these cases global routing only determines the way in which the wires maneuver between these parts (for example, whether a `mainly horizontal' longer wire segment goes round an obstacle from below or from above). The final course of the wires will only be determined in the detailed routing phase.

- **Detailed Routing**

The detailed routing phase produces the actual geometries which realize the required connectivity on the fabricated chip. Before delving further into the concepts, we will define some of the terms and describe some of the concepts that apply to channel routing in general.

- **Net:** A net consists of a set of terminals that must be interconnected through some routing path.

- **Via:** A connection between two wire segments from different layers is called a via. It is sometimes also known as a contact.
- **Segment:** A segment is an uninterrupted horizontal or vertical wire, and any connection between two terminals will consist of one or more wire segments.

Routing regions:

Routing regions are typically rectangular and are classified on the basis of number of sides that have terminals of nets to be routed. A two-sided routing region is called a **channel** and the associated routing problem is called a **channel routing problem**. This approach allows us to reduce the extremely difficult VLSI layout problem to a collection of simpler subproblems. Moreover, close observation reveals that the channel routing problem is nothing but a special case of the switchbox routing problem in which all the terminals of each net are situated on two opposite boundaries of the grid.

Routing Solution:

A routing solution is a set of wires that connects, i.e., spans, the terminals of a net so that a signal generated at the source will be propagated to all the sinks. VLSI and printed circuit board technologies admit multiple routing layers, where a preferred-direction routing methodology is used to facilitate design, manufacturability and reliability. In other words, the available wiring layers are partitioned, with horizontal wire segments preferentially routed on certain layers, and vertical wire segments routed on the other layers.

1.5 Why Optimize Routing?

Currently, even on a chip with a couple million transistors, a router may have to make several million connections. Even a casual glance confirms the time-consuming nature of this task. But more importantly, these interconnections also occupy a large amount of real estate on the chip. Even when the number of transistors hovers around the 100 million mark, the number of connections to be completed and real estate occupied by the interconnections will become the primary focus of the design. At this point the progress made by the fabrication technology by reducing the feature size may be lost to the increased area required for interconnection. In the future, the performance of systems, as well as the number of components that can be integrated on a chip, depends on the interconnections and packaging technologies. Interconnections and packaging will gain further importance as the feature sizes of the transistors decrease and die sizes keep increasing.

One of the main design goals in the layout of chips is to reduce the die size of the chip, since delay, yield, and cost are directly affected by the die size. To improve performance, yield, and cost, it would be necessary to reduce the die size without reducing the functionality of the chip. Given a specific fabrication process, which determines the minimum feature size, the only way to reduce the die size is to reduce or completely eliminate the real estate used by the interconnect or the routing footprint. The elimination of the routing footprint will reduce the area, thus reducing the worst case interconnect length and increasing the performance. Thus the primary objective while performing routing optimization is to minimize area and maximize performance.

As integrated circuit technology scales into the deep-submicron range, the effect of interconnect on chip performance becomes increasingly dominant. And so does the need to optimize routing.

2.1 ROUTING IN VLSI

In physical design of VLSI, routing involves the determination of the layouts for the wires connecting the terminals on circuit blocks and gates. In most of the cases, the position of these circuit blocks is considered to be fixed. A terminal is a location on the block, typically along one of the edges, which needs to be connected to other blocks. A net is a set of terminals, all of which need to be connected to each other. The routing space between two blocks is called channel. The routing problem is the problem of locating a set of wires in the routing space that connects all the present nets.

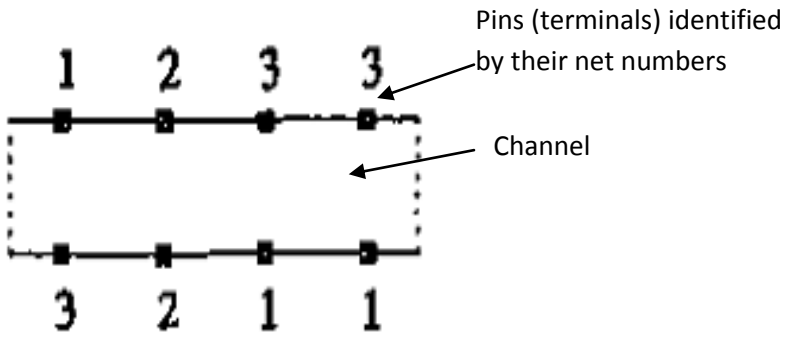


Fig: 2.1 Typical Channel Routing problem

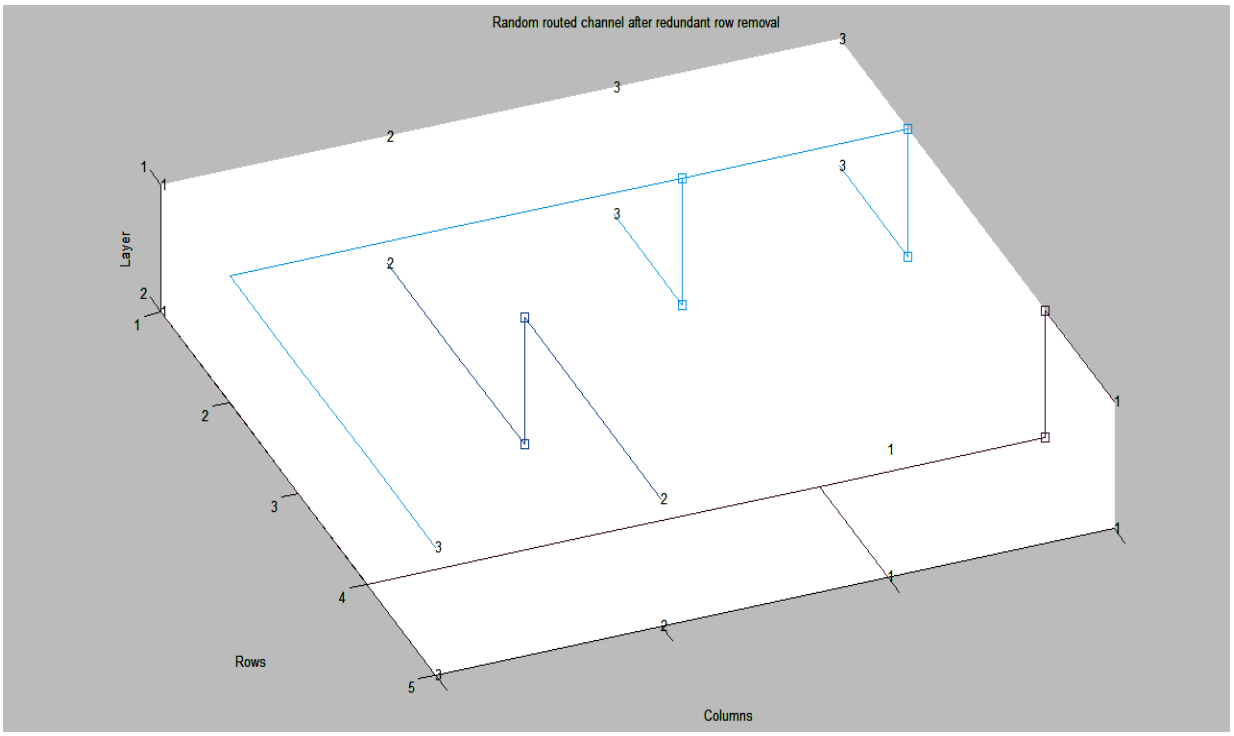


Fig: 2.2 Graphical representation of a possible solution for the above channel routing problem

Routing is the process of connecting pins subject to a set of routing constraints. VLSI routing is usually divided into global routing (to assign nets into certain routing regions) and detailed routing (to assign nets to exact positions inside a routing region). According to the position of the pins, detailed routing can be separated into channel routing (pins are only located on two parallel sides of the routing area) and switchbox routing (pins are placed on all four sides of the routing area).

Here the case of channel routing has been considered for optimization. Proper care has been taken for extension of the code to switchbox routing as well.

2.2 Channel Routing Problem Description

There are certain constraints while carrying out the routing in the channel. They are as listed below:

1. Only horizontal and vertical net segments allowed. Routing done according to Manhattan Geometry
2. Two layers available for routing.
3. Connections between two layers using Vias.
4. Nets should follow minimum distance rule. This is for minimizing crosstalk between parallel running lengths of wires.

2.3 Parameters to be Optimized

The factors concerning the physical layout design which influence the performance of the VLSI circuit are the Channel Routing Area, the Crosstalk the Network Delays. Hence in an optimized solution it is required to minimize these three factors. This gives us three quality/fitness parameters which can be minimized for obtaining the optimized solution for a given channel. They are as follows:

- Minimum Routing Area-

It is observed that routing area has a direct bearing on the length of the routing structure and thus the cost and the complexity of the solution. Thus the need for optimizing the same becomes evident. It is evident that for a given routing problem, the number of columns is fixed. Thus ensuring the use of least area translates to using the least number of rows possible.

- Net length-

The shorter the length of the interconnects, the smaller the associated delay.

- Number of vias-

The introduction of a via between two layers means larger propagation delays and lower fabrication yield. Thus the structure with fewer vias is preferred.

.

3. ROUTING MODELS AND OPTIMIZATION TECHNIQUES

The routing optimization has been an area of extensive study in VLSI design due to its ability to vastly improve the efficiency of the integrated circuit. Over the years a number of Routing models (called routers) have been suggested for the IC Routing problem with each of them having varying degree of success. These models represent different configurations in which the terminals from the active elements of an IC can be connected. These Routers form an important part of the Electronic Design Automation Tools used for synthesis of the VLSI circuits.

3.1 Routers

Routers are a set of points over the IC that are required to be connected. Different Routers have different arrangement of these set of points. The Routers used in detailed routing have been discussed here. These Routers can be primarily classified under two categories: General Purpose and Restricted Routers. The general purpose includes Maze Router and Line Probe Routers. Channel routers, Switchbox routers, power and ground routers, river routers constitute the Restricted Routers category.

3.1.1 Global Routers

3.1.1.1 The Maze Router

This is one of the foremost Routing Models proposed for IC design and was first introduced by Lee [4]. This is based on the Lee-Moore Algorithm. Here the whole routing region is considered as a collection of square cells (grids) arranged in an array [**5**]. The design rules limit these grids to a certain minimum width and the Router can be extended to a multilayer routing. The obstacles in the routing region which can be due to the presence of components, specialized

regions or previously made interconnections are considered as blocked cells in the array. The Lee- Moore algorithm is then used to find the shortest path between pair of terminals that need to be connected to each other along these grids.

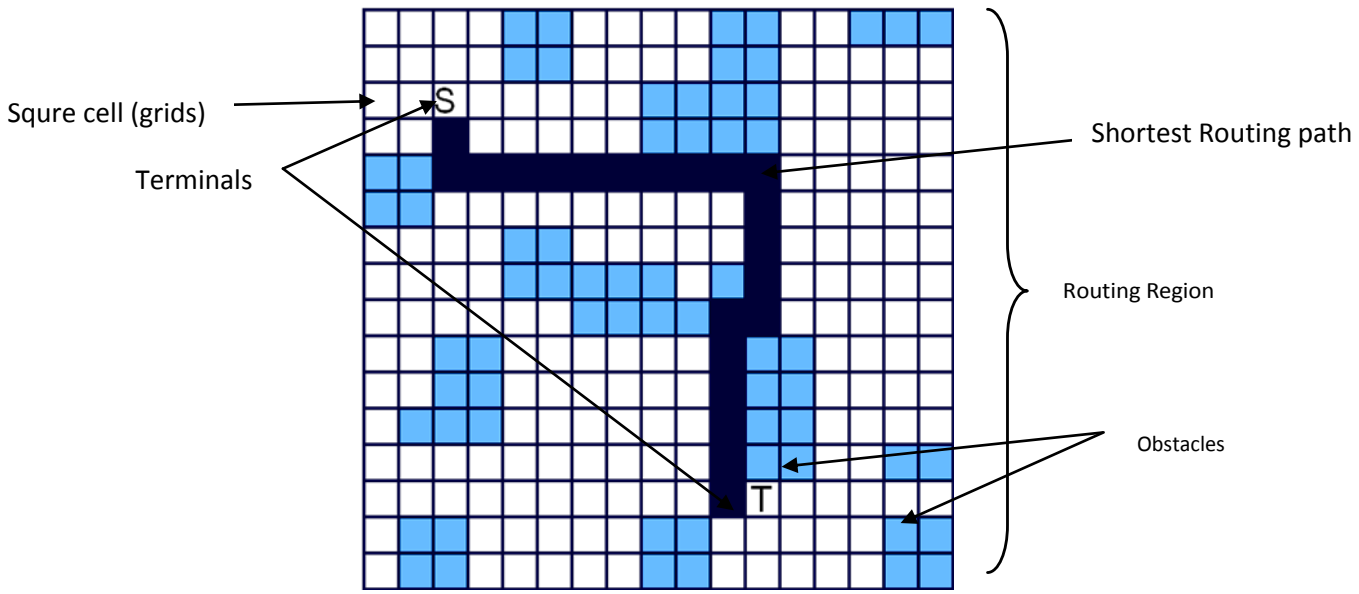


Fig: 3.1 The maze router

These routers have high computation time and require large memories for complex layouts. There are other issues to with this router when there are separate design rules for different layers

3.1.1.2 Line Probe Router:

This Router has the advantage of consuming much lesser memory space as compared to Maze Router. Here the Obstacles and the Routing Boundaries are stored as set of line segments [6]. Line probes from each of the two terminals are extended alternatively until they meet an obstacle. Each time an obstacle is met, a new extension is made from an arbitrary point of the previously extended line. This continues until the line extensions from the two terminals meet each other thus making a successful connection.

This Router though being memory efficient, might not always produce the shortest distance path between the two terminals.

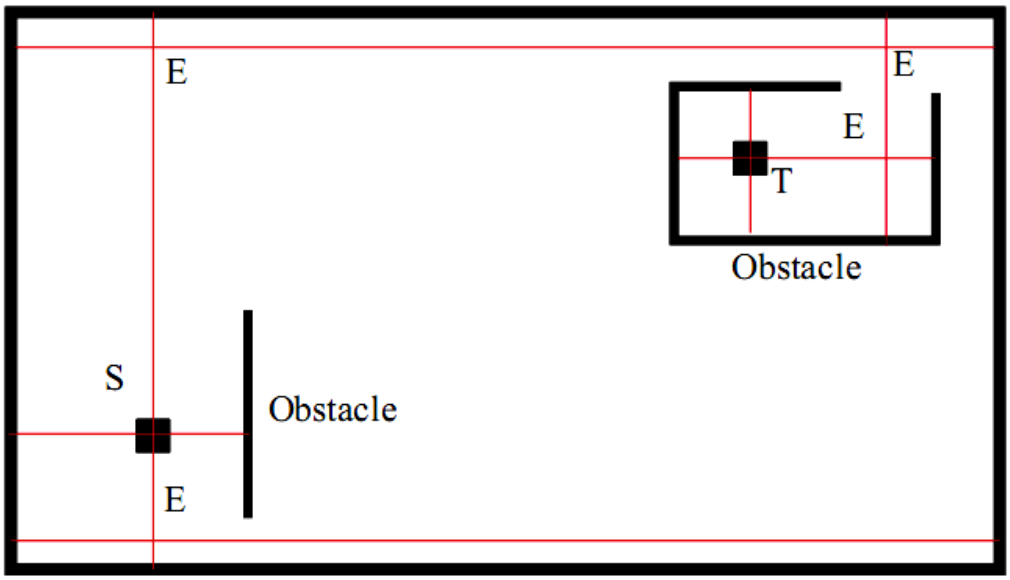


Fig: 3.2 Example of a line-probe router [6] [7]. Escape points are labeled E.

These general purpose routers feature sequential routing wherein only one connection between a given pair of terminals is considered at a time. Hence though the interconnections which have been made previously are taken into account while carrying the present connection, the subsequent connections to be made are not accounted for which makes the Router order dependable.

3.1.2 Restricted Routers

3.1.2.1 Channel Router:

The single wire routers are not ideally desirable as they don't consider the blocking of subsequent connections due to present connections. The Channel Routing acts as an alternative which has concurrency in the connections to the nets that are made. Here the channel is decomposed into a number of rectangular routing regions called channels with certain netlist assigned to each of these. This is done in global routing. The channel has terminals on the top and the bottom side with identical nets to be connected in the channel area. The connections are made along a grid (with grid dimensions limited by minimum distance rule). Though this reduces the complexity, this also limits the scope for space optimization. This method has been quite prevalent for 2 layered routing and many Routing models have been developed based on this. A few important ones have been discussed here.

- Manhattan Routing

In Manhattan Routing there are separate layers for horizontal and vertical line segments thereby having no spacing restriction due to presence of opposite direction segments on the same layer. Changing of direction requires change of layer. Due to this such methods require a number of via connections between layers which increases the manufacturing cost and propagation delay and decreases the fabrication yield. The advantage of such technique is its simplicity and modularity. There are many Routing styles based on this Routing strategy.

- Left Edge Routing

Here the horizontal segments are assigned to the tracks in a way that minimizes the channel width. The left ends of the Horizontal segments required to make the connections between nets are scanned for from left to right and are sorted. The segment with left end closest to the left edge of the channel is processed first. This is placed in the bottom most track. The rest of the segments are checked for the possibility of fitting in the same track with the objective of cramming as many horizontal segments as possible in each track without overlap. This process continues till all the horizontal segments get placed in the channel.

There are problems of Vertical segment conflict in this method. Though this can be solved by Trial and error based segment swapping methods, a more definite solution for this requires further modifications in the routing technique.

- Dogleg Routing

Using doglegs conflict between vertical segments can be avoided. In this technique, the horizontal segments can be split into number of smaller segments to place them in separate tracks and these segments can be connected to each other using vertical segments called doglegs. This method can be further extended to vertical segments as well with horizontal doglegs connecting many parts of a vertical segment. Though in manhattan modeling, dogleg routing requires a number of via connections. So though this method avoids vertical segment conflicts and reduces the size of the channel, it requires additional manufacturing cost and delays due to increased number of vias.

- Non-Manhattan Routers

This includes routing techniques like the Knock Knee routing and the Unrestricted Overlap modeling. These Routers generally exhibit better channel area minimization and reduced number of vias than Manhattan routers. Here the rule of having one direction per layer is relaxed to some extent. That is layers with segments in one direction can have few segments in other direction as well. This technique can be said to be a variant of Manhattan model allowing limted length wire overlap across two layers. Though larger extents of parallel running overlaps can increase the crosstalk between the signals in the two layers.

3.1.3 Area/Switchbox Router

These have fixed boundaries with terminals present on all sides of the boundaries. There functionality is similar to the Channel Routing. These Routing Techniques are widely being used for multilayer routing problems.

3.1.4 River Routing

This is used in specialized cases where it is known that the different connections wouldn't cross each other as they run parallel to one another. Here the objective is to set the alignment and spacing between the connections. These type of routers are used in serial bus architectures, they are planar in structure and use a single layer for routing.

The Routing Technique used in this work is Unrestricted Overlap, grid based Channel Routing for a two layered Routing problem.

3.2 Routing Optimization

3.2.1 NP Complete

It has been earlier proved by [10] ,[11] that Dogleg Manhattan Channel Routing is NP complete problem under restricted settings.

- NP-complete problems:

The complexity class **NP-complete** (abbreviated **NP-C** or **NPC**), is a class of problems having two properties:

- ✓ Any given solution to the problem can be *verified* quickly (in polynomial time); the set of problems with this property is called NP (nondeterministic polynomial time).
- ✓ If the problem can be *solved* quickly (in polynomial time), then so can every problem in NP.

But the most notable characteristic of NP-complete problems is that no fast solution to them is known. That is, the time required to solve the problem using any currently known algorithm increases very quickly as the size of the problem grows. As a result, the time required to solve even moderately large versions of many of these problems easily reaches into the billions or trillions of years, using any amount of computing power available today.

Usually NP-complete problems are targeted using metaheuristic approaches. A Heuristic is defined as an algorithm that works "reasonably well" in many cases, but for which there is no proof that it is both always fast and always produces a good result. Thus a **Metaheuristic** is a heuristic method for solving a very general class of computational problems by combining user-

given black-box procedures — usually heuristics themselves — in the hope of obtaining a more efficient or more robust procedure. A particular sub-set of heuristics known as evolutionary computing techniques has met with considerable success in tackling NP complete problems.

3.3 Evolutionary computation techniques:

Evolutionary computation is a subfield of artificial intelligence that involves combinatorial optimization problems. Evolutionary computation uses iterative progress, such as growth in a population. This population is then selected in a guided random search using parallel processing to achieve the desired end. Such processes are often inspired by biological mechanisms of evolution. Some of the most popular and effective evolutionary techniques include, evolutionary algorithms (comprising genetic algorithms, evolution strategy, evolutionary programming), swarm intelligence (comprising particle swarm optimization and ant colony optimization) etc. We have chosen the technique of genetic algorithm to tackle the problem of Channel Routing Optimization.

3.3.1 Genetic Algorithm:

The evolution process in nature optimizes the fitness of an individual in its environment and thus, can be used as a strategy for mathematical optimization. This is the basic concept behind usage of genetic algorithms to tackle combinatorial optimization problems. The strength of a genetic algorithm results from the ability to perform a fairly efficient search in the search space even if the available knowledge is limited to an evaluation procedure that can measure the quality of any point in the search space [1]. Consequently, genetic algorithms belong to the category of these called weak methods, i.e., problem solving methods that make few assumptions about the problem domain; hence, they usually enjoy wide applicability.

Genetic algorithms, in general, carry out optimization by simulating biological evolutionary processes. The environment in which individuals live affects their ability to survive and the individual best suited for the environment has the highest probability of survival and reproduction. The descendants that inherit desirable characteristics for survival in the environment also have a high probability of survival and reproduction, while other, less fit individuals die out. This principle is known as "the survival of the fittest" and can be used in optimization.

We present a genetic algorithm for channel routing that is based on such a problem specific representation scheme and problem specific genetic operators. The algorithm starts by performing a random path search to create different routing solutions of the channel. These non-optimized routing structures are seen as individuals of an initial population. They are coded in 3-dimensional chromosomes with integer representation. Based on certain quality factors, these routing structures are improved by genetic operators to eventually present a globally optimized routing result. It is shown that the resulting routing structures are either qualitatively similar to or better than the best results available in the literature.

4. THE ALGORITHM FOR CHANNEL ROUTING PROBLEM

The algorithm for the implementation of GA for channel routing problem has been totally taken from [1]. The various procedures, equations and diagrams presented here have been taken from [1] and have just been explained over here for the understanding of the readers.

- Implementation of GA: GA consists of following procedures

1. Initial Population generation: Create random population of n individuals

2. Fitness Evaluation : Evaluate fitness function f(x) of each individual

3. New Population

    0. Selection : Based on f(x)

    1. Recombination : Cross-over chromosomes

    2. Mutation : Mutate chromosomes

    3. Acceptation : Reject or accept new one

3 Reduction: Replace old with new population

4 Loop : Continue step 1 – 4 until optimized result is obtained

4.1 Algorithm for implementation of GA for Channel Routing Problem

```
input(channel)
input(no_of_generations)
input(population_size)
for i=1 until population_size        <-- Initial Population Generation by random Routing of channel
        create(individual_i)
endfor

fitness_calculation(P_i)
individual_best=best_individual(P_i)
P_0= P_i
for g=1 until no_of_generations      <-- Loop for recursive optimization over no. of generations
        P_g= P_g-1
        P_crossover=ϕ
        for offspring=1 until max_descendant
                p_α=selection(P_g)
                p_β=selection(P_g)   } Probability bases selection of Parent
                P_crossover= P_crossover U crossover(p_α,p_β)   <-- Crossover using new individuals (offspring)
        endfor

        fitness_calculation(P_crossover)
        P_g= reduction(P_g U P_crossover)   } Reduction: Removal of low quality individuals to keep the total

        individual_best=best_individual(individual_best, P_g)
        mutation(P_g)                <-- Mutation of a few random selected individuals
        fitness_calculation(P_g)
endfor
optimize(individual_best)            <-- Optimized individual
```

4.2 Initial Population Generation

The first procedure in the implementation of Genetic algorithm is to create a randomly generated initial population of individuals. Each of these individuals is a solution to the channel routing problem. These individuals are created randomly with a view to create sufficient diversity within the population which is of vital importance in the implementation of GA. The channel is mapped onto a 3 dimensional matrix where the 2 dimensions represent the channel area and the 3rd dimension represents the two layers. A connection present at a node is represented by a "node value" corresponding to the pins from which the connection has been made. For the purpose of

mapping the interconnections for each node with the adjacent nodes, a “direction of connection value” has been concatenated with the nodal value. During the whole development of the code, special care is taken so that the constraints of interconnections aren’t violated in any of the possible random configuration of the connections. The whole procedure can be divided into the following sub-procedures.

1. Empty(Unconnected) channel creation

   The inputs for the number of pins, and the corresponding net values to the pins are provided according to the channel routing problem. The no. of pins decide the no. of columns for the matrix. The no. of rows for the channel matrix is then randomly selected from a range of values. Using this information an empty channel is created with the pins on the top and bottom row of the channel matrix represented by their corresponding net number. The purpose of the rest of the code is to rout all the connections in the channel matrix.

Net no. corresponding to a pin

```
abs_channel(:,:,1) =

     3     2     1     3     1
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     1     3     2     3     2

abs_channel(:,:,2) =

     3     2     1     3     1
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     1     3     2     3     2
```

Fig: 4.1 A typical 10 pin, 3 net empty channel matrix

2. Source and Target pin selection

Each time 2 pins - a source and a target pin (having same net numbers) are randomly selected which undergo repeated trials for making a connection in between them. As the whole random routing procedure is recursive, at any point of time we have a set of unconnected pins (represented by S) and a set of pins with connections existing with one or more pins (represented by T). The source pin is always one of the unconnected pins (selected from set S). The target pin could be either an already connected one or an unconnected one.

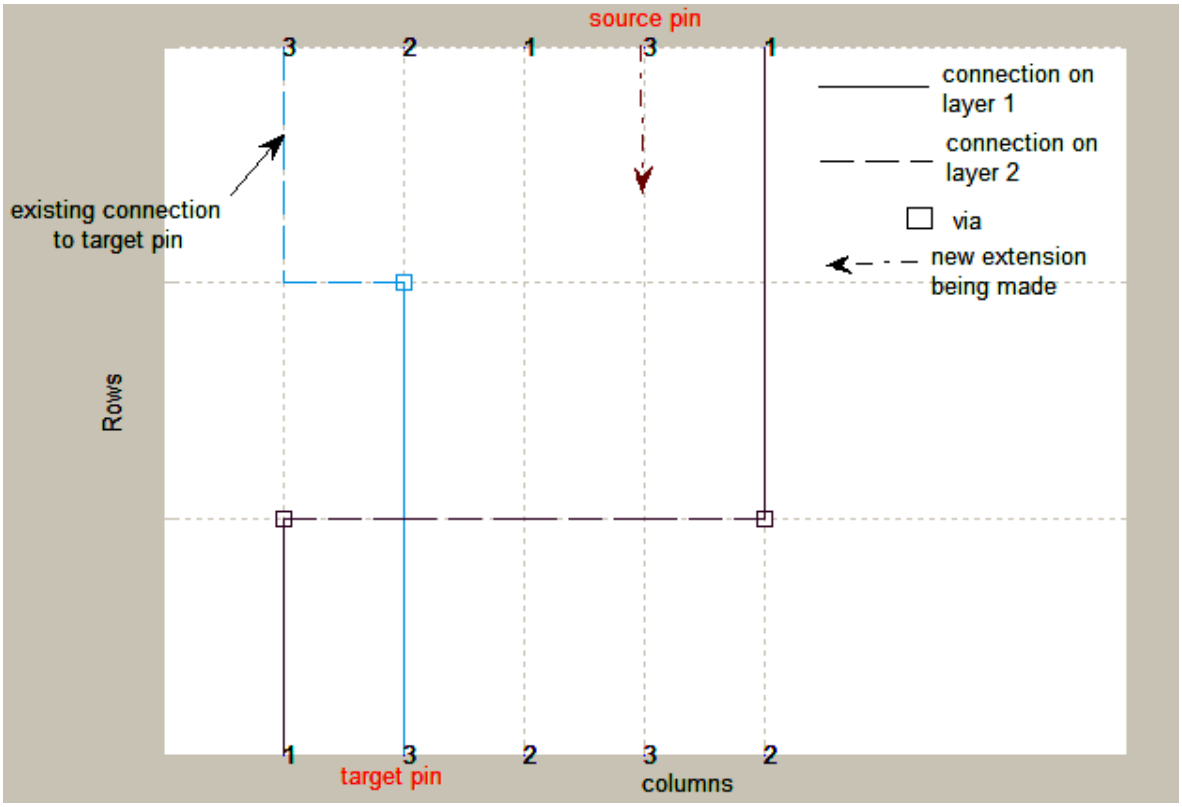


Fig: 4.2 Target pin (selected from set T) and source pin

3. Repeated horizontal and vertical extensions from the selected pins

   Horizontal and vertical extensions are carried out from both the pins in alternate turns. For this first the layer for the extension is chosen. In one of the layers horizontal layers extensions are preferable and in other vertical extensions are preferred. For this purpose for each layer, extension to be made in preferred direction is given a 2/3 probability of undergoing the extension in that layer and the unpreferred extension is given 1/3 probability. After extension in one direction (horizontal or vertical), a randomly selected point on the extension is chosen which decides the point through which extension has to be carried out in other direction. This process is carried out until

   a. Extension lines of both points meet each other on the same layer.
   b. The extension lines of "source pin" touches an already existing connection of the "target pin".

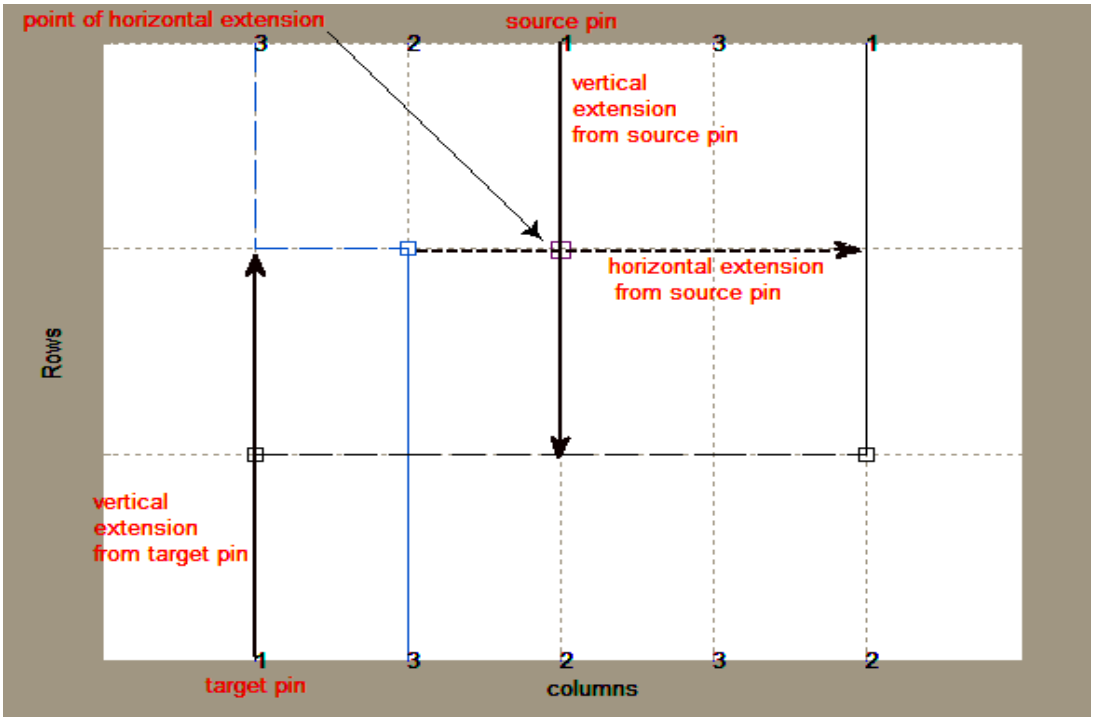


Fig: 4.3 Extensions made from source and target pins

4. Backtracking

Once the connection between the source and the target has been made, the parts of the extensions not involved in the actual connection have to be removed. This is done by backtracking from the point of connection.

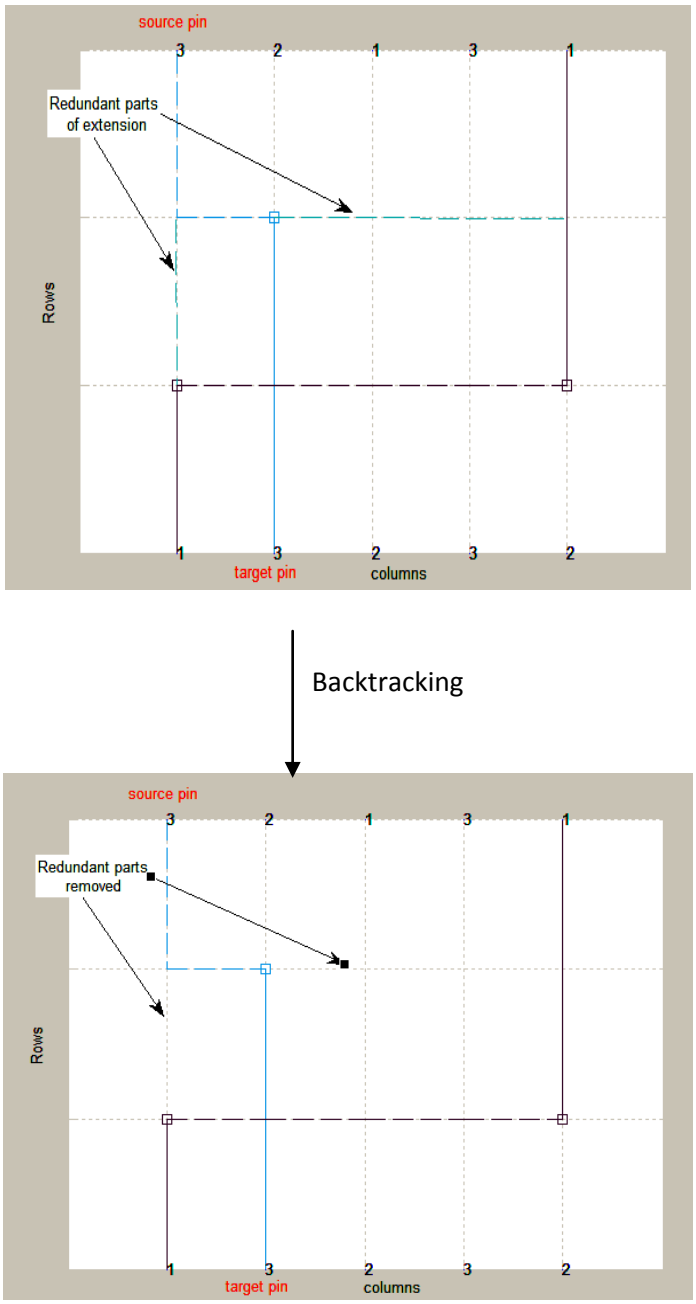


Fig: 4.4 Backtracking of a connection

For the purpose of backtracking, first the particular extension to one of the pins (the "connection made" pin) to which the other pin (the "connection making" pin) has got connected is determined. The extensions that have been made after that particular extension of the "connection made pin" are erased as they are redundant and not part of the actual connection. After this the for each of the remaining extensions, the part of the extension which is in between the two points of extensions are kept as it is and the part which doesn't fall in between the two points is erased. This procedure finally gives us a connection between two pins with all the redundant parts of extensions removed.

5. Adding the direction of connection data for each node

For each node, it is necessary to add the information about the connection with the adjacent node so that using this information the actual channel connections can be mapped from the channel matrix numerical data. This information is again necessary to determine the connections of the newly created nodes after row addition procedure. The convention for the nodal value has been made to have a maximum of 99 pins in the channel.

The convention used for the direction value is as follows:

Nodal value=Direction Value* 10000 + Source Pin*100+ Target Pin

Direction value: L C R

L: Layer direction value

L= 0 ; no via present at the node

= 1; via present at the node

C: Column direction value

C= 0 ; no vertical connection at the node

= 1 ; downward connection present to the node

= 2 ; upward connection present to the node

= 3 ; both downward & upward connection present to the node

R: Row direction value

R= 0 ; no horizontal connection at the node

= 1 ; Right connection present to the node

= 2 ; Left connection present to the node

= 3 ; both left & right connection present to the node

Typical example of a nodal value: 1321004

The node has a via connection (1), both upward and downward connection (3) and left connection (2). The source pin is pin no. 10 (10) and the target pin is 4 (04).

This is carried out in the Backtracking process itself. Initially the “direction value” is added at the point of connection. And then as the extensions are backtracked, the directional value is added according to the direction of a given extension. At the node where a given extension in a particular direction ends and there is either another extension in a different direction or there is the presence of a via, the direction value of previous extension is stored so that it can be added to the nodal value while carrying out the new extension. Care has been taken for the direction value of a node in case of loops by having the direction value for a new connection to be added to the direction value of existing connection for a particular node.

100	2	1	3	500
1100506	0	0	0	0
300506	0	0	0	0
300506	0	0	0	0
300506	0	0	0	0
300506	0	0	0	0
300506	0	0	0	0
200506	700	800	3	1000

```
abs_channel(:,:,2) =
```

100	2	1	3	100506
1010506	30506	30506	30506	220506
0	0	0	0	0
0	0	110810	30810	120810
0	0	300810	0	300810
0	0	300810	0	300810
0	0	300810	0	300810
600	700	200810	3	200810

Fig: 4.5 Nodal values for a 10 pin, 3 net channel during routing process

6. No scope for extension: Multiple extension trials

The process discussed till now is for the case where after multiple extensions the connection between the two selected pins has been made. However in step 3 there is always a good possibility of generation of a situation wherein there is no scope for further extension from one of the given selected pins from the “random point of extension” present on its previous extension. In such a case the strategy of “Multiple extension trials” is used. Here first a possibility of extension in other layer from the “random point of extension” is tried for. If that is not possible the “random point of extension” on the previous extension is redefined and a possibility of extension from this new point is considered. This process of changing the layer and the “random point of extension” is carried out a certain number of times (5 in our case). If even after this the connection is

not possible the existing extensions for the selected pins is erased and a row is added to the channel at a random position. After this horizontal and vertical extensions are again made from the selected pins with the purpose of making connection between the two.

7. Random Row Addition

A position is randomly selected wherein a row is inserted in the channel. For such a row insertion, it is required that the vertical extensions that were initially present between the rows adjacent to the inserting position are also present in the newly made row. Also it is required to update the row no. of the pins of the channel.

100	2	1	3	100506
1010506	30506	30506	30506	220506
0	0	0	0	0
0	0	110810	30810	120810
0	0	300810	0	300810
0	0	300810	0	300810
0	0	300810	0	300810
0	0	300810	0	300810
600	700	200810	3	200810

Row added after fourth row

100	2	1	3	100506
1010506	30506	30506	30506	220506
0	0	0	0	0
0	0	110810	30810	120810
0	0	300810	0	300810
0	0	300810	0	300810
0	0	300810	0	300810
0	0	300810	0	300810
0	0	300810	0	300810
600	700	200810	3	200810

Newly added row

Fig: 4.6 Row addition after 4th row, the downward connections of the previous row are retained in the newly inserted row

8. Connection not possible

   If in spite of making a certain no of random row addition (5 in our code), the connection between the selected pins are not possible then the whole channel is erased and the random routing process is carried out all over again. This is even done when the total number of random row addition for a given channel exceeds a certain number.

9. Redundant Row removal

   Once all the connections in the channel have been made, the redundant rows in the channel can be removed. These rows neither have any horizontal connections nor do they have any vias. So removing such rows doesn't affect any of the existing connections and it helps our purpose to reduce the channel area.

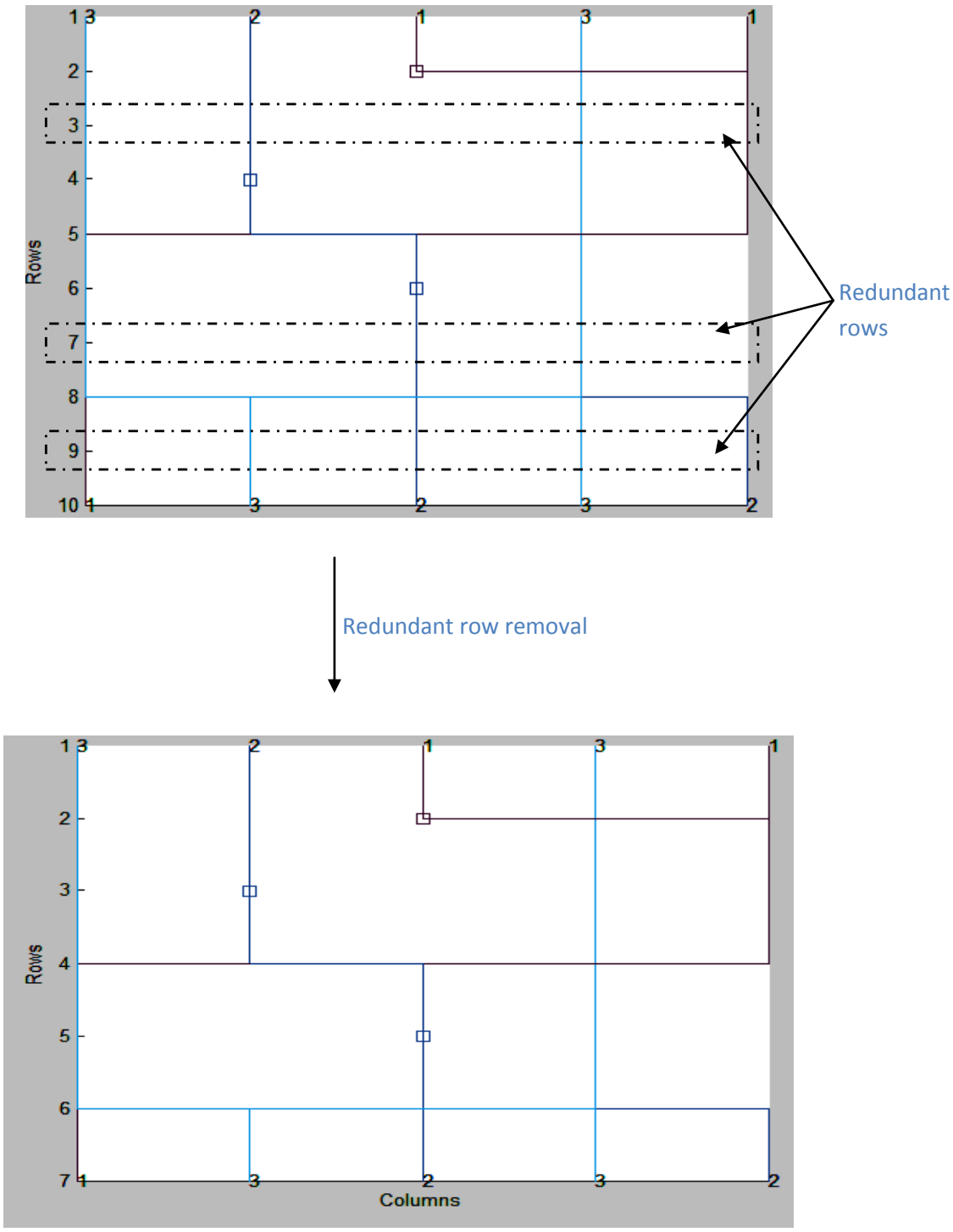


Fig: 4.7 Redundant row removal for rows 3, 7 and 9 due to the absence of horizontal extensions of vias in these rows (showing the top view of the channel)

10. Graphical representation of the channel

After each individual creation, the nodal data of the channel matrix can be mapped to get the 3-D graphical representation of the channel. This gives a much clearer representation of the connections and hence can be used to detect the errors in the final individual created. This in turn makes it easier to debug the code. Here for distinguishing between connections from different nets, code has been written such that a different colour is given to each of these connections. Also at via positions square markers have been used. These facilitate easier readability of the connections.

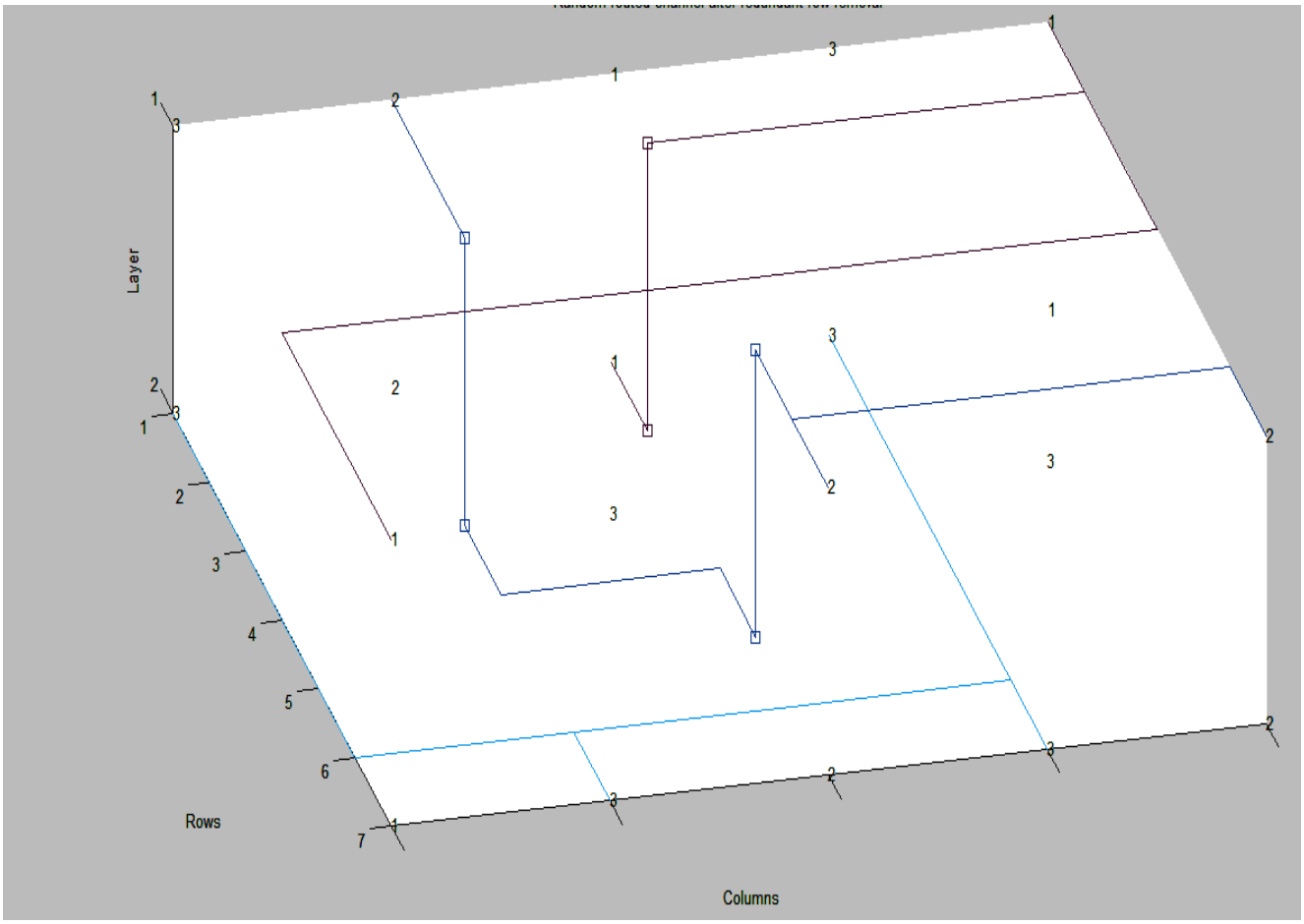


Fig: 4.8 A 3-D graphical representation of the routed channel showing all the pin connections (each net connection shown by a different colour)

11. Creation of number of individuals to create a population

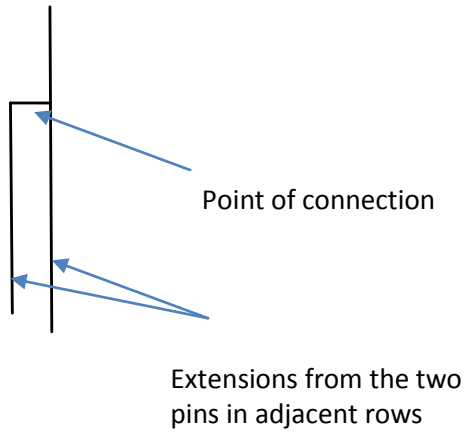
    A number of individuals are created by following the above procedure thus generating the initial population necessary for the implementation of the GA. Each of these individuals can be stored in a 4 dimensional cell array.

The coding for the random routing procedure became quite complicated due to the presence of a number of special cases each of which had to be considered while the development of the code. Few of the troubles faced during the process is as given below.

- ✓ Need for pin based nodal values rather than net based: The nodal values given to the nodes while carrying on the extensions were based on the pin numbers form which the extensions are being made (not on the net number). This was done in order to differentiate between an actual connection between two pins and the case where the extension from same pin gets connected to another extension of the same pin (thus forming a loop). Because of this the nodal value was pin based and the track of connected pins with same net no. had to be kept separately. And then once all the connections were made (random routing is done) the pin based nodal values were converted to net based nodal values.
- ✓ Difference in the characteristic of Source and the Target pin: Source pin is always selected from set S while the target pin could be from set S or T. Now if both the source and the target pins are from S, then extensions from the two pins needs to meet each other to make a connection. However if the target pin is from T, the source pin can get connected to any of the existing connections between pins of the same net and there good possibility of cases when extensions from target pin do not become a part of the new connection that has been made. For this firstly its required to keep track of all the

connected pins (present in set T) with the same net no. each time a source pin is selected and a target pin is selected from set T. Then during extension when nodes are checked for the possibility of connection, links connected to each of these pins needs to be checked for. Generation of such a connection during the extension process from source pin has to be traced and in such cases the extensions from T have to be removed without altering the nodal values of any of the previous connections from T (during backtracking).

- ✓ In case there is no room for extension for target pin in cases when it has been selected from T, multiple extension trials is applied. However if that fails to make any further extensions for target pin, the row addition process is not carried out as there is still scope for source pin to get connected to the links from one of the pins with same net numbers.
- ✓ There were number of troubles during the backtracking in cases were the target pin was selected from T. If the source pin was to get connected to one of the existing connections instead of the extensions from T, then care had to be taken to avoid any sort tampering with the nodal values of the previously existing links which were overlap with the new extensions made from T. Also in cases where the target pin was from T and the source pin gets connected to the extensions from target pin, we had to take care while determining the direction values of the code so that the previously existing nodal connections due to the already existing links from connected pins do not get tampered by the new nodal connections formed by the overlapping new extensions of target pin.
- ✓ Case of successful connection without any visible extension in the channel matrix: This is the case in which the two extensions one each from source and target run parallel to each other in adjacent rows or columns and then make a connection without any extension in other direction as seen in the channel matrix.



The actual channel

0	1500
0	1500
1200	1500
1200	1500
1200	1500
0	1500

Point of connection

Fig: 4.9 The case as shown in channel matrix form

As seen from the channel matrix, there is no horizontal extension visible in the channel matrix that is actually making the connection. And hence due to this the horizontal extension is not considered as the extension making the connection by the code. Hence this generates an error while finding the directional value at the point of connection. So for special conditions have been added to find the occurrences of such cases.

4.3 Crossover:

Crossover is the pre-dominant component used in the simulated evolution process to which the randomly generated population is subjected. It is simply the genetic operator which allows the combination of two individuals to create a descendant. The individual here refers to an already created solution to the given routing problem which is chosen from the population according to its fitness value. The objective here is to obtain a new solution which possesses a combination of the routing structures of both parents. This is an entirely random process. The task of making this process convergent is done by the process of selection as mentioned before. The procedure involved in crossover is as follows:

1. Selection of parents:

Two individuals, alpha and beta, are chosen according to their respective fitness values from the existing population. The no. of columns remains the same in both individuals as they are solutions to the same routing problems. The no. of rows though may or may not be the same as the individuals are generated through random routing procedure and may have been subjected to multiple crossovers and mutations.

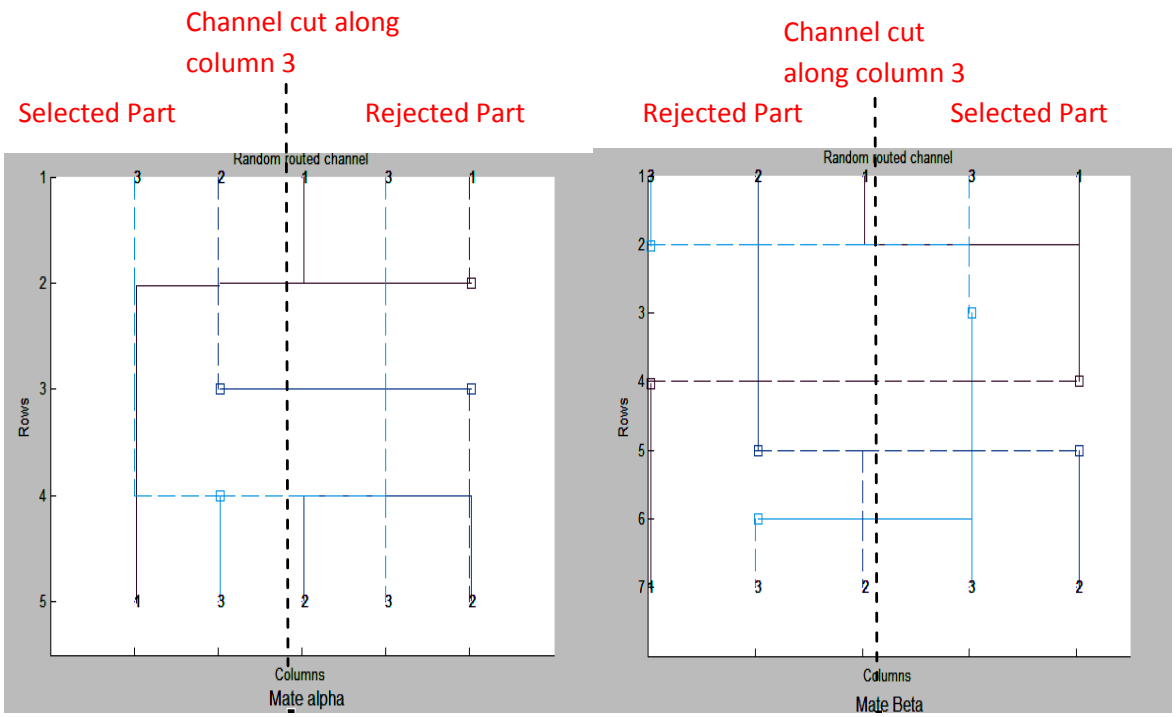


Fig: 4.10 Each Parent channel is cut along a selected column and only one of the parts of each parent is used for crossover

2. Selection of sub-solutions to pass on:

   This is done by choosing what is known as a cut column denoted by x_c. The cut column is randomly chosen subject to the condition given by $1 \leq x_c < x_{ind}$,

   where x_{ind} refers to the number of columns of the individuals.

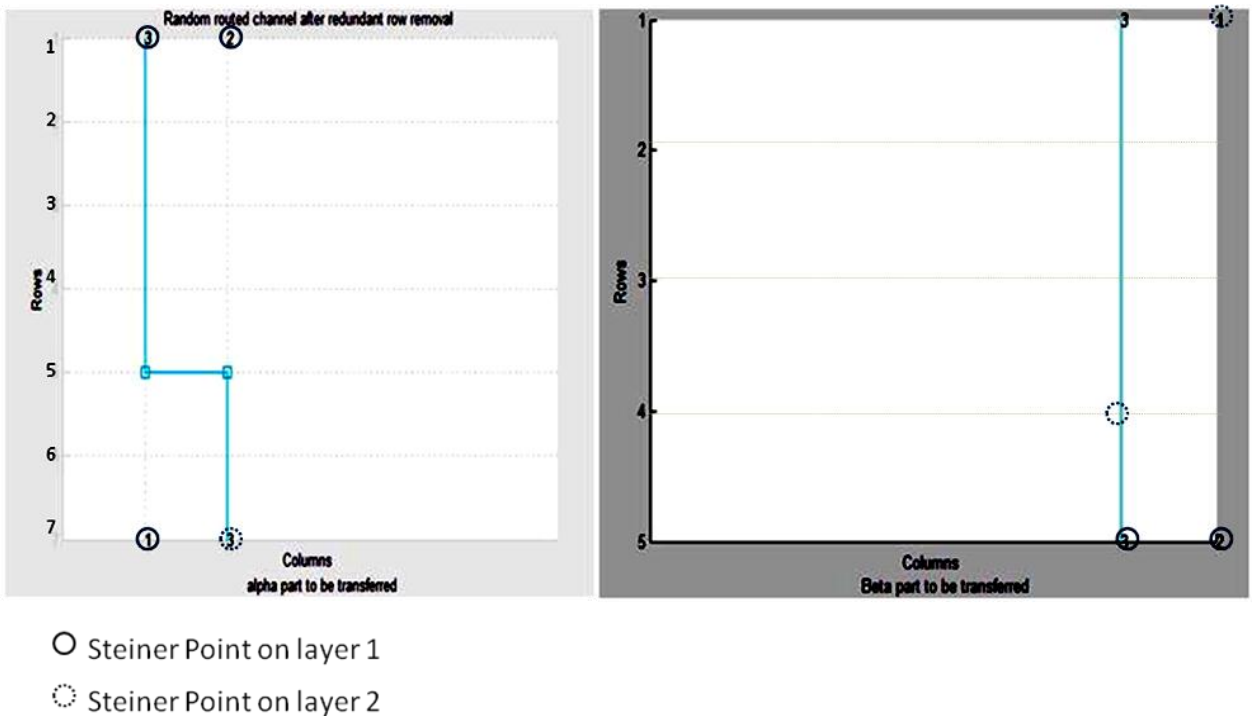


Fig: 4.11 The parent channels Alpha and Beta with their steiner points obtained by retracing back and removing the dangling connections created by cut column

Then the individual alpha transfers its routing structure, to the descendant gamma, which is

- ✓ Located on (x,y,z) such that $1 \leq x \leq x_c$, $1 \leq y \leq y_\alpha$ (where y_α refers to the number of rows of individual alpha) , $1 \leq z \leq 2$ and
- ✓ Not cut by the cut column x_c.

Correspondingly individual beta also transfers its uncut routing structures such that $x_c < x \leq x_{ind}$, $1 \leq y \leq y_\beta$, $1 \leq z \leq 2$.

It is important to note that the parents are not overwritten in the effort to create a new individual. Only copies of the 2 parents are used in order to create a new individual. The

decision as to which individuals get passed onto the next generation is solely based on their respective fitness values.

3. Here it is important to note that individuals alpha and beta have their cut connections wiped out until their next Steiner point or pin is reached and are not transferred into descendent gamma. The steiner point is nothing but a junction at which 2 or more routing structures of the same net can be taken to be linked.

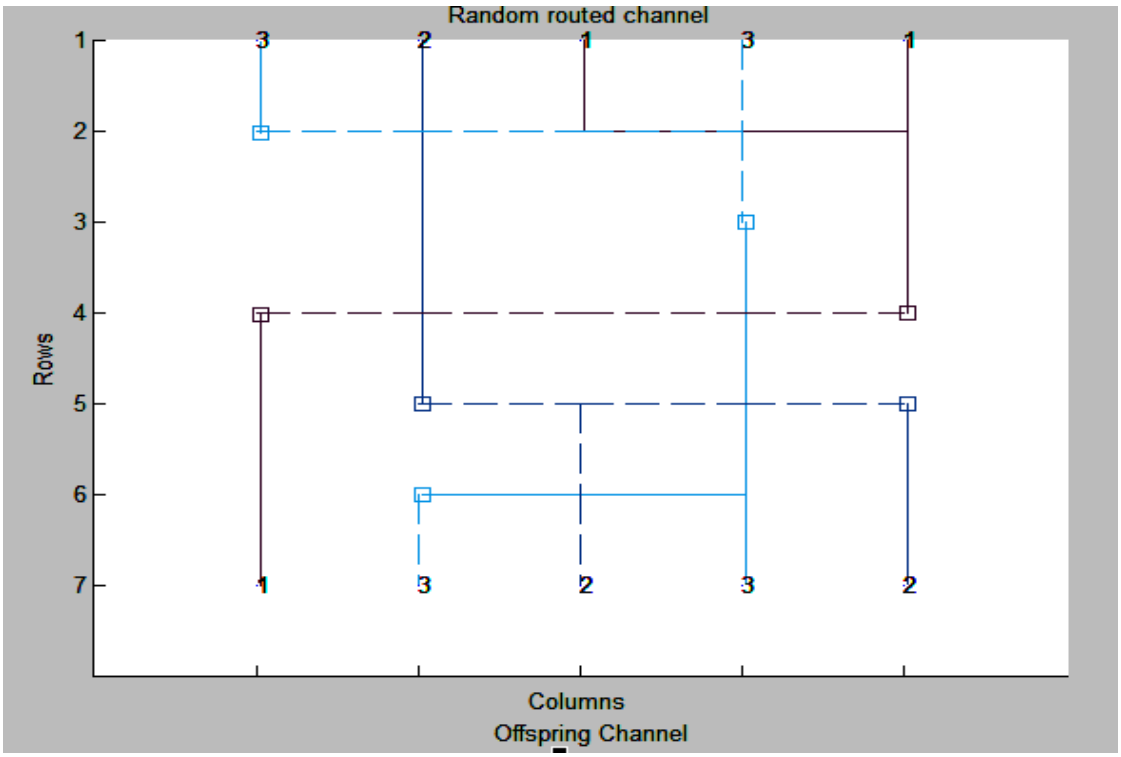


Fig: 4.12 The offspring channel generated by crossover between parent channels Alpha and Beta

An important pitfall that needs to be carefully avoided is when a connection on one layer is wiped out; vias (if any) should have their corresponding connection wiped out from the other layer. Care needs to be taken that only this inter-layer connection is wiped out and not any intra-layer connection unless specified by the procedure.

4. After the connections are wiped out as per the requirements, the individual is no longer a valid `solution of the given routing problem. Thus, this partly connected individual is again subjected to the same random routing routine described before in order to make it a proper candidate solution. Since the choice of the cut column is entirely random, it happens more often than not that the newly routed alpha and beta are incompatible with each other (i.e. they might have different number of rows). It is also to be noted that it is most likely that alpha and beta had different number of rows right at the outset.

   So this difference in the number of rows of the two individuals is dealt with by allocating the descendent gamma with number of rows equal to max(no_of_rows_alpha, no_of_rows_beta). The mate which contains fewer rows is now extended with additional row(s) at random position(s) before transferring its routing structure to the descendant. Since our intention at every juncture is to reduce the number of rows as much as possible, any row which does not have any horizontal routing connection is removed.

5. Both alpha and beta are subjected to inner-routing. If the attempts at random routing are not successful within a pre-set number of extension lines, these extension lines are deleted and the channel is extended at a random position through random row addition. If even after a given number of row additions if the channel is not completely routed, then the generated individual gamma is deleted. Individuals alpha and beta are again subjected to the crossover operator using a new random cut column.

Approach And Problems Faced:

- ✓ The two individuals alpha and beta in their genotype form are taken as arguments. In order to perform the crossover operation two separate functions horizontal and vertical are defined. The functions horizontal and vertical take care of the removal of the horizontal and vertical parts of routing structures that need to be removed respectively.
- ✓ Since any given routing structure is almost certain to contain both horizontal and vertical segments, the approach chosen is essentially recursive.
- ✓ The horizontal and vertical functions call each other alternatively as per the characteristics of the given routing structure.
- ✓ Thus in essence what can be described as a forward-tracking process is undertaken. The algorithm traverses along the cutting column. Once a connection is found its corresponding net is discerned from the coding scheme. Using this and the directional markers this connection is tracked until its next steiner point or pin is reached and the intervening routing structures are deleted.
- ✓ Whilst all this removal and deletion is being done, the directional information is constantly updated so as to not to leave any inconsistencies.
- ✓ In order to extract all this information from the genotype, case-specific decoding segments have been written to meet the various specific conditions that arise on different occasions.
- ✓ Sometimes it so happens that the adjacent pin belongs to the same net as the connection to be deleted but is in fact part of another connection. In such cases care has to be taken so that necessary connections are not removed on account of this.
- ✓ The connection which is touched by the cutting column is removed until its steiner point or pin is found. While this removal is being undertaken it might so happen that one of the pins might also

have a via facilitating inter–layer connection. When this connection is removed it must also be ensured that the corresponding via on the other layer is also removed. This makes sure that the inter-layer connection is severed in a proper manner.

- ✓ Row addition and removal:

  In some cases, a certain row might not have any horizontal connections or vias. These rows are redundant and in accordance with our primary requirement are removed. A separate code segment has been written in order to allow row addition at random position to make the two individuals compatible.

- ✓ Redundancy-removal:

  In some cases after the entire cut connection removal process is completed it is found that there are some routing structures which do not contribute to any meaningful connections. Sometimes dangling vias are also found. These are traced and removed so as to remove redundancy.

- ✓ Our attempt is to retain good sub-solutions from both the parents to produce better offspring. But often it happens that there are not many viable connections being retained. So it might be better to randomly choose some nets from both the parents and retain them. The unconnected portions can then be subjected to the usual random routing procedure.

- ✓ It is easily observed that the quality of the individual is highly sensitive to the choice of the cutting column. This offers scope to improve the results of the crossover operator by making the process of choosing the cutting column more pseudo-random with a slight dependence on the fitness values.

4.4 Mutation:

Mutation is another genetic operator that is used in which a chosen individual is subjected to random changes in its routing structure. Such random mutations allow introduction of new characteristics into the next generation. They facilitate movement in the search space. This is because only mutation can introduce new information (alleles). The main purpose of mutation is to allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution. Moreover mutation is the only means by which lost information can be restored to the population. Despite all this the probability of an individual to be subjected to mutation is maintained at a very low level (~0.001). This is because if the mutation probability is increased then the method becomes closer to just wild guessing

In this case four different mutation operators are applied in a random order with certain mutation probability to each of the individuals. Those mutation operators being:

- ✓ Mut_1: A surrounding rectangle with random sizes (xr,yr) around a random center position (x,y,z) is defined. All routing structures within the rectangle are deleted. The remaining net points on the edges of this rectangle are now connected again in a random order with the same random routing method.
- ✓ Mut_2: A random number of nets n_r is defined such that $1 \leq n_r < n_{ind}$. Now n_r nets are chosen randomly and are deleted. Now the channel is routed again with the same random routing method.

- ✓ Mut_3: At a random row position y_{add}, with $1 \leq y_{add} \leq y_{ind}$, an additional row is added. Now net segments are chosen randomly from the neighbor row(s) and are placed on the newly added row. Now the channel is routed again with the same random routing method.
- ✓ Mut_4: At a random row position y_{del}, with $1 < y_{del} < y_{ind}$, a row is removed. Affected net segments are traced until their next Steiner point or pin is reached and the channel is routed again with the same random routing method.

4.5 Selection:

The genetic algorithm entails maintenance of constant population size in every generation. But using crossover operator we generate new individuals in every generation. So at every stage pruning of less promising solutions is done. The entire procedure is as detailed below.

- ✓ The first step in this process is to calculate the fitness function corresponding to every individual. The fitness function of an individual depends upon the fitness parameters of the Routing Area, the Net length and the number of vias for that individual.
- ✓ Once the fitness functions are calculated taking into account the above three parameters, selection is done. The principle followed is "Survival of the fittest".
- ✓ However, it is also necessary to maintain diversity and avoid *crowding* so that the entire space gets explored (local minima vs global minima).
- ✓ In order to abide by the above requirements, the approach used here is Stochastic sampling with replacement.

- ✓ In other words it is nothing but *Fitness Proportionate Selection* (aka roulette wheel selection). Every individual has a chance but the fittest are more likely to be passed onto the next generation.

- ✓ So the actual fitness functions are given by,

$$F_1 = \frac{1}{\text{No of rows}}$$

$$F_2 = \frac{1}{\sum_{i \in nets} (l_{acc} + a * l_{opp} + b * v_p)}$$

Where,

l_{acc} = Net length in the preferable direction in a layer

l_{opp}= Net length in the opposing direction in a layer

v_p = Number of vias

a, b = Cost factor

- ✓ The calculation of the final fitness value is done in such a manner so that area minimization i.e. number of rows predominates over other factors.
- ✓ This fitness value plays a crucial role in selection of mates for application of crossover operator.
- ✓ Thus the probability of any individual solution P_x, $P_x \in P_c$, getting selected to the next generation is given by,

$$\text{Prob}\{P_x \text{ getting selected}\} = \frac{F(P_x)}{\sum_{y \in P} F(P_y)}$$

- ✓ The fitness function is crucial to the performance of the algorithm. Too strong fitness selection bias can lead to sub-optimal solution. While on the other hand, too little fitness selection bias results in unfocused and meandering search.

- ✓ An attempt to increase population diversity by increasing the population results in increased computational time for each generation which might translate to exponential increase in total time taken. Thus an optimum value needs to be chosen for population size.
- ✓ Another factor which has been considered is that there is a very slight chance of the best members of a generation getting deleted during normal evolution. Thus an effort can be made to preserve such individuals by keeping the best individual seen so far in a special register. This ensures that their characteristics continue to influence subsequent generations. This is otherwise known as Elitism.

5. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The Algorithm has been implemented in Matlab and spans to over 3500 lines of code (The code in [1] extended to about 8000 lines and was written in FORTRAN). Throughout the code development, there has been a consistent effort to keep the code as concise as possible. For this purpose a number of generic functions have been created (over 40 in number) due to which the final code has become quite modular and flexible with proper scope for future modifications, inclusion of additional features and extension of the algorithm for similar Routing Problems. The code was run on a 1Gb RAM, 1.66 GHz AMD dual core Hardware Platform with Windows XP as the OS. We provide here the final routing optimization results obtained by our work. It can be seen that the results obtained are at par with the previous works that have been done with few instances of minor improvements over the existing works.

5.1 Measurement Condition

The algorithm each time allows for the maximum possibility of connection between the source pin and the target pin. This is done by having up to 5 trials for random selection of different points of extension each time there is no further room for extension (Multiple extension trials). Again if even this fails to provide any further extension, then the channel can undergo a maximum of 16 (chosen on a trial basis) row addition for carrying out connection between a pair of pins. This is done to maximize the chances of row addition at the exact position required to facilitate further extension. The size of the initial population and the number of crossovers are being experimented with right now and further study is being carried out for determining the size based on the complexity of the channel routing problem. For the results presented here, the initial population size has been less than and equal to 50 in each case. And the number of crossovers

has been 30. The results presented are for a maximum of 30 generations only with further experimentation being done for larger number of generations.

5.2 Results: Randomly Routed Individuals

A number of solutions for certain benchmark channel routing problem were found using random routing procedure to test the efficiency of the algorithm. While creating each individual it has been made sure that total randomness in the creation of channel is maintained so that the population of individuals generated has good diversity. This is one of the essential requirements for an efficient implementation of Genetic Algorithm.

The channels obtained by random routing procedure had the following results for different benchmark channels.

1. **Joo6_16**: Average no. of rows in population: 12.9788

   avg. no.of vias=21.2850

   Avg. link length=212.3125

   Avg time/ channel creation=37 sec

2. **Burstein's difficult channel:** Average no. of rows in population: 8.6200

   Avg. no.of vias=15.5600

   Avg. link length=156.43

   Avg time/ channel creation= 10.9373 sec

5.3 Best Randomly Routed Individuals Obtained For Benchmark Channels

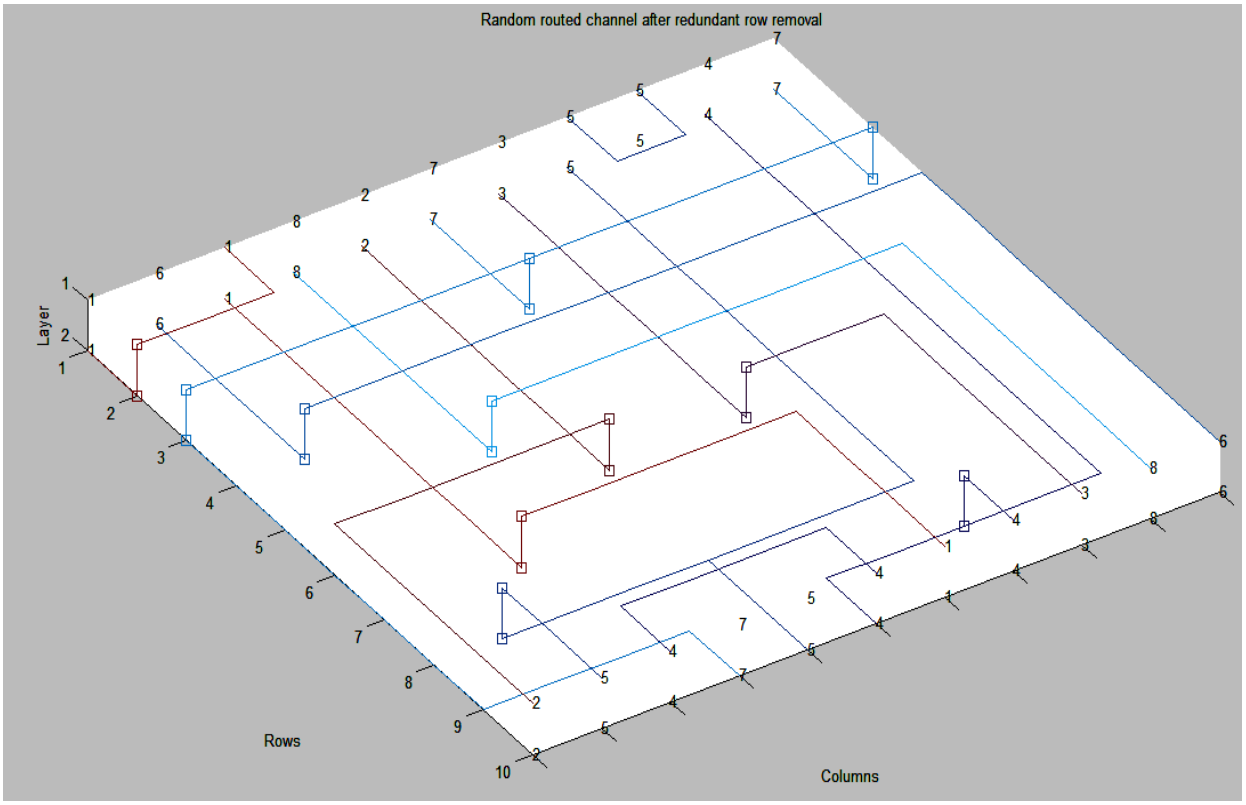


Fig: 5.1 Best Random Routing Solution for Joo6_16 channel

No of rows=8

No of vias=11

Link length=134

5.4 Previous Work Results and Results from Our Initial Random Routing Implementation

A number of individuals were created by random routing itself (without the application of GA) and the best channels so obtained were considered. This was done to test if the offspring channels being created by the algorithm suggested in [1] effectively carried the good characteristics of their parent channels over the generations or not. This was tested as in the crossover procedure in [1], few of the characteristics from both of the parent channel parts were being lost (connections being cut at the selected column are traced back and erased to remove all dangling connections) and random characteristics are being added to compensate for that. In a few cases it was even seen that the best channel from a very large randomly generated initial population (over 300 individuals), gave a better result than the algorithm being proposed in [1]. This leaves a possibility for suggesting a better random routing and crossover technique for channel routing problem where the loss of characteristics during crossover from parent channel to the offspring is minimal.

channel		**Best Previous work**	**Our work (Random Routing)**
Joo6_16	no. of rows	6	8
	no.of vias	15	11
	link length	116	134
Burstein's difficult channel	no. of rows	4	4
	no.of vias	8	6
	link length	82	81

The best Burstein's channel obtained by generating a large initial population at times showed better results than the GA based algorithm suggested in [1].

5.5 RESULTS: GA Algorithm Implementation for Channel Routing

The Channel Routing Optimization was carried out using the Genetic Algorithm method suggested in [1] with minor modifications to increase the efficiency of the algorithm. The performance of the algorithm was then checked on different benchmark channels. The initial results obtained compared favourably with the previous works.

- **Improvement over the previous work**

5.5.1 Efficiency of the Random Routing Algorithm:

Our initial population best channel obtained (for a population of 50 individuals) was much better than that obtained in [1]. Due to the efficiency of random routing algorithm, our algorithm converges much faster towards the best channel than previous similar works. A comparison of the best individual obtained after initial population generation for Burstein's benchmark channel has been given below. A significant difference in the fitness value and the channel width can be seen from the results.

work	Fitness value of the best individual obtained:	Corresponding no. of rows
Lienig-Thulasiraman [1]	0.16	9
Our work	0.25	6

This can be attributed to the following modifications implemented during random routing over the original algorithm in [1]

- **Redundant row removal after row addition for connection of each set of 2 pins:**

While making extensions for possibility of connections between the source and the target pin, if there is no room for further extensions in between, additional rows are added at random positions. However it is seen that such row additions are required only at certain specific positions and hence adding rows randomly causes many unnecessary rows (which wouldn't be a part of the connection between the present 2 pins) to get added to the channel. Now if these unnecessary rows are removed after connection has been made between the present 2 pins, it is found that the packing density of each of the rows increases and also the number of rows required for the channel generation on an average reduces. What we are essentially doing here is we are trying to make the connection between 2 pins while keeping the channel width as minimum as possible. The channel width is only increased (by row addition) when a connection using the existing rows is not possible. This row addition is again done only at the exact position which would facilitate further extensions and hence rows get added only at positions where they are required and all unnecessary rows are done away with.

- **Only one row is selected initially for each channel in random routing procedure:**

Only one row is selected initially for random routing of a channel. And where ever additional rows are required, row addition is carried out along with redundant row removal for the unwanted added rows. This again helps to improve the packing density as for any new horizontal extension to be made, it is first tried to cram the extension in already existing

densely populated rows and the channel width is extended by row addition only when there is no further scope for addition in the existing rows.

- **Multiple extension Trials**

Multiple extension trials: trying for possibility of extension through other randomly selected points and in other layers before going for random row addition. This has been implemented in the code and its giving a solution which is on an average ~1 row lesser than the random row addition technique.

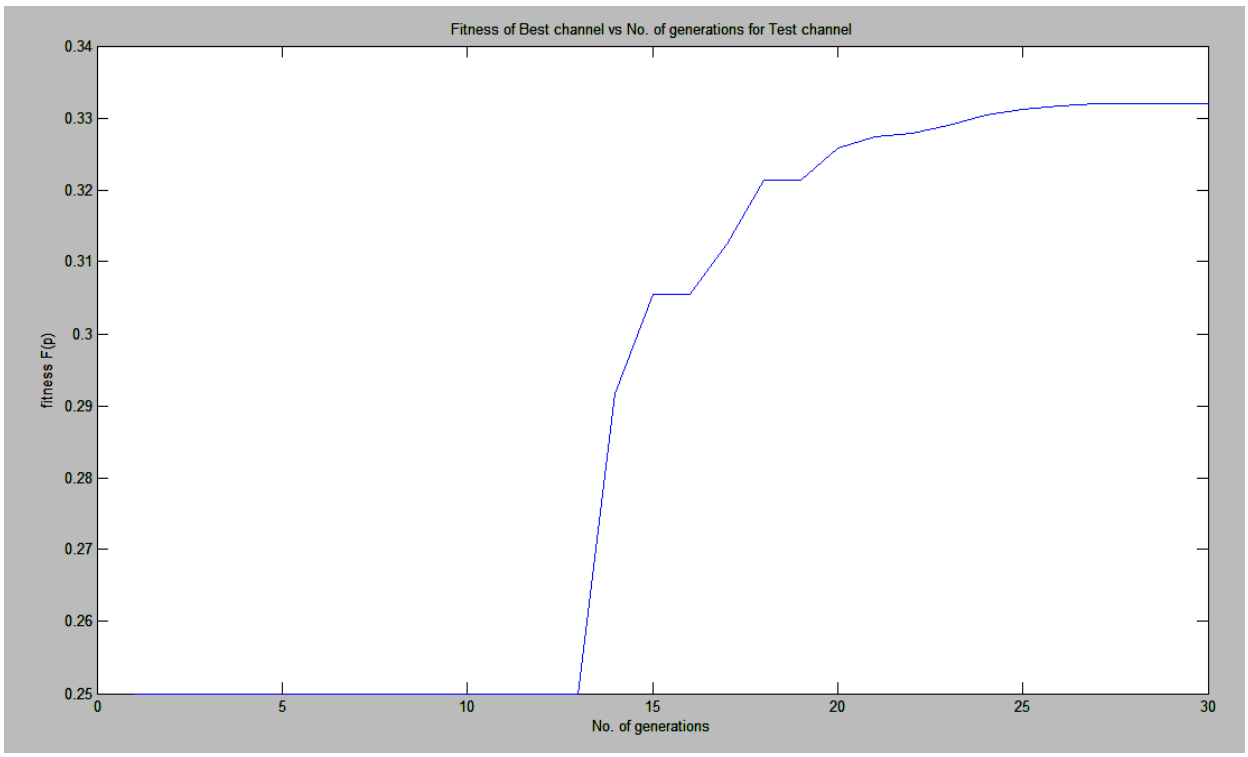


Fig: 5.2 Fitness vs No. of generations plot for the best channel

As it can be seen we got the best result in 30 generations itself with a fitness of 0.3328, which in [1] has taken over 120 generations and has obtained a fitness of around 0.29. This shows that the minor modification brought to [1] have enabled the algorithm to converge much faster towards the best solution.

5.5.2 Efficiency in offspring channel generation:

Before the two parent channels are combined together and routed to produce the offspring channel, the parts of the parent channel alpha and beta being used for the random routing undergo redundant row removal. This way the unnecessary rows (ones which don't have vias, horizontal connections or permanent nodes are removed) are removed. This improves the packaging of the extension and increases the chances of having a greater fitness for the offspring channel produced.

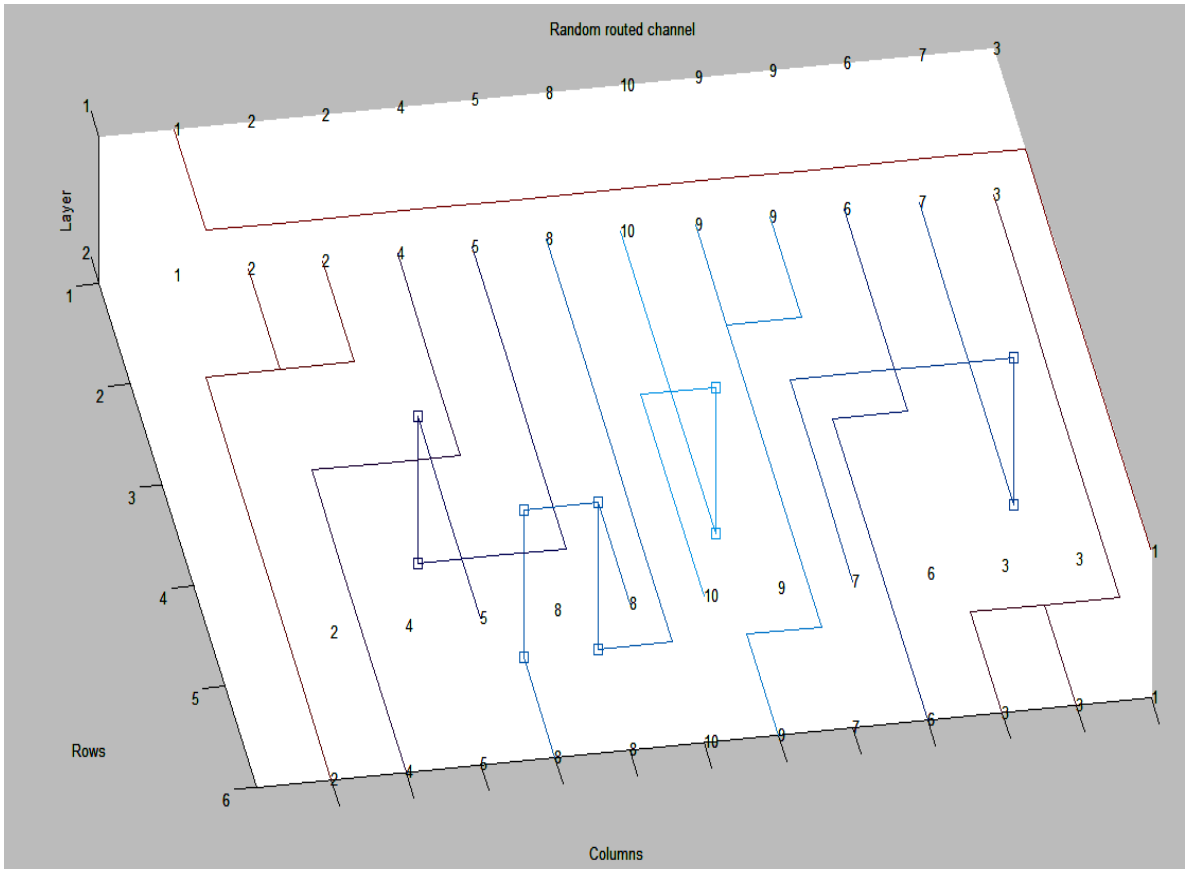


Fig: 5.3 Best Random Routing Solution for Burstein's difficult channel

Benchmark	System	col	rows	Netlength	Vias
Burstein's difficult channel	Lienig-Thulasiraman[1]	12	6	82	8
	Our work	12	6	81	5

As it can be seen, the best routing channel produced by our algorithm for this benchmark channel has fewer number of vias which makes the channel more efficient than the best routing channel obtained in [1]

6. SCOPE FOR IMPROVING OVER THE EXISTING WORK.

1. Loop removal: Loops generated during random routing can be detected and can be removed. By this the link length decreases. Also the presence of such loops can unnecessarily obstruct the extensions being made from other pins which may lead to the insertion of additional rows and hence increase in the channel area. So, by removing the loops formed after each connection between pins with same net, there is a possibility of reducing the channel size of individuals during the random routing procedure.

2. Random selection (for nets, source/target pins. point of extension) should be replaced by probability based selection depending on the beneficiary character brought in by a particular selection at a particular instant. This includes probability on the order of selection of pins, probability in choosing random point of extensions, probability of choosing the target pin for every given source pin, probability of choosing the position for random row addition.

3. Via removal and via compression: Vias appearing in the row adjacent to the top and the bottom row (the rows having the terminals) can at times be removed. Also there is possibility of removal for 2 or more vias appearing along the same extension in some cases.

4. New crossover strategy: this involves selecting each of the net connections from either of the two parent channels. This could be either selection of the better of the two net connections from the two parent channels (i.e. a hybridization process instead of crossover) or random selection of net connections from one of the two channels.

Overlapping can be taken care of by repeated row addition and eventual redundant row removal.

5. Once the GA is implemented and well optimized solutions for various channels are obtained, then different statistics like the choice of point of extension, the selection of source and target pins, the characteristics retained after crossover etc can be noted down to decide upon the probability function for probability based selection required at various instants during the random routing and crossover procedure can be obtained.

6. In the algorithm suggested in [1], the fitness for the whole channel is considered while selection for crossover. However only a part of the channel is used for crossover. Hence fitness doesn't well represent the part being used for the crossover. Hence as an alternative, the fitness can be separately chosen for the left and the right half of the channel. Extrapolating the same idea, implementation of Vector Genetic Algorithm can be thought of which involves the decomposition of the channel into a number of parts and GA is applied independently on each of them before putting all the parts together.

7. FUTURE SCOPE FOR WORK

1. A number of tests based on the improvement in quality of individuals over a number of generations and on the characteristics of the final optimized result are being carried out. Various characteristics of the Best channels so produced like path blocking, row density etc are being studied to find a pattern common to all of them. Based on the observations, other ideas and approaches thought during the course of development of the code will be suitably implemented.

2. Use of Vector Genetic Algorithm: Vector GA is known to give more optimized solution as compared to normal GA in various applications. Hence it is being thought to make modifications in the existing code for GA to apply Vector GA to the Channel routing problem.

3. Use of Particle Swarm Optimization: This is a relatively new optimization Technique modeled on swarm intelligence. The technique is considerable adaptable to variations and hybrids which is seen as a strength over other robust evolutionary optimization mechanisms, such as genetic algorithms. Being relatively new, there is good scope for research in its implementation for Channel Routing Problem.

8. REFERENCES

[1] J. Lienig and K. Thulasiraman, "A Genetic Algorithm for Channel Routing in VLSI Circuits," Evolutionary Computation, MIT Press, Vol. 1, No. 4 (1994): pp. 293-311

[2] J. Lienig, A parallel genetic algorithm for performance-driven VLSI routing, IEEE Trans. Evol. Comput. 1 (1997), pp. 29–39.

[3] Hee Ahn , Seung Kee Han and Tae Won Cho, GenRouter: a genetic algorithm for channel routing problems" Proceeding of TENCON 95, IEEE Region 10 International Conference on Microelectronics and VLSI (November 1995): pp. 151-154

[4] C. Y. Lee, "An Algorithm for Path Connections and its Applications", IRE Transactions on Electronic Computers, vol. EC-10 (September 1961) pp. 346-365

[5] Michael Vai M. ,"VLSI design", Florida: CRC Press; 1 edition (September 19, 2000)

[6] http://www.internetcad.com/thesis/intro.pdf

[7] D. W. Hightower, "A Solution to the Line Routing Problem on a Continuous Plane", Proc. 6th Design Automation Workshop, 1969 pp. 1-24.

[8] By Naveed A. Sherwani, Siddharth Bhingarde, Anand Panyam "Routing in the third dimension: from VLSI chips to MCMs", New York: Wiley-IEEE Press (February 15, 1995)

[9] Andrew B. Kahng, Gabriel Robins "On optimal interconnections for VLSI", Boston, MA: Kluwer Academic Publishers, 1995

[10] TG Szymansky, Dogleg Channel Routing is NP-Complete, IEEE Transactions on Computer- Aided Design of Integrated Circuits and Systems, 4 (1985), 31-41

[11] Middendorf, M., “Manhattan Channel Routing' is NP- Complete Under Truly Restricted Settings”, Vol. 2 (1996), pp. l-19,

[12] Baick, T., Schwefel, H.-P.:” An Overview of Evolutionary Algorithms for Parameter Optimization.” Evolutionary Computation 1 (1993) pp. 1-23

[13] Cohoon J.P., Martin W.N., Richards D.S., “Genetic Algorithms and Punctuated Equilibria in VLSI” Proceedings of Parallel Problem Solving from Nature (PPSN 1),

1st Workshop, 1-3 Oct. 1990, p.134-144

[14] DE Goldberg and JH Holland, "Genetic algorithms and machine learning", Machine Learning, Vol. 3, Nos. 2 3 (1988), pp. 9599

[15] Holland JH, editor. “Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence” 3 rd ed. Cambridge (MA): MIT Press; 1992