A Synopsis on

Study and Design of Algorithms for Optimization in Global Routing for 3D Integrated Circuits

By

Debashri Roy

Registration No. 210811003 of 2011-2012

Examination Roll No. 161108003

Submitted to

Department of Information Technology, Bengal Engineering and Science University, Shibpur in fulfillment of the requirements for the degree of

Master of Engineering
in
Information and Communication Engineering

Under the supervision of

Dr. Prasun Ghosal
Assistant Professor
Department of Information Technology
Bengal Engineering And Science University, Shibpur

Department of Information Technology
Bengal Engineering And Science University, Shibpur
West Bengal, India – 711103

May, 2013

DEPARTMENT OF INFORMATION TECHNOLOGY
BENGAL ENGINEERING AND SCIENCE UNIVERSITY, SHIBPUR
P.O. Botanic Garden, Howrah–711 103

CERTIFICATE OF APPROVAL

This Thesis is approved as a creditable presentation on a study of Computer Science and Information Technology subject titled **Study and Design of Algorithms for Optimization in Global Routing for 3D Integrated Circuits**, carried out and presented satisfactorily to warrant its acceptance as a fulfillment for the degree of *Master of Engineering* in *Information Technology* of the Bengal Engineering And Science University, Shibpur. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement expressed and conclusion drawn therein but approve the Thesis only for the purpose for which it is submitted.

BOARD OF EXAMINERS :

____

____

____

Department of Information Technology
Bengal Engineering And Science University, Shibpur
P.O. Botanic Garden, Howrah–711 103

FORWARD

It is certified that, the thesis entitled **Study and Design of Algorithms for Optimization in Global Routing for 3D Integrated Circuits**, is a record of bona fide work carried out by **Debashri Roy** under my supervision and guidance.

In my opinion, the work for the thesis is satisfactory and it has reached the standard necessary for the submission in the fourth semester of *Master of Engineering* in *Information and Communication Engineering* of the Bengal Engineering And Science University, Shibpur.

(**Dr. Prasun Ghosal**)
Assistant Professor
Department of Information Technology
Bengal Engineering And Science University, Shibpur

Counter signed by:

(**Dr. Santi Prasad Maity**)
Associate Professor and Head
Department of Information Technology
Bengal Engineering And Science University, Shibpur

(**Dr. Partha Protim Chattopadhyay**)
Professor and Dean (FEAT)
Bengal Engineering And Science University, Shibpur

ACKNOWLEDGEMENT

I would hereby like to avail this opportunity to extend my heartfelt gratitude to my project guide Dr. Prasun Ghosal, Assistant Professor, Department of Information Technology, Bengal Engineering and Science University, Shibpur, for his incessant encouragement, sincere supervision, valuable advice and perfect guidance without which it would not have been possible to complete the portion of the project scheduled for the semester and the thesis in time.

I would also like to thank Dr. Santi Prasad Maity, Associate Professor and Head of the Department, Department of Information Technology, Bengal Engineering and Science University, Shibpur, as well as other faculty members and staff members for their support and help, as and when required.

Finally, I cordially thank my fellow classmates of Information Technology, B.E.S.U., my parents and all others who have been attached to this phase of the development of the project directly or otherwise.

Debashri Roy
Registration Number - 210811003
Examination Roll Number - 161108003
M.E. Fourth Semester
Department of Information Technology
Bengal Engineering and Science University
Shibpur

Preface

Progressive growth of VLSI industry critically depends upon the research and development of Physical Design (PD) Automation tools. During last two decades, the research in physical design automation has been very intense, and literally thousands of research articles covering all phases of physical design automation have been published.

During early days, physical design has faced lots of challenges due to rapid changes from time to time. Such change can be viewed as frightening, as long-held assumptions and paradigms, such as Moore's Law, lose relevance. Challenging times are also important opportunities to try new ideas. Three-dimensional IC technology is such a new idea.

The advent of 3D circuits extends the analogy of skyscrapers as ICs are being built upward, with stacks of active devices placed on top of each other. More precisely, unlike conventional 2D IC technologies that employ a single tier with one layer of active devices and several layers of interconnect above this layer, 3D ICs stack multiple tiers above each other. This requires the enhancement of conventional 2 dimensional routing algorithms to the layer to layer approach for the 3 dimensional ICs.

During past few years, as process technologies for 3D have neared maturity and 3D circuits have become a reality, this field has seen a flurry of research effort. The objective of this work is to capture the current state of the art and to provide a comprehensive introduction to VLSI physical design, existing 2D and 3D routing algorithms and 3D ICs architecture. This collection consists of contributions from some of the most prominent research groups in this area, providing detailed insights into the challenges and opportunities of applying global routing for 3D ICs.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

This is the global era of Information and Communication Technology which demands extreme efficiency of storage, transmission, processing and security. Today's world is built on the dynamic systems, sequential circuits and especially the ICs i.e. the Integrated Circuits. Present day VLSI technology permits us to build systems with hundreds of thousands of transistors on a single chip. For example: The Intel 80286 microprocessor has over 10^5 transistors, The 80386 has 275,000 transistors, The 80486 has approximately 10^6 transistors, The RISC processor from National Semiconductor NS32SF641 has over 10^6 transistors, The Pentium processor has over 3×10^6 transistors. The VLSI physical design is now of great concern.

1.1 VLSI Design Steps

The VLSI design is divided into many modular steps [2, 3]. Different steps have different constraints and satisfy various objective functions. The designing procedure in stepwise fashion is stated below.

- Idea to generate a new chip
- Architectural or Functional Design
- Logic Design
- Physical Design
  - Partitioning
  - Floorplanning
  - Routing
    - Global Routing
    - Detailed Routing
  - Compaction,Extraction and Verification
- Fabrication

1.2 Global Routing

In the placement phase the exact locations of the circuit blocks and pins are determined. In the routing phase all the connections are established between the blocks according to the netlist. It is performed after cell placement. Routing is accomplished using computer programs (routers).It consists of precisely defining paths that carry electrical signals run. It takes up almost 30% of design time and a large percentage of layout area. Routing algorithms were first applied to design of PCBs. The main application of automatic routers has been in the automated design of VLSI circuits.

The accepted practice to routing consists of adopting a two-step approach: global routing and the detailed routing [2]. The objective of the global routing is to elaborate a routing plan so that each net is assigned to particular routing regions, while attempting to optimize a given objective function. Then, detailed routing takes each routing region and, for each net, particular tracks within that region are assigned to that net. Global routing is also known as topological routing and loose routing.

Global Routing involves generating a *loose* route for each net. It assigns a list of routing regions to a net without actually specifying the geometrical layout of the wires represented in Figure 1.1.

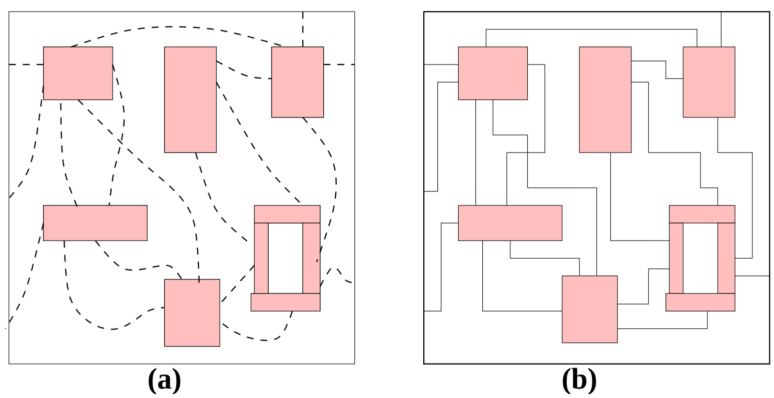


Figure 1.1: Two Phases of Routing (a)Global Routing (b)Detailed Routing

1.3 Problem Definition

A set of constraints and inputs needs to be predefined for formulating the general global routing problem. From that set of given inputs and constraints the favourable path for global routing will be generated as output, after achieving some objective function. The various inputs and constraints are stated below.

Given:

- A set of cells with ports (inputs, outputs, clock, power and ground pins) on the boundaries
- A set of signal nets

- Locations of cells on the layout floor and
- Geometrical constraints and number of routing layers

Goal: find suitable paths on the available layout space that is those paths that minimize the given objective functions, subject to constraints.

Constraints may be:

- Imposed by the designer
- The implementation process or
- Layout strategy and design style

Examples of constraints are:

- Minimum separation between adjacent wire,
- Minimum width of routing wires
- Number of available routing layers
- Timing constraints
- Crosstalk constraints etc.

Examples of objective functions include:

- Reduction of wire length
- Minimization of congestion
- Minimizing crosstalk
- Avoiding obstacle
- Avoidance of timing problems

1.3.1 Cost Functions and Constraints

Several cost functions and constraints may be of different types. The cost functions can be of satisfying a single or multiple objectives. Different objectives may be like

- Reduction of wiring area and congestion
- Improvement in performance
- Improvement in response time
- Improve yield (by reducing cuts)

Geometrical Constraints: Similarly several geometrical constraints will also affect the global routing problem formulation. The several possible such constraints are enlisted below.

- Minimal geometries must be maintained (minimum width and spacing dictated by the technological process).
- Must be able to consider all geometrical constraints abolishing the need for DRC.
- For routing purposes, only those design rules must be considered which define geometries of wires and contact holes.
- Commonly, this is achieved by using a proper equidistant grid.
- Wires are represented by lines and restricted to grid line positions.
- Wire widths and separation between wires is constant for all nets and design rules are avoided.

1.4 3D IC Concept

Much as the development of steel girders suddenly freed skyscrapers to reach beyond the 12-story limit of masonry buildings, same concept for the VLSI chip design leads to the 3 dimensional Integrated Circuits [3].

Three-dimensional integration technologies pack together multiple tiers of active devices, permitting increased levels of integration for a given footprint. The advantages of 3D are numerous and include the potential for reduced interconnect lengths and/or latencies, as well as enhancements in system performance, power, reliability, and portability. However, 3D designs also bring forth notable challenges in areas such as architectural design, thermal management, power delivery, and physical design. To enable the design of 3D systems, it is essential to develop CAD infrastructure that moves from current-day 2D systems to 3D topologies.

One aspect of this is topological, with the addition of a third dimension in which wires can be routed (or can create blockages that prevent the routing of other wires). Strictly speaking, 3D technologies do not allow complete freedom in the third dimension, since the allowable coordinates are restricted to a small number of possibilities, corresponding to the number of 3D tiers. As a result, physical design in this domain is often said to correspond to a 2.5D problem. A second aspect is related to performance issues in general and thermal issues in particular.

1.4.1 Three Dimensional Process Technology Considerations

Both form-factor and performance-scaling trends are driving the need for 3D integration, which is now seeing rapid commercialization. While overall process integration schemes are not yet standardized across the industry, nearly all processes feature key elements such as vertical through-silicon interconnect, aligned bonding, and wafer thinning with backside processing.

Moore′s Law: the number of transistors on a single chip to correspondingly grow at a geometric rate, doubling roughly every 18 months, a trend originally predicted by Gordon Moore and now referred to as Moore′*s* law [4]. The tremendous success of Moore′*s* law encourages the ongoing efforts to continue this trend into the future. However, several serious roadblocks exist.

- The first is the difficulty and expense of continued lithographic scaling, which could make it economically impractical to scale devices beyond a certain pitch.

- The second roadblock is that, even if lithographic scaling can continue, the power dissipated by the transistors will bring clock frequency scaling to a halt.

So in the near future, it will no longer be possible to improve system performance through scaling alone, and that additional methods to achieve the desired enhancement will be needed.

1.4.2 3D Technology: A New Way

Three-dimensional (3D) integration technology offers the promise of being a new way of increasing system performance even in the absence of scaling. This promise is due to a number of characteristic features of 3D integration including

1. Decreased total wiring length, and thus reduced interconnect delay times

2. Dramatically increased number of interconnects between chips and

3. The ability to allow dissimilar materials, process technologies, and functions to be integrated

Overall 3D technology can be broadly defined as any technology that stacks semiconductor elements on top of each other and utilizes vertical, as opposed to peripheral, interconnects between the wafers. Nearly all 3D ICs have three main process components:

1. A vertical interconnect

2. Aligned bonding and

3. Wafer thinning with backside processing

The order of these steps depends on the integration approach chosen, which can depend strongly on the end application.

1.5 Recent Emerging Areas

Global routing for VLSI circuits has received wide attraction now days. The recent researches on global routing in 3D ICs are pointed to optimize a multi objective function. The different objective functions which are being mainly discussed now a day are :

1. Performance and congestion driven

2. Thermal aware based routing like insertion of thermal vias

3. Sensitivity aware routing

4. Minimization of wire length and number of critical paths, which is already a well studied field in 2D routing. For 3D some extra constraints will be added.

5. Crosstalk aware routing to meet RLC crosstalk constraints with consideration of net ordering problem.

6. Obstacle aware routing problem in recent era′s OTC routing. This is being researched with a great interest now a day. This problem area considering non uniform obstacle, is also being focused in this report.

7. For finding the global solution for the global path using genetic algorithm approach. Nowadays, finding solution using genetic algorithm is one of the most researched area.

8. Some thermal driven possibility based or probability based method escaping the deterministic way.

9. Another approach is a placement and global routing in two phases iteratively.

1.6 Fuzzy Logic Concept

Fuzzy logic is a form of many-valued logic or probabilistic logic; it deals with reasoning that is approximate rather than fixed and exact. In contrast with traditional logic they can have varying values, where binary sets have two-valued logic, true or false, fuzzy logic variables may have a truth value that ranges in degree between 0 and 1. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false. Furthermore, when linguistic variables are used, these degrees may be managed by specific functions [5].

1.6.1 Fuzzy Sets

Fuzzy sets as a generalization of classical crisp sets by generalizing the range of the membership function (or characteristic function) from 0, l to a real number in the unit interval [0,1].

1.6.2 Grade of Membership Function for Fuzzy Sets

A fuzzy set, on the other hand, introduces vagueness by eliminating the sharp boundary that divides members from non members in the group. Thus, the transition between full membership and non membership is gradual rather than abrupt. Hence, fuzzy sets may be viewed as an extension and generalization of the basic concepts of crisp sets; however, some theories are unique to the fuzzy set framework. A fuzzy set A in the universe of discourse U can be defined as a set of ordered pairs, $A = \{(x, \mu_A(x)) \| x \in U\}$ Where $\mu_A(.)$ is called the membership function (or characteristic function) of A and $\mu_A(x)$ is the grade (or degree) of membership of x in A, which indicates the degree that x belongs to A.

1.6.3 Fuzzy Relation

Fuzzy binary relations are a generalization of crisp binary relations to allow for various degrees of association between elements. The traditional crisp relation is based on the concept that everything is either related or unrelated. Hence, a crisp relation represents the presence or absence of interactions between the elements of two or more sets. By generalizing this concept to allow for various degrees of interactions between elements, the fuzzy relation is being obtained. Hence, a fuzzy relation is based on the philosophy that everything is related to some extent or unrelated. More specifically, crisp relations and fuzzy relations can be defined in terms of subsets.

1.6.4 Linguistic Variable

While variables in mathematics usually take numerical values, in fuzzy logic applications, the non-numeric linguistic variables are often used to facilitate the expression of rules and facts. Linguistic variable is an important concept in fuzzy logic and approximate reasoning and plays a key role in many of its applications, especially in the realm of fuzzy expert systems and fuzzy logic control. Basically, a linguistic variable is a variable whose values are words or sentences in a natural or artificial language. For example, speed is a linguistic variable if it takes the values such as slow, fast, very fast, and so on. The concept of linguistic variables was introduced by Zadeh to provide a means of approximate characterization of phenomena that are too complex or too ill-defined to be amenable to description in conventional quantitative terms. A linguistic variable is a variable of a higher order than a fuzzy variable, and it takes fuzzy variables as its values. Eg. Speed is a linguistic variable. Slow speed and fast speed are the two fuzzy variable and very slow speed, slow speed, less fast speed, very fast speed all are the example of the fuzzy sets.

1.6.5 Fuzzy Logic

As fuzzy sets are extensions of classical crisp sets, fuzzy logic is an extension of classical two-valued logic. As there is a correspondence between classical crisp sets and classical logic, so is there a correspondence between fuzzy set theory and fuzzy logic. For instance, the *union* operator corresponds to the logic OR, *intersection* to AND, and *complement* to NOT. Furthermore, the degree of an element in a fuzzy set may correspond to the truth value of a proposition in fuzzy logic.

1.6.6 Fuzy Rule Base

Fuzzy rule-based approach to modelling is based on verbally formulated rules overlapped throughout the parameter space. They use numerical interpolation to handle complex non-linear relationships. Many of existing systems need the rules to be formulated by an expert. However rules can be also generated automatically on the basis of numerical data describing a certain phenomenon. In a more explicit form, if there are I rules each with K premises in a system, the i^{th} rule has the form according to equation 1.1.

$$R^i : If\, a_i\, is\, A_{i,1}\, \Theta\, a_2\, is\, A_{i,2}\, \Theta \ldots \Theta\, a_k\, is\, A_{i,k}\, then\, B_i \tag{1.1}$$

In the above equation a_k represents the crisp inputs to the rule and $A_{i,k}$ and B_i are linguistic variables. The operator Θ can be AND, or OR, or XOR.

1.6.7 Fuzzy Logic Control System

During the past decade, fuzzy logic control (FLC), initiated by the pioneering work of Mamdani and Assilian, has emerged as one of the most active and fruitful areas for research in the application of fuzzy set theory, fuzzy logic, and fuzzy reasoning. Its application ranges from industrial process control to medical diagnosis and securities trading.

The basic idea behind FLC is to incorporate the *expert experience* of a human operator in the design of the controller in controlling a process whose input-output relationship is described by a collection of fuzzy control rules (e.g., IF-THEN rules) involving linguistic variables rather than a complicated dynamic model. The typical architecture of a FLC is shown in Figure 1.2, which is comprised of four principal components: a fuzzifier, a fuzzy rule base, an inference engine, and a defuzzifier.

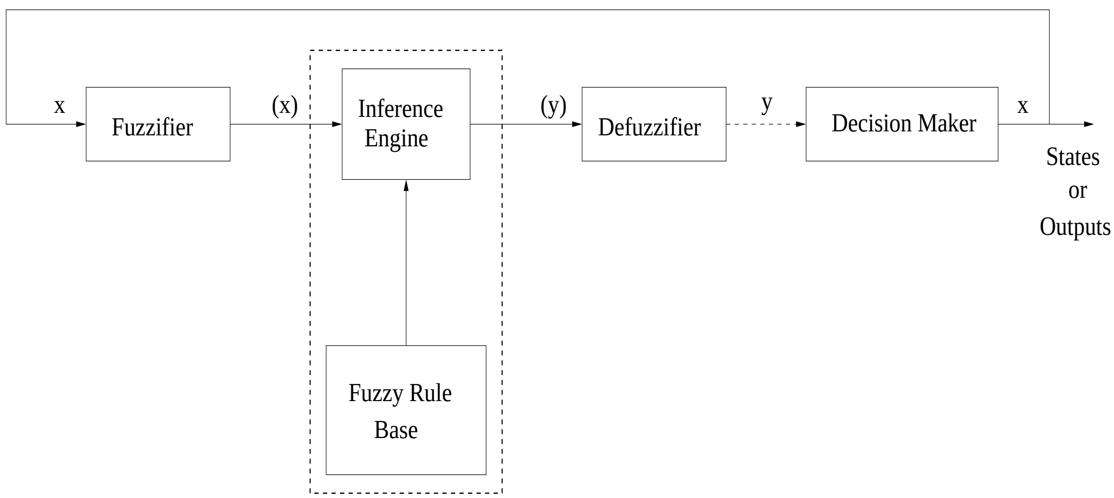


Figure 1.2: Basic architecture of a fuzzy logic controller (FLC)

If the output from the defuzzifier is not a control action for a plant, then the system is a fuzzy logic decision system. The fuzzifier has the effect of transforming crisp measured data (e.g., speed is 10 miles per hour) into suitable linguistic values (i.e., fuzzy sets, for example, speed is too slow). The fuzzy rule base stores the empirical knowledge of the operation of the process of the domain experts. The inference engine is the kernel of a FLC, and it has the capability of simulating human decision making by performing approximate reasoning to achieve a desired control strategy. The defuzzifier is utilized to yield a non-fuzzy decision or control action from an inferred fuzzy control action by the inference engine.

1.6.8 Comments

Fuzzy logic has been and still is, though to a lesser degree, an object of controversy. The several interesting characteristics of fuzzy logic is enlisted below.

1. For the most part, the controversies are rooted in mis-perception, especially a misperception of the relation between fuzzy logic (possibility) and probability theory
2. A source of confusion is that the label *fuzzy logic* is used in two different senses
3. Narrow sense: fuzzy logic is a logical system
4. Wide sense: fuzzy logic is coextensive with fuzzy set theory
5. Today, the label *fuzzy logic(FL)* is used for the most part in its wide sense

Chapter 2

Related Works

2.1 Obstacle Aware Multi Net Routing for 3D ICs: Related Works

In VLSI Routing , Obstacles are a common and inherent consequence caused by several factors like pre-routed nets, macro cells, IP blocks etc. A routing obstacle is an obstacle that causes a dead end with a large void, or a routing trap. In recent era the most important routing approach is Over the Cell (OTC) routing. Here main concern of routing will be only obstacle. So Obstacle Aware Rectilinear Minimum Steiner Tree (RMST) construction is one of the main focused parts of routing in VLSI. An example of Obstacle Aware Routing is shown in Figure 2.1.

In subsequent sections, a brief literature review over obstacle aware RMST construction in 2 dimensional space as well as its extension to 3 dimensional space approach for both single net and multi net, is presented. Also some routing technique by group steiner tree approach and the process of shaping the obstacles. All these topics are very much related to this reported work.

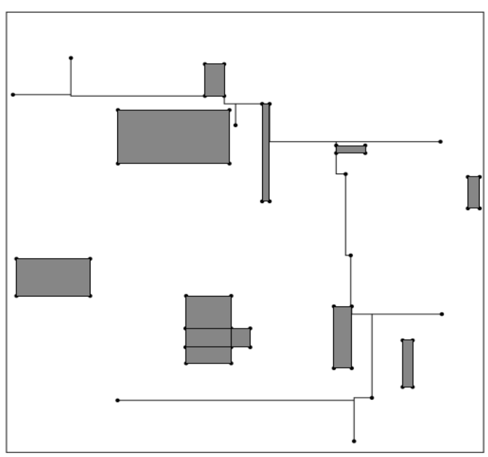


Figure 2.1: An Example of Obstacle Aware routing

2.1.1 Obstacle Aware RMST Construction: 2D Approach

Construction of RMST is a NP Complete Problem [6] and with avoidance of obstacle constraint makes it NP Hard. There are several heuristic based approaches which have tried to perform RMST construction avoiding obstacle polynomial time.

Routing in two-terminal net is a well studied problem. The technique is called maze routing [7]. Then in [8] comes the multi terminal net routing. Here one heuristic for good but non optimal solutions, for five or more terminals have been presented by the concept of escape graph and escape segments (which are a generalization of the line search segments used by Hightower [9]). Next comes [10], where routing of multi terminal net had done in presence of obstacles by using Guillotine-cut technique. A 3 step heuristic FORst [11] divided the total process into terminal partitioning, sub tree construction and separate sub tree connection step. Which gave the time complexity max $(O(n^3), O(n^2 elog(e)))$where n is the number of terminals and e is the number of edges of obstacles. In [12] a rectilinear steiner minimum tree with rectilinear blockages was constructed by connection graph containing O(n) edges and vertices. An O(n log n) time algorithm was proposed to construct spanning graph for RSMTRB. The total approach achieved a solution with significantly reduced wire length with a negligible increase of time. An ant colony optimization approach was taken in [13] for OARSMT problem, named An-OARSMan. A greedy obstacle penalty distance (OP-distance) local heuristic was used in the algorithm and was performed on the track graph. Though the algorithm can handle complex obstacles (both concave and convex) but only with less than 7-terminal net. In another research [14] by OASG(Obstacle Avoidance Spanning Graph) and MTST(Minimum Terminal Spanning Tree) the OARST construction was done. In EBOARST an edge-based global refinement technique as well as a local refinement technique called *segment translation* was adapted. Here 16.56 times speed up can be accrued. In [15] the maze routing based approach can handle large scale (i.e. multiple nets) OARMST problems effectively. Another well accepted and efficient approach [16] , where a top-down partitioning was done . FOARS is one of the best algorithms in terms of both wire length and runtime for test cases both with and without obstacles. In comparison with heuristics, few traditional exact algorithms have been proposed. Li et al. [17] proposed an exact algorithm for multi-terminal nets. Huang et al. [18] presented a two-phase based obstacle avoiding full Steiner trees (OAFSTs) based on [17] by using OAFSTs to construction an OARSMT. The [19] presented a particle swarm optimization in OARMT problem.

2.1.2 Routing for 3D ICs

3D architectures were proposed as an alternative to the classical 2D architectures with certain specific advantages such as reduced interconnect lengths and hence the delay. However negative issues like through-silicon vias (TSVs), excessive heating effects etc also come into play. Routing problem in 3D ICs become even more complicated in presence of obstacle across the routing layers.

In [20] the net and pin distribution problem for global routing is targeting three dimensional packaging layout via System-On-Package (SOP). This was the first work to formulate and solve the multi layer net and pin distribution for layer, wire length and crosstalk minimization. In [21], [22] a physical layout algorithm for 3D SOP designs that includes thermal aware 3D placement and crosstalk aware 3D routing was presented. For single net an efficient algorithm for routing in 3D architecture in presence of obstacle is [23] where the concept of pseudo terminal was introduced for routing between layers. Other different approaches for obstacle aware routing in 3D are [24], [25], [26].

2.1.3 Group Steiner Tree Construction: Graph Theoretic Approach

A group Steiner tree problem is a generalization of Steiner tree problem where several subsets (groups) of vertices in a weighted graph is given. And the goal is to find a minimum-weight connected sub graph containing at least one vertex from each group.

This problem was introduced by Reich et al. [27]. The group steiner tree problem generalizes the set covering problem, and then as hard as this. Then Garg et al. [28] gave a randomized $O(log^3 n\, logk)$- approximation algorithm for the Group Steiner tree problem on an n-node graph, where k is the number of groups. The recent best approximation for Group Steiner tree problem is [29], where for every fixed constant $\varepsilon \rightarrow 0$ the algorithm gives an$O((log \sum_i |g_i|)^{1+\varepsilon} logm)$(where m is the number of groups)approximation in polynomial time. A fault tolerant model of Group Steiner tree problem is presented in [30], where two disjoint path will exist between every two groups.

In this project report one 3 step multi net routing algorithm avoiding obstacles is being presented, where formation of group Steiner tree is one of the step for coarse refinement.

2.1.4 Obstacle Shaping: Computational Geometric Approach

In most of the researches the obstacles are considered as uniform obstacles. But in reality the obstacles are mainly of non uniform size. In this project report the obstacles are taken as non uniform size. So the shaping of non uniform obstacles is necessary for this. Finding the cover of an object in computational geometry is the required technique of interest. In [31] one efficient approach for construction of isothetic covers of a digital object have been proposed. When the cover *tightly* encloses the object, it is said to be an outer isothetic cover. An isothetic cover of a digital object not only specifies a simple representation of the object but also provides approximate information about its structural content and geometric characteristics. But isothetic cover may not totally appropriate for using it in global routing directly. Sometime the more accurate bends may increase the number of vias. But vias are costly. So need to modify this algorithm in a VLSI routing specific manner.

2.2 Fuzzified Approach Towards Intra-layer 2-pin Net Global Routing: Related Works

In modern technology paradigm i.e. in DSM (deep sub-micron) regime, VLSI technology permits us to integrate systems with hundreds of thousands of transistors on a single chip. The recent researches on global routing are aimed to optimize different multi objective functions related to performance and congestion driven routing, thermal aware routing, proper insertion of thermal vias [32], sensitivity aware routing, minimization of wire length and number of critical paths, crosstalk aware routing [33] to meet RLC crosstalk constraints with consideration of net ordering problem, obstacle aware routing problem in recent era′s OTC routing, and for finding the global solution using genetic algorithm approach etc. But to the best of our knowledge, there is no initiation of any fuzzified method for global routing that may be a way out to the problems with deterministic approaches for large scale problems.

Fuzzy logic is a form of multi-valued logic or probabilistic logic, it deals with reasoning that is approximate rather than fixed and exact. In contrast with traditional logic they can have varying values, where binary sets may have two-valued logic, true or false, fuzzy logic variables may have a truth value that ranges in degree between 0 and 1. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false. Furthermore, when linguistic variables are used, these degrees may be managed by specific functions [5]. A fuzzy set [34], on the other hand, introduces vagueness by eliminating the sharp boundary that divides members from non members in the group.

During the past decade, fuzzy logic control [35], initiated by the pioneering work of Mamdani and Assilian [36], has emerged as one of the most active and fruitful areas of research in the application of fuzzy set theory, fuzzy logic, and fuzzy reasoning. Its application ranges from industrial process control to medical diagnosis and securities trading. The basic idea behind FLC is to incorporate the *expert experience* of a human operator in design of controller in controlling a process whose input-output relationship is described by a collection of fuzzy control rules (e.g. IF-THEN rules) involving linguistic variables rather than a complicated dynamic model.

The concept of thermal sensitivity reported in [37] is adopted for this project work. A fuzzy logic expert system for global routing technique has been presented by us to optimize a multi objective function. Sait et. al. had proposed a fuzzy simulated evaluation algorithm for placement in [38]. But as far as our knowledge is concerned, till now there is no such work on fuzzification of routing problem during physical design.

2.3 A Fuzzified Approach Towards Multi-terminal Inter-layer Net Global Routing: Related Works

In high-performance VLSI circuits, the on-chip power densities are playing dominant role due to increased scaling of technology, increasing number of components, frequency and bandwidth. The consumed power is usually converted into dissipated heat, affecting the performance and reliability of a chip. Generation of hot spots is a critical issue in the VLSI physical design phase. Several researches have already been done over thermal aware placement and routing [39–45] for 2 D as well as 3 D integrated circuits. This is an emerging direction of global routing. Pathak et. al. [46] [47] have presented a novel algorithm on 3D Steiner routing by NLP based approach for thermal aware global routing in 3D stacked ICs. The complexity of Steiner tree based approach becomes very high with multi net. All these considerations encouraged me to ponder over the problem from a different aspect that can lessen complexity of deterministic solution approach for global routing.

In [48] there are several techniques of implementing steiner tree for different objective function are described briefly. But till now there a few implementation for 3D Integrated Circuits. Chang et al. in [49] had presented one ILP base inter die routing technique. Similarly in [50] one Integer Programming based approach was proposed optimizing wirelength and via cost without going through a layer assignment phase. In [51] 2D-to-3D mapping was done by the layer assignment which is powered by progressive via or blockage-aware integer linear programming. All these are global routing approaches in three dimensional space. In [52] Das et al. has designed some routing and placement specific tools for 3D ICs. Different benchmarks for global routing available now like ICCAD 2009 [53] and ISPD 2011. But there also the 3 dimensional situation is considered for only metal layers. For fully 3 dimensional concept(i.e. multiple device layers) of 3D ICs there is no such benchmark available till now.

In modern era the design complexity of different problems are increasing in exponential order and size of the problem is so huge that such problems are seemed to be unsolvable in feasible time even by some heuristics also. So taking decision only deterministically for such problems approaches towards NP Completeness. Global routing is also facing such problem in present days. In the proposed approach, a degree of reliability was tried to achieve for each solution of global routing. Since fuzzy systems have already recognised as universal approximator [54], it is used to formulate one pioneering approach as the proposed work. In fuzzified approach the search space is decreased for a particular solution, so the time and design complexity. Then further progress is directed to optimize the design and timing complexity of proposed technique. This is possible only for global routing, for detailed routing it is not so far applicable as fine tuning will be necessary there, which can only be done by deterministic way.

Chapter 3

Obstacle Aware Multi-net Routing in 3D ICs

3.1 Introduction

In this chapter the main concentration is on the obstacle aware routing. The total routing area of a chip have increased in the modern era, by the concept of Over The Cell(OTC) routing. Till there are some obstacles which need to be avoided for routing purpose. So obstacle aware routing is now a catching area of VLSI researchers. The obstacles of non uniform size is considered here. The total algorithm is mainly divided into three step. The main concepts needed for this algorithm are shaping the non uniform obstacles, group steiner tree formation at coarse refinement step and directed steiner tree formation in fine refinement step.

3.2 Problem Formulation

The total problem formulation for global routing can be stated as below. Here problem is formulated to fulfil some objective function satisfying some specific constraints.

3.2.1 Problem Statement

Let $P = \{p_1, p_2, p_3, ..., p_m\}$ *be a set of pins of* m *pin net distributed across* n *layers. Let* $B = \{b_1, b_2, b_3, ..., b_k\}$ *be a set of rectangular obstacles spread over the* n *layers. Let* $V = \{v_1, v_2, v_3, ..., v_l\} = P \cup$ *corners of* B *as the vertex set in the problem, where, each* v_i *has coordinates* (x_i, y_i)*. Each obstacle has at least 4 corners, so here* $l \geq m + 4k$ *. The objective of the problem is to construct a Multi Layered Rectilinear Minimum Steiner Tree that connects all the pins through some extra points (called Steiner points) to achieve a minimal total wire-length, while avoiding the intersection with any obstacle in any layer in the design.*

3.2.2 Problem Description

Concept of Non-uniform Obstacles

Each non-uniform obstacle is actually a collection of a number of uniform obstacles with nearer proximity among them. There will be a certain proximity range with respect to the layout's length or width. If the distance between any two small uniform obstacles falls in that range, then those two obstacles will form a larger obstacle, may be uniform or non-uniform. The larger the size of the obstacle, possibility of being it's shape non-uniform, increases.

Concept of Routing Specific Outer Cover of Obstacles

In [31] A. Biswas et al. proposed an efficient algorithm for finding outer cover of an non uniform shape. And the cover is isothetic which is very much acceptable with the grid structure of a routing layer. For routing the perfect isothetic cover may not be helpful. So for the sake of compatibility of isothetic cover in global routing, a routing specific isothetic cover generation approach, is designed using the specified algorithm. In this approach the unnecessary bends in isothetic cover have been deleted for reason of minimising the use of costly vias. Ignorance of unnecessary bends subsequently results in wastage of routing region. Avoiding a particular bend will sacrifice a minimal routing region in this proposed algorithm, but that is accepted.

3.3 Proposed Solution

3.3.1 Proposed Algorithm for Routing Specific Outer Isothetic Cover Generation for Non-uniform Obstacles

Shaping of non-uniform obstacle is executed after placement and before the starting of global routing. Each non-uniform obstacle can be convex or concave. In Algorithm 1 the procedure *Genrate_Routing_Spec_Isothetic_Cover( )* is generating the routing specific outer isothetic cover of the non-uniform obstacle shape. The Isothetic_Cover() method is producing the minimal outer isothetic cover according to [31]. The corner vertex information of the outer isothetic cover is stored in $IsotheticCoverInfo$. Then width and length of the outer isothetic cover is found. In *Expand_Horizontal()* (*Expand_Vertical()*) procedure expansion of each horizontal (vertical) edge of the isothetic cover horizontally (vertically) in the opposite direction of the obstacle up to the end of the obstacle in that direction is done. Then in procedure *Eliminate_Via()*, if the distance between any two parallel expansion is lesser than $LengthLimit$ ($WidthLimit$), then draw a perpendicular line form the distant (closer) corner point upon the expansion of another one. The extended region is now included in the isothetic cover. At last the procedure *Include_Region( )* is applied on the result of the previous procedure. In this method, if one expansion touches the outer cover then stop expanding the line and compute the distance between the corner point and that touch point. If the distance is lesser than $WidthLimit$ or $LengthLimit$ include the extension into the outer cover. Here minimal outer isothetic cover is being modified according to the routing specification to minimize the number of bending in the obstacle cover. The total procedure described above will be more clearer from the Figure 3.1.

Generate_Routing_Spec_Isothetic_Cover( )

Input : Set of rectangular obstacles($ObstacleInfo = \{O_1, O_2, ...O_n\}$), Proportion ratio ($\alpha$)
Output: Corner points of outer isothetic cover ($RSICoverInfo$)

---

begin
$IsotheticCoverInfo$ = Isothetic_Cover($ObstacleInfo$);
$Width$ = Width($IsotheticCoverInfo$);
$Length$ = Length($IsotheticCoverInfo$);
$HorizontalInfo$ = Expand_Horizontal($IsotheticCoverInfo$);
$VerticalInfo$ = Expand_Vertical($IsotheticCoverInfo$);
$WidthLimit = \alpha \times Width$;
$LengthLimit = \alpha \times Length$;
$VEICoverInfo$ = Eliminate_Via($HorizontalInfo$, $VerticalInfo$, $WidthLimit$, $LengthLimit$);
$RSICoverInfo$ = Include_Region($VEICoverInfo$);
end

Algorithm 1: Generate the Routing Specific Outer Isothetic Cover

3.3.2 Proposed Algorithm for Three Level Intra-layer Multi Net Routing Using Group Steiner Tree Avoiding Non Uniform Obstacles

The algorithm is divided in three main steps. The first part corresponds to obstacle bypassing and group steiner tree generation. In second part the coarse refinement is done by generating steiner trees for each individual clusters. At last the fine refinement is being done by adding the rest of single terminals to the generated overall steiner tree. In Algorithm 2 the procedure *Obstacle_Aware_Multi_Net_Routing( )* is stating the proposed approach in a pseudo code.

Steps

The total procedure is mainly divided in three main steps, as discussed earlier. Here the brief descriptions regarding to those steps are depicted below.

1. *Obstacle Bypassing and Grouping the Terminals:* For bypassing the big non-uniform obstacles first need to find the routing specific outer isothetic cover for the non-uniform obstacles by calling the previously described procedure *Generate_Routing_Spec_Isothetic_Cover()*. Next the corner points information of the outer cover of the non-uniform obstacles as well as the uniform ones are stored in $ObstacleInfo$. After that grouping the terminals and generation of clusters is done in *Generate_Cluster()* method by k means clustering [55] technique. In next stage backbone tree construction is done by the two subsequent procedure *Construct_Group_Steiner_Tree()* and *Construct_Backbone_Tree()*. The group steiner tree generation technique is used as referred in [28], connecting all the clusters in a rectilinear steiner tree considering $ObstacleInfo$.

Obstacle_Aware_Multi_Net_Routing( )

Input : Set of terminals of m pin net($PinInfo = \{p_1, p_2, p_3, ..., p_m\}$), Set of rectangular obstacles($ObstacleInfo$=$\{O_1, O_2, ..., O_n\}$), Placement information of m pins ($PlacementInfo$= $\{pl_1, pl_2, pl_3, ..., pl_m\}$), Proportion ratio ($\alpha$)
Output: Global Routing Path Information($GlbRoutingPathInfo$)

---

begin
  For each non-uniform obstacle $RSICoverInfo$ =
  Generate_Routing_Spec_Isothetic_Cover($ObstacleInfo$, α);
  $ObstacleInfo = RSICoverInfo + UniformObstacleInfo$;
  $ClusterInfo$ = Generate_Cluster($PinInfo$, $PlacementInfo$);
  $GroupSteinerInfo$ = Construct_Group_Steiner_Tree($ClusterInfo$);
  $BackboneTreeInfo$ = Construct_Backbone_Tree($GroupSteinerInfo$);
  for $\forall cluster \in ClusterInfo$ **do**
    $uniformObstacle = UniformObstacleInfo$ for $cluster$;
    $SteinerTreeInfo = SteinerTreeInfo$ + Obstacle_Avoiding_Steiner_Tree(
    $cluster$, $uniformObstacle$);
  end
  $SingleTerminalInfo$ = single or pair of terminals;
  $GlbRoutingPathInfo$ =
  $BackboneTreeInfo + SteinerTreeInfo + SingleTerminalInfo$;
end

Algorithm 2: Obstacle Aware Three Level Multi-net Routing

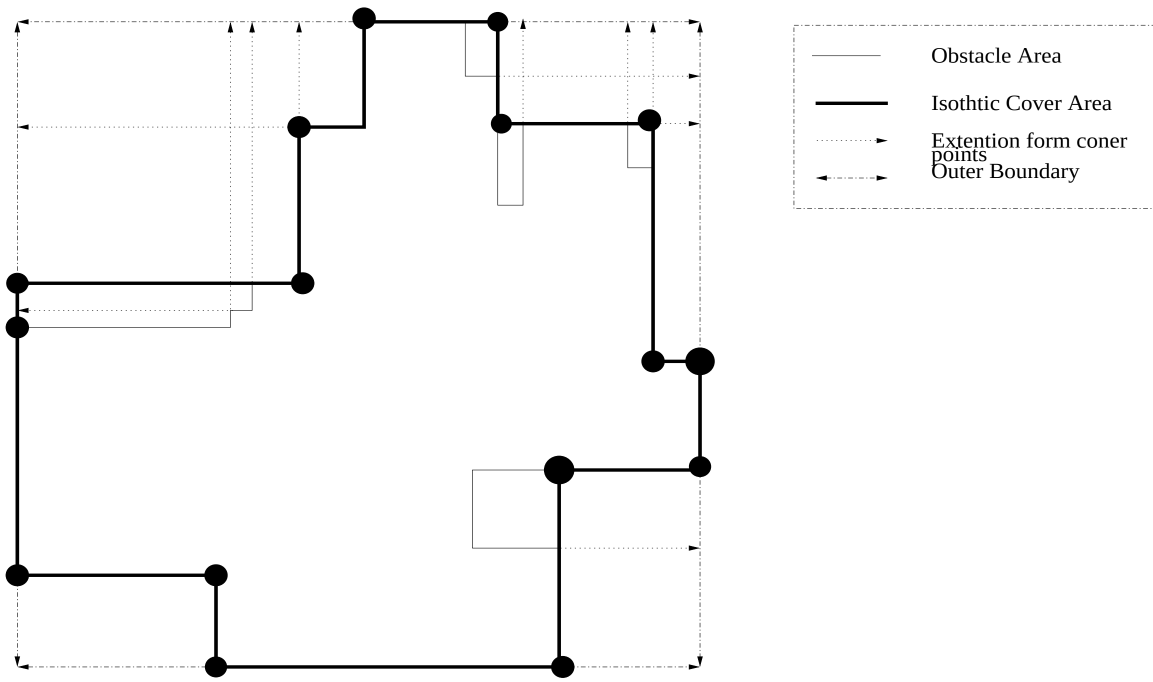


Figure 3.1: Shaping the Non Uniform Obstacle by Routing Specific Isothetic Cover

2. *Coarse Refinement:* This step is repeated for each cluster of a particular net. One cluster contain small uniform obstacles. So here the steiner tree generation will be for small number of terminals near about 3 to 7. In *Obstacle_Avoiding_Steiner_Tree()* procedure the efficient FLUTE [56] algorithm with obstacle avoiding heuristic is used. So here steiner tree generation algorithm is with less time complexity as the terminals are divided into several clusters .

3. *Fine Refinement:* After all this cluster generation. backbone tree construction and group steiner tree generation there may be some single or a pair of terminals left which are till not connected to the backbone tree. Those will be then added to the backbone tree avoiding the obstacles, if exists.

3.4 Implementation

The proposed algorithm has been implemented in Java on a standard desktop environment with an Intel chip running at 2.30 GHz. As the proposed solution is till in development phase so the algorithm have been tested with some test cases defined by us. This algorithm is also applied on ibm benchmarks with some random data of non-uniform obstacles for a single device layer. From the implementation it can be ensured that the proposed algorithm is capable giving the routing path avoiding non-uniform obstacles. For 3 dimensional case till now there is no such benchmark available for fully 3D device layers.

3.5 Conclusion

The proposed approach only includes the solution of the problem formulation in 2 dimensional approach. But its enhancement to three dimensional cases is possible by different ways. All the layers may be projected to a 2 dimensional plane and after routing is done they can be restored in their previous form. Routing among multi layer will be done by some grouping technique. This part and the implementation level is the future plan of this project work.

Chapter 4

An Intra-layer Fuzzified Approach for Two-pin Nets

4.1 Introduction

In DSM (deep sub-micron) regime, together with the integration density interconnects play a dominant role during layout design of integrated circuits. It eventually increases the importance of global routing problem making it more challenging day by day. To cope up with this ever increasing design complexity, the challenging time faced by researchers provides the important opportunity to explore new ideas to solve it within some reasonable time. Heuristic based approaches are generally used for global routing. Large problem space leads global routing problem to a NP Complete one which is less compatible with modern trends. The proposed multi-objective global routing technique is formulated using fuzzy logic to get rid of the limitations of deterministic approaches. After placement and prior to routing phase a set of guiding information is generated from the proposed approach, which will help routing in subsequent steps. During global routing the decision is taken from a fuzzy logic expert system. A GUI is implemented based on the proposed algorithm which is tested for its feasibility study and experimental validation. Success of this proposed approach will open up an avenue for research in global routing phase.

4.2 Problem Formulation

The total problem formulation for global routing can be stated as below. Here problem is formulated to fulfil certain objective function satisfying some specific constraints.

4.2.1 Problem Statement

Let $P = p_1, p_2, p_3, ..., p_m$ be a set of pins of m pin net distributed across the routing layer. Let $M = m_1, m_2, m_3, ..., m_k$ be a set of modules spread over the routing layer, where, each m_i has its bottom left coordinates (x_i, y_i). The sensitivity, congestion and distance factor for each module will either be provided or be determined according to the algorithm and those will be in the range of 0 to 1. The three cost factor coefficients viz. α, β, and γ will also be

Thermal Sensitivity Ratio	Weakly Sensitive(WS) Moderately Sensitive(MS) Highly Sensitive(HS)
Congestion Ratio	Weakly Congested(WC) Moderately Congested(MC) Highly Congested(HC)
Distance Factor	Far Distant(FD) Moderately Distant(MD) Least Distant(LD)

Table 4.1: Derived fuzzy sets from linguistic variables

provided, where $\alpha + \beta + \gamma = 1$. *The weighted cost factor will be generated from the sensitivity, congestion and distance factor information incorporated with the* α, β, *and* γ *values. The objective of the problem is to build a fuzzy logic control system for determining the routing region with minimum wire-length depending upon the weighted cost function maintaining the three constraints.*

4.2.2 Problem Description

Concept of Thermal Sensitivity, Congestion and Distance factor

The total routing layer is represented as a grid structure. The size of the grid can be user defined. The modules are constituted by multiple grids over x and y direction in the routing layer. Each grid have its own sensitivity value. The sensitivity corresponds to the temperature. If the sum of the sensitivity of the grids of an area is very high then that heat dissipation of that area will be very high and that area will be considered as hot spot. There are several already proposed placement technique which can effectively decrease the generation of hot spot. Now during routing avoiding those hot spots is also a great challenge. So the preferable routing region from a source will be a region which have less sensitivity value and less congested and the nearer to target. The congestion increases with the increasing number of pre routed net. So routing in a netlist becomes critical for the latter nets. The thermal window concept from [37] is taken here to divide the total routing layer into different sub regions. The subregions are selected unit for the total algorithm. The routing eligibility is inversely proportional to the thermal sensitivity, congestion and distance factor of each subregion. That is why ineligibility factor is considered as the linguistic variable in consequent part.

Fuzzification of Thermal Sensitivity, Congestion and Distance factor

In fuzzy logic concept thermal sensitivity information(s_r), congestion ratio(o_r) and distance factor (p_r) all are the linguistic variables. All these variables vary within [0,1]. In the premise part the derived nine fuzzy sets from the three linguistic variables are shown in Table 4.1.

Fuzzification of Ineligibility Factor

In the consequent part the linguistic variable is ineligibility criteria (μ_r). Here a number of fuzzy sets may be present. Three rule bases are used in this approach. Different number of fuzzy sets are derived from the consequent part for different rule bases. For original rule base there are total 27 i.e. all possible fuzzy sets($i^{th} I$, where, $i = 1 \ldots 27$) for this particular linguistic variable, is used. Next for first order minimized rule base there are 15 fuzzy sets and for second order minimized rule base there are total 9 fuzzy sets. The used rule bases and its minimization techniques are described in the next section.

4.3 Proposed Solution

4.3.1 Grade of Membership Function

The membership functions for different fuzzy sets of both premise and consequent part are trapezoidal in nature because a moderately sensitive information with higher grade of membership value may also be considered as a highly sensitive information with a lesser grade of membership value. The nature of the membership function is shown in Figure 4.1. The grade

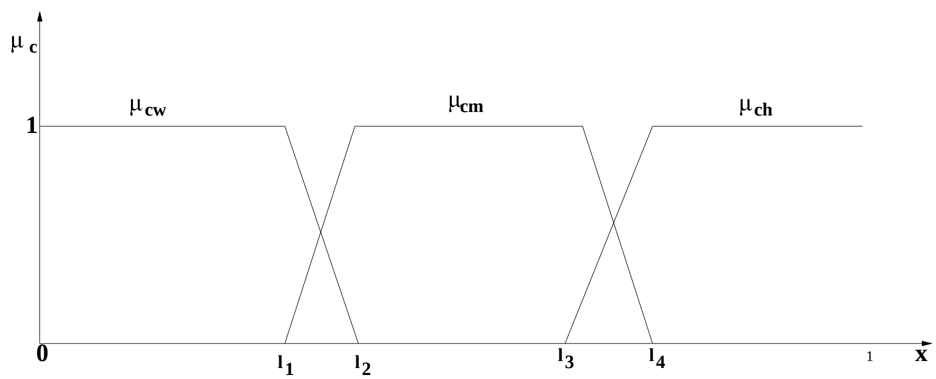


Figure 4.1: The graph corresponding to the grade of membership values for sensitivity, congestion, and distance factor

of membership values for three fuzzy sets corresponding to a linguistic variable of premise part will be according to equation 4.1, 4.2, and 4.3.

$$\begin{aligned} \mu_{ch} &= 0 && for\, x < l_3 \\ &= 1 && for\, l_4 \leq x \leq 1 \\ &= (x - l_3)/(l_4 - l_3) && for\, l_3 < x < l_4 \end{aligned} \tag{4.1}$$

$$\begin{aligned} \mu_{cm} &= 0 && for\, x < l_1 \,\&\, x > l_4 \\ &= 1 && for\, l_2 \leq x \leq l_3 \\ &= (x - l_1)/(l_2 - l_1) && for\, l_1 < x < l_2 \\ &= (l_4 - x)/(l_4 - l_3) && for\, l_3 < x < l_4 \end{aligned} \tag{4.2}$$

$$\mu_{cw} = 0 \qquad for\, x > l_2$$
$$= 1 \qquad for\, 0 \leq x \leq l_1 \tag{4.3}$$
$$= (l_2 - x)/(l_2 - l_1) \qquad for\, l_1 < x < l_2$$

In consequent part the nature of membership function for different fuzzy sets is same as previous mentioned equations. The different boundary values l_1, l_2, l_3, l_4 in equation 4.1, 4.2, and 4.3 for the fuzzy sets of premise part will be generated in the algorithm. Similarly, for the consequent part the boundary values of different fuzzy sets will be determined according to the boundary values of corresponding fuzzy sets of the premise part mainly depending upon the rule base.

4.3.2 Proposed Algorithm for Generation of Guiding Information

My algorithm is divided into two main part distinctly. First part corresponds to generation of relevant information regarding to a specific unit of layout called subregion. This part is executed after placement phase during VLSI physical design. The total procedure works here as guided routing which will be further fed to the fuzzy expert system to take decision during global routing between a source and destination. In Algorithm 3 the procedure *Generate_Guiding_Info()* is generating the guiding information as mentioned. In this algorithm the total layout is divided into number of subregions with help of thermal window concept [37]. The grid size of the layout is scalable and can be controlled by the user. Then for each subregion normalized weighted average sensitivity ratio, normalized congestion ratio and normalized distance factor is generated. Mean and variance for all subregion is determined, which is used for determining the boundary values of membership functions for corresponding fuzzy sets.

4.3.3 Description of Algorithm

Let us suppose, for r^{th} sub region(width W and height H) the total number of module is m. For i^{th} module, height is H_i and width is W_i. So the total area occupied by m modules in r^{th} sub region is $Area_r$ which is formulated in equation 4.4.

$$Area_r = \sum_{i=1}^{m} W_i H_i \tag{4.4}$$

Say for r^{th} sub region sensitivity value of i^{th} module is s_i. Now, the average sensitivity of r^{th} sub region is s_r as stated in 4.5. Then the mean of thermal sensitivity for the total region is s' and variance is v as shown in equation 4.6 and 4.7, where N = total no of sub region. .

$$s_r = \frac{(\sum_{i=1}^{m} W_i H_i s_i)}{W \times H} \tag{4.5}$$

$$s' = \frac{1}{N} \sum_{r=1}^{N} (s_r) \tag{4.6}$$

$$v = \frac{1}{N} \sum_{r=1}^{N} (s_r - s')^2 \tag{4.7}$$

Generate_Guiding_Info( )

Input : Set of sensitivity information for each module ($SInfo$), Set of test modules ($TestModule$), Destination coordinate ($Dest$), Parameter values ($ParamInfo$)
Output: $GeneratedInfo$

---

begin
- $Subregion$ = one subregion;
- SRN = number of $Subregions$ in the layout;
- MD_{max} = Maximum Manhattan Distance between any two $subregions$;
- **for** $i \leq SRN$ **do**
  - $TotalArea = Area\ (Subregion_i)$;
  - $ModulesInSub$ = set of $TestModule \in Subregion_i$;
  - $AreaCovered$ = area covered by $ModulesInSub$ in $Subregion_i$;
  - $WeightedSensitivity$ = area covered by $ModulesInSub \times SInfo$ for each $ModulesInSub$;
  - $SubDis$ = Manhattan Distance between the $Subregion_i$ and $Dest$ subregion ;
  - $CongestionRatio = \frac{AreaCovered}{TotalArea}$;
  - $SensitivityRatio = \frac{WeightedSensitivity}{TotalArea}$;
  - $DistanceFactor = \frac{SubDis}{MD_{max}}$;
  - $IneligibilityFactor = SensitivityRatio \times ParamInfo \rightarrow \alpha + CongestionRatio \times ParamInfo \rightarrow \beta + DistanceFactor \times ParamInfo \rightarrow \gamma$;
  - Add $CongestionRatio$, $SensitivityRatio$, $DistanceFactor$ and $IneligibilityFactor$ to $GeneratedInfo$;
- **end**
- $MeanSRatio$ = mean of sensitivity $\forall\, Subregions$;
- $VarSRatio$ = variance of sensitivity $\forall\, Subregions$;
- $MeanCRatio$ = mean of congestion $\forall\, Subregions$;
- $VarCRatio$ = variance of congestion $\forall\, Subregions$;
- $MeanDRatio$ = mean of distance $\forall\, Subregions$;
- $VarDRatio$ = variance of distance $\forall\, Subregions$;
- Add $MeanSRatio$, $VarSRatio$, $MeanCRatio$, $VarCRatio$, $MeanDRatio$ and $VarDRatio$ to $GeneratedInfo$;

end

Algorithm 3: Algorithm for generation of information required for guidance during routing

The standard deviation for thermal sensitivity of the total region(d) = $\sqrt{v}$. Hence r^{th} sub region will be recognised as Highly Sensitive(HS) or Moderately Sensitive(MS) or Weakly Sensitive(WS). The classification is determined dynamically during execution of routing procedure is as follows.

1. $s_r \geq s' + d$, the sub region is HS
2. $s' - d < s_r < s' + d$, the sub region is MS
3. $0 < s_r \leq s' - d$, the sub region is WS

Next the congestion ratio for r^{th} sub region is o_r as stated in equation 4.8.

$$o_r = \frac{Area_r}{W \times H} \tag{4.8}$$

Then again the recognition of r^{th} sub region as Highly Congested(HC) or Moderately Congested(MC) or Weakly Congested(WC) is determined by the same mathematical procedure i.e. by mean and variance calculation.

The distance factor is determined by normalizing Manhattan distance between r^{th} sub region and the destination subregion with the maximum possible Manhattan distance possible in that layout. Again the same procedure is repeated for determination of its corresponding fuzzy set regarding to distance factor. So before routing is started, a prior information is generated related to each subregion, which will guide the global routing procedure further. The representation of membership functions of sensitivity ratio, congestion ratio and distance factor with dynamically determined boundary values during execution for a particular test case is depicted in Figure 4.2.

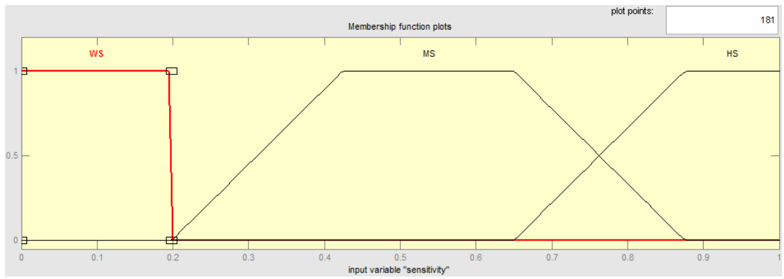
4.3.4 Description of Used Rule Models

Fuzzy logic systems address the imprecision of the input and output variables by defining fuzzy numbers and fuzzy sets. Fuzzy rules are linguistic IF-THEN constructions that have the general form *IF A THEN B* where A and B are (collections of) propositions containing linguistic variables, where A is the antecedent and B is the consequence of the rule. In effect, the use of linguistic variables and fuzzy IF-THEN rules exploits the tolerance for imprecision and uncertainty. In this respect, fuzzy logic mimics the crucial ability of the human mind to summarize data and focus on decision-relevant information.

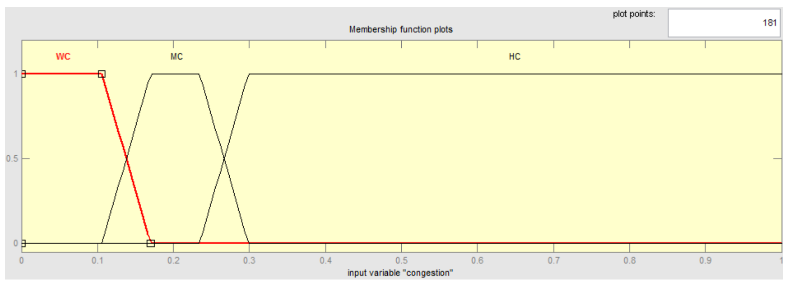
Fuzzy logic control system works according to the fuzzy rule base. Two widely used rule models are: Mamdani [57] and TS model [58]. Where Mamdani [57] is a nonadditive rule model and TS Model [58] is a representation of additive rule model. In Mamdani [57] model is purely fuzzy approach for representation of fuzzy rule base. The rule structure of according to Mamdani model is stated in equation 4.9.

$$R^i : If\, a_i\, is\, A_{i,1}\, \Theta\, a_2\, is\, A_{i,2}\, \Theta \,...\, \Theta\, a_k\, is\, A_{i,k}\, then\, B_i \tag{4.9}$$

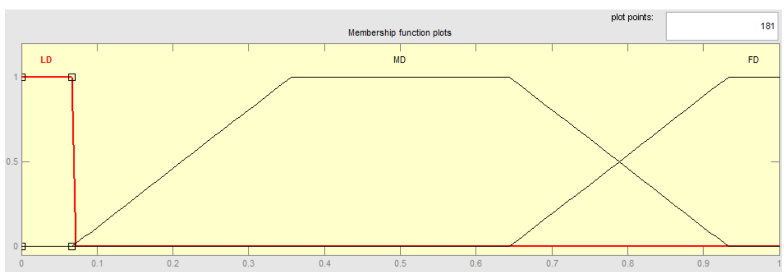
In equation 4.9 a_k represents the crisp inputs to the rule and $A_{i,k}$ and B_i are linguistic variables. The operator Θ can be AND, or OR, or XOR. This model is computationally less efficient but more intuitive with ensuring the better degree of reliability for fully fuzzy applications for comparative simpler problems.



(a) Plot of membership functions for sensitivity ratio



(b) Plot of membership functions for congestion ratio



(c) Plot of membership functions for distance factor

Figure 4.2: Dynamically generated membership functions for linguistic variables in premise part

The main motivation of TS model is to reduce the number of rules required by the Mamdani [57] model. TS model replaces the consequent fuzzy sets of Mamdani rules with function or equations of input variables. It is also used to develop a systematic approach to generate fuzzy rules from a given input-output data set related to complex and high-dimensional problems. The structure of rule in this model is, IF x is A and y is B THEN $z = f(x, y)$, where $f(x, y)$ is a crisp function in consequence. Usually $f(x, y)$ is a polynomial in the input variables x and y, but it can be any function describe the output of the model within the fuzzy region specified by the antecedence of the rule. One example of two-input and one-output first-order TS rule model is stated in equation 4.10, where $f(x, y)$ is a first order polynomial.

$$IF\, x\, is\, A_j\, and\, y\, is\, B_k\, THEN\, z_i \;=\; p \times x + q \times y + r \tag{4.10}$$

The degree the input matches i^{th} rule is typically computed using min operator: $w_i = \min(\mu_{A_j}(x)\ ,\ \mu_{B_k})$. Each rule has a crisp output. Overall output is obtained via weighted average (reduced computation time of defuzzification required than Mamdani model) $\mu_r = \sum_i (w_i z_i) / \sum_i (w_i)$. To further reduce computation, weighted sum may be used, i.e. $\mu_r = \sum_i (w_i z_i)$. The pictorial representation of the working principle of TS model is stated in Figure 4.3.

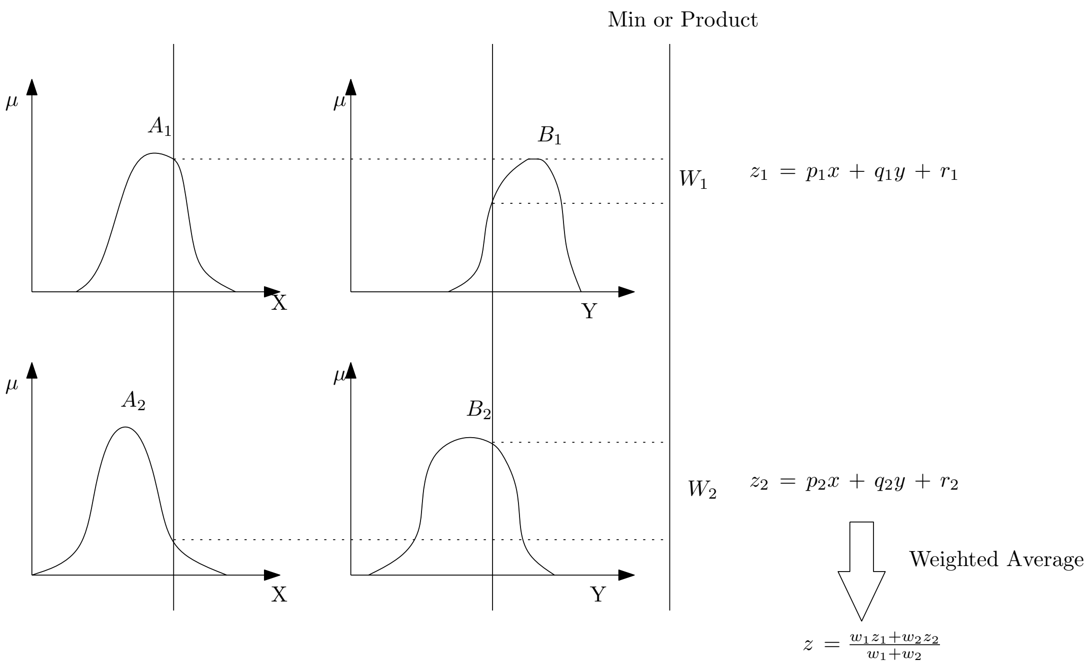


Figure 4.3: Working principle of two-input and one-output first-order TS model

4.3.5 Proposed Rule Base

TS model is computationally efficient and it works well in optimization and adaptive techniques. The consequent part of the rules are not fuzzy so the similarity can be considered only in the premise part of the rules. So the fuzzy sets only in the premise part can be deleted or merged for minimizing the rule base, which is irrelevant in this work. The Mamdani model is computationally less efficient but it is intuitive and well suited for human input. So it has widespread acceptance. In the expert system this model is used to generate rule base for total fuzzification of the process. The ineligibility weight factor is being determined by a mathematical function so here need a conversation between the mathematical function to fuzzy sets. According to the TS model here the structure of rule can be formed like, IF s_r is A_j and o_r is B_k and p_r is C_l THEN $z = f(.)$. The representation of mathematical function $f(.)$ is according to the equation 4.11, where $\alpha + \beta + \gamma = 1$.

$$f(.) = \frac{(\alpha \times s_r + \beta \times o_r + \gamma \times p_r)}{(\alpha + \beta + \gamma)} \tag{4.11}$$

The ineligibility weight factor (μ_r) of the consequent part depends upon sensitivity ratio (s_r), congestion ratio (o_r), and distance factor (p_r) and is calculated as follows.

$$\mu_r = f(\alpha, s_r, \beta, o_r, \gamma, p_r) = \frac{(\alpha \times s_r + \beta \times o_r + \gamma \times p_r)}{(\alpha + \beta + \gamma)} \tag{4.12}$$

where, $\alpha + \beta + \gamma = 1$. The preferable value for α, β, γ will be determined according to the requirement of objective function. Definition of the fuzzy sets for each rule in the consequent part is generated by putting the lower and upper limits for each fuzzy sets of the premise part in above function to determine the lower and upper limits respectively. In this way the proposed original rule base (containing 27 rules) can be represented according to Mamdani Model. The procedure of converting the TS model to Mamdani model is represented in Figure 4.4 and equation 4.13 for better understanding.

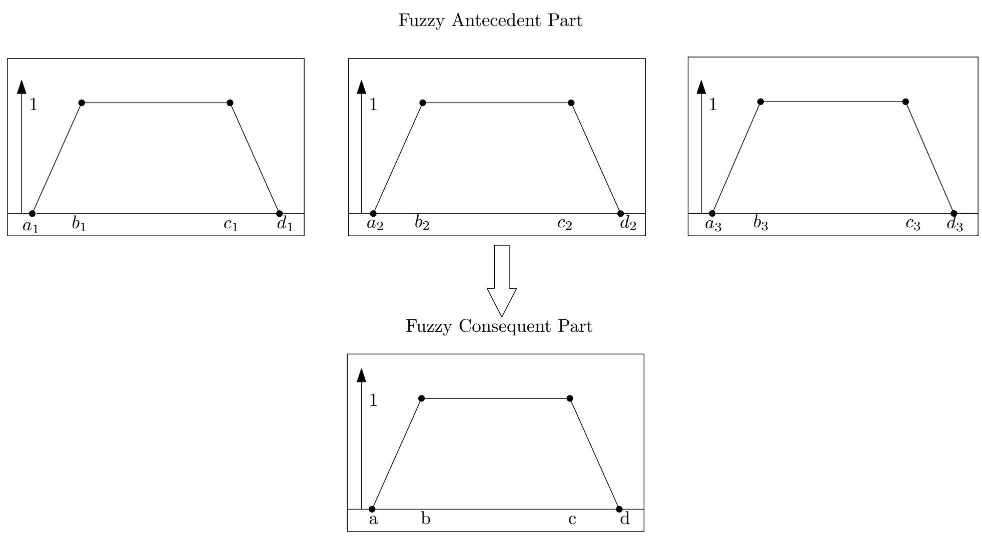


Figure 4.4: Conversion procedure from TS model to Mamdani model

$$
\begin{aligned}
a &= \frac{a_1 \times \alpha + a_2 \times \beta + a_3 \times \gamma}{\alpha + \beta + \gamma} \\
b &= \frac{b_1 \times \alpha + b_2 \times \beta + b_3 \times \gamma}{\alpha + \beta + \gamma} \\
c &= \frac{c_1 \times \alpha + c_2 \times \beta + c3 \times \gamma}{\alpha + \beta + \gamma} \\
d &= \frac{d_1 \times \alpha + d_2 \times \beta + d_3 \times \gamma}{\alpha + \beta + \gamma}
\end{aligned}
\tag{4.13}
$$

In the corresponding implementation three rule bases are considered. The first one is the

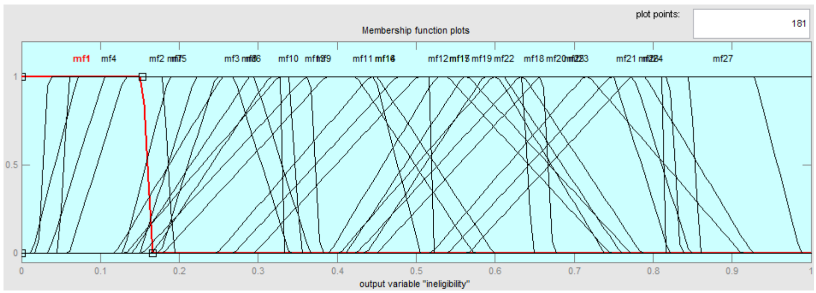


Figure 4.5: Plotted membership functions of 27 fuzzy sets of consequent part in original rule base

original rule base which consists of 27 rules with 27 fuzzy sets corresponding to a linguistic variable of consequent part and 9 fuzzy sets corresponding to three linguistic variables of premise part. The original rule base considers all possible rules with all fuzzy sets of premise part. The membership function′s characteristics of trapezoidal fuzzy sets in the consequent part for original rule base for that same test case(for which the representation of membership functions for premise part are given earlier)is shown in Figure 4.5. The original rule base is stated in Table. 4.2. The other two rule bases are first order and second order minimized version of the original rule base.

4.3.6 Minimizing Rule Base

The concept of minimizing large rule bases came for better timing and design complexity. The minimization concept here works with detecting the similarity between two fuzzy sets of same linguistic variables and correspondingly deleting the rules with detected irrelevant fuzzy sets. Hence some fuzzy sets in the premise or consequent part are either merged together or deleted. If n fuzzy sets are merged together then replace every n fuzzy sets with the new fuzzy set and replace the same in the rule base and for n similar rules $(n-1)$ rules are deleted. In this approach two step of minimization is done. The first order minimized rule base will consists of 15 rules and the second order will be of only 9 rules.

Used Algorithm for Minimizing Rule Base

In this work minimization of the rule base is done only by merging [1], no annihilation or fusion [59] is required. The fuzzy sets are merged according to their similarity measures. The

1.	If x is HS and y is HC and z is FD then w is $1^{st}I$
2.	If x is HS and y is HC and z is MD then w is $2^{nd}I$
3.	If x is HS and y is HC and z is LD then w is $3^{rd}I$
4.	If x is HS and y is MC and z is FD then w is $4^{th}I$
5.	If x is HS and y is MC and z is MD then w is $5^{th}I$
6.	If x is HS and y is MC and z is LD then w is $6^{th}I$
7.	If x is HS and y is WC and z is FD then w is $7^{th}I$
8.	If x is HS and y is WC and z is MD then w is $8^{th}I$
9.	If x is HS and y is WC and z is LD then w is $9^{th}I$
10.	If x is MS and y is HC and z is FD then w is $10^{th}I$
11.	If x is MS and y is HC and z is MD then w is $11^{th}I$
12.	If x is MS and y is HC and z is LD then w is $12^{th}I$
13.	If x is MS and y is MC and z is FD then w is $13^{th}I$
14.	If x is MS and y is MC and z is MD then w is $14^{th}I$
15.	If x is MS and y is MC and z is LD then w is $15^{th}I$
16.	If x is MS and y is WC and z is FD then w is $16^{th}I$
17.	If x is MS and y is WC and z is MD then w is $17^{th}I$
18.	If x is MS and y is WC and z is LD then w is $18^{th}I$
19.	If x is WS and y is HC and z is FD then w is $19^{th}I$
20.	If x is WS and y is HC and z is MD then w is $20^{th}I$
21.	If x is WS and y is HC and z is LD then w is $21^{th}I$
22.	If x is WS and y is MC and z is FD then w is $22^{th}I$
23.	If x is WS and y is MC and z is MD then w is $23^{th}I$
24.	If x is WS and y is MC and z is LD then w is $24^{th}I$
25.	If x is WS and y is WC and z is FD then w is $25^{th}I$
26.	If x is WS and y is WC and z is MD then w is $26^{th}I$
27.	If x is WS and y is WC and z is LD then w is $27^{th}I$

Table 4.2: The proposed original rule base for intra-layer net routing

following similarity measure based on the set-theoretic operations of union and intersection is used to determine the similarity between two fuzzy sets [60].

$$S(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{4.14}$$

where, $|.|$ denotes the cardinality of the set and the $\cap$ and $\cup$ operators represents intersection and union respectively [61]. If S(A,B) is larger than a defined threshold then merging technique is applied. Suppose, fuzzy set A is defined by a parametric membership function $\mu_A(x; a_1, a_2, a_3, a_4), a_1 < a_2 < a_3 < a_4$ according to equation 4.15.

$$\begin{aligned} \mu_A(x; a_1, a_2, a_3, a_4) &= 0,\ x \le a_1\ or\ x \ge a_4 \\ &= 1, a_2 \le x \le a_3 \\ &= \alpha,\ \alpha \in (0,1), otherwise \end{aligned} \tag{4.15}$$

Another fuzzy set B with same type of parametric membership function and similarity measure between them is greater than a predefined threshold. The favourable way of merging is to replace both A and B by the new fuzzy set C. The kernel of C is given by aggregation the parameters describing the kernel of A and B. Thus merging of two fuzzy sets A and B, defined as $\mu_A(x; a_1, a_2, a_3, a_4)$ and $\mu_B(x; b_1, b_2, b_3, b_4)$ respectively, produces a fuzzy set C defined by $\mu_C(x; c_1, c_2, c_3, c_4)$ according to equation 4.16.

$$\begin{aligned} c_1 &= min(a_1, b_1) \\ c_2 &= \lambda_2 a_2 + (1 - \lambda_2) b_2 \\ c_3 &= \lambda_3 a_2 + (1 - \lambda_3) b_3 \\ c_2 &= max(a_4, b_4) \end{aligned} \tag{4.16}$$

In equation 4.16, the parameters λ_2, λ_3 determines which of the fuzzy sets A and B has the most influence on the kernel of C. $\lambda_2 = \lambda_3 = 0.5$ is used here. In the merging technique two

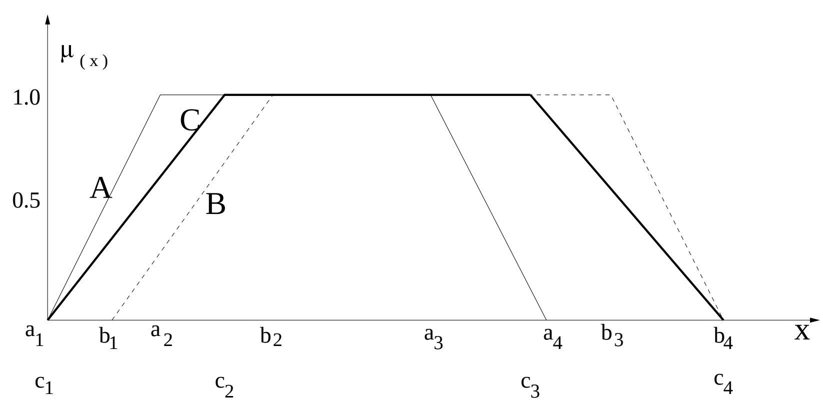


Figure 4.6: Creating fuzzy set C to replace A and B in the rule base according to [1]

fuzzy sets can be merged together to form a new fuzzy set and similarly two rule is merged together resulting a single rule. In the first iteration the first 24 plotted fuzzy sets are merged pair wise resulting 12 fuzzy sets and the last 3 fuzzy sets remain as usual i.e. giving some extra weight to higher ineligibility factors. So total 12+3 = 15 fuzzy sets resulting 15 rules in the corresponding rule base. In the second iteration the first plotted 12 fuzzy sets from first iteration are pair wise merged resulting 6 sets and the 3 remaining higher order fuzzy sets are left as its is. So here total 6+3 = 9 fuzzy sets, resulting total 9 rules in the corresponding rule base. Now three rule base is generated with different types of impact in the implementation.

Original rule base (27 rules) which will have a slow and efficient impact is stated earlier. First order minimized rule base(15 rules) for the previously determined original rule base will make the implementation faster and less efficient for larger problems is shown in Figure 4.7 and membership functions of those fifteen fuzzy sets in consequent part is shown in Figure 4.9. The second order minimized rule base (9 rules) is fastest and least efficient is favoured most for taking decision during routing, is shown in Figure 4.8 and membership functions of those nine fuzzy sets(which are derived from the first order minimised rule base, stated earlier) in consequent part is shown in Figure 4.10. Some basic characteristics of the three proposed rule bases were observed and stated in Observation 1.

1. If (congestion is WC) and (sensitivity is WS) and (distance is LD) then (ineligibility is mf1) (1)
2. If (congestion is WC) and (sensitivity is WS) and (distance is MD) then (ineligibility is mf2) (1)
3. If (congestion is WC) and (sensitivity is WS) and (distance is FD) then (ineligibility is mf3) (1)
4. If (congestion is HC) and (sensitivity is WS) and (distance is MD) then (ineligibility is mf4) (1)
5. If (congestion is WC) and (sensitivity is MS) and (distance is LD) then (ineligibility is mf5) (1)
6. If (congestion is MC) and (sensitivity is MS) and (distance is LD) then (ineligibility is mf6) (1)
7. If (congestion is HC) and (sensitivity is MS) and (distance is LD) then (ineligibility is mf7) (1)
8. If (congestion is MC) and (sensitivity is MS) and (distance is MD) then (ineligibility is mf8) (1)
9. If (congestion is HC) and (sensitivity is MS) and (distance is MD) then (ineligibility is mf9) (1)
10. If (congestion is MC) and (sensitivity is HS) and (distance is LD) then (ineligibility is mf10) (1)
11. If (congestion is HC) and (sensitivity is MS) and (distance is FD) then (ineligibility is mf11) (1)
12. If (congestion is WC) and (sensitivity is HS) and (distance is FD) then (ineligibility is mf12) (1)
13. If (congestion is HC) and (sensitivity is HS) and (distance is MD) then (ineligibility is mf13) (1)
14. If (congestion is MC) and (sensitivity is HS) and (distance is FD) then (ineligibility is mf14) (1)
15. If (congestion is HC) and (sensitivity is HS) and (distance is FD) then (ineligibility is mf15) (1)

Figure 4.7: Generated first order minimized rule base by using fuzzy toolbox in MATLAB

1. If (congestion is WC) and (sensitivity is WS) and (distance is LD) then (ineligibility is mf1) (1)
2. If (congestion is WC) and (sensitivity is WS) and (distance is FD) then (ineligibility is mf2) (1)
3. If (congestion is WC) and (sensitivity is MS) and (distance is LD) then (ineligibility is mf3) (1)
4. If (congestion is HC) and (sensitivity is MS) and (distance is LD) then (ineligibility is mf4) (1)
5. If (congestion is HC) and (sensitivity is MS) and (distance is MD) then (ineligibility is mf5) (1)
6. If (congestion is HC) and (sensitivity is MS) and (distance is FD) then (ineligibility is mf6) (1)
7. If (congestion is HC) and (sensitivity is HS) and (distance is MD) then (ineligibility is mf7) (1)
8. If (congestion is MC) and (sensitivity is HS) and (distance is FD) then (ineligibility is mf8) (1)
9. If (congestion is HC) and (sensitivity is HS) and (distance is FD) then (ineligibility is mf9) (1)

Figure 4.8: Generated second order minimized rule base by using fuzzy toolbox in MATLAB

Observation 1 *The original rule base will achieve a higher degree of reliability with larger time complexity and* 2^{nd} *order minimised rule base will be faster but with degraded degree of reliability.*

4.3.7 Proposed Algorithm for Global Routing

After generation of rule bases a fuzzy expert system is constructed with nine fuzzy sets for premise part and a rule base with corresponding fuzzy sets in consequent part. In Algorithm 5 the function *Fuzzy_Expert_System()* set three modes to get different expert system according

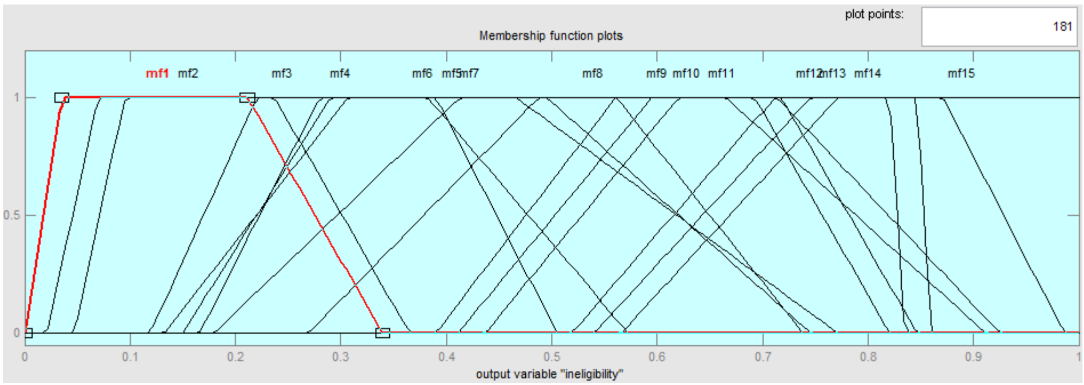


Figure 4.9: Plotted membership functions of 15 fuzzy sets of consequent part in first order minimized rule base

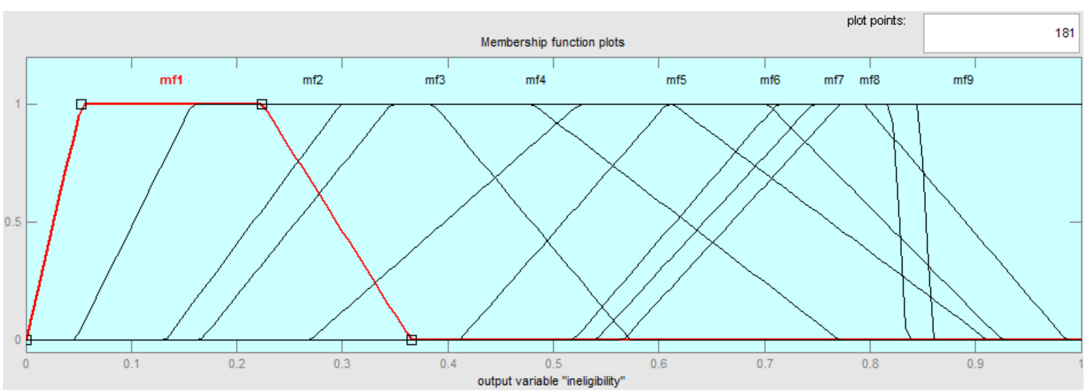


Figure 4.10: Plotted membership functions of 9 fuzzy sets of consequent part in second order minimized rule base

Get_Routing_Path( )

Input : Set of sensitivity information for each module($SInfo$), Set of test modules ($TestModule$), Source coordinate(Src), Destination coordinate($Dest$), Parameter value($ParamInfo$)
Output: Favoured $Subregion$s

```
begin
    MDSrc = Manhattan distance between Src and Dest containing subregions;
    MD = MDSrc;
    TSrc = Src;
    Mode = 0;
    GeneratedInfo = Generate_Guiding_Info (SInfo, TestModule, Dest,
    ParamInfo);
    Fuzzy_Expert_System (GeneratedInfo, Mode);
    while MD > 0 do
        Explore four Neighbours;
          NeighInfo = sensitivity, congestion and distance factor information of each
        Neighbour;
          IneligibilityFactor = Eval_Fuzzy_Expert_System (NeighInfo, Mode);
        Select Neighbour with min IneligibilityFactor;
        TSrc = Neighbour;
        MD = Manhattan distance between TSrc and Dest containing subregion;
        if !Routability then
            Mode = Mode+1;
            MD = MDSrc;
            TSrc = Src;
            Fuzzy_Expert_System (GeneratedInfo, Mode);
        end
    end
end
```

Algorithm 4: Get the sequence of subregions as path of global routing

Fuzzy_Expert_System( )

Input: Mode to select original or minimised rule base($Mode$), Information of premise and consequent part($GeneratedInfo$)

begin
    build membership functions for premise part from $GeneratedInfo$;
    build membership functions for consequent part from $GeneratedInfo$;
    for $OriginalRuleBase$ set $Mode = 2$;
    for $FirstOrderMinimisedRuleBase$ set $Mode = 1$;
    for $SecondOrderMinimizedRuleBase$ set $Mode = 0$;
end

Algorithm 5: Generate the fuzzy expert system with different fuzzy sets and rule bases

Eval_Fuzzy_Expert_System( )

Input : Sensitivity, Congestion and distance factor information of a subregion($SubregInfo$), Mode of evaluation($Mode$)
Output: $IneligibilityFactor$

begin
    if $Mode = 0$ **then**
        $IneligibilityFactor$ = defuzzified value by centroid method using $SecondOrderMinimizedRuleBase$;
    end
    if $Mode = 1$ **then**
        $IneligibilityFactor$ = defuzzified value by centroid method using $FirstOrderMinimisedRuleBase$;
    end
    if $Mode = 2$ **then**
        $IneligibilityFactor$ = defuzzified value by centroid method using $OriginalRuleBase$;
    end
end

Algorithm 6: Evaluate the fuzzy expert system and get a crisp output for crisp input values to the requirement during routing. In Algorithm 4 the global routing approach is actually described. The guiding information and expert system is generated as a prior information before routing. *Get_Routing_Path()* generates the sequence of dedicated subregions for global routing between source and destination. Algorithm iterates over Manhattan distance between source and destination subregion continuously updating the source subregion and stops when the distance become zero. Starting from the source subregion the guiding information for it's four neighbours are passed to function *Eval_Fuzzy_Expert_System()* which determine the ineligibility factors for each neighbours with help of the designed expert system i.e. described in Algorithm 6. The algorithm initially starts with the second order minimized rule base and if rout-ability is not ensured then the expert system is switched to use first order minimized rule base, failing of which leads the use of original rule base which will give result in most cases. If original rule base also can not find the favourable path then the algorithm will suggest for better placement technique.

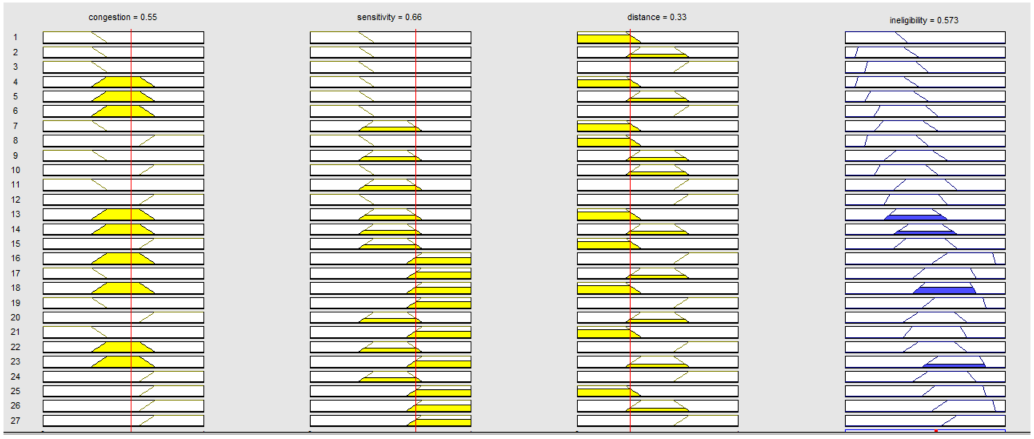
4.4 Experimental Results

4.4.1 Framework

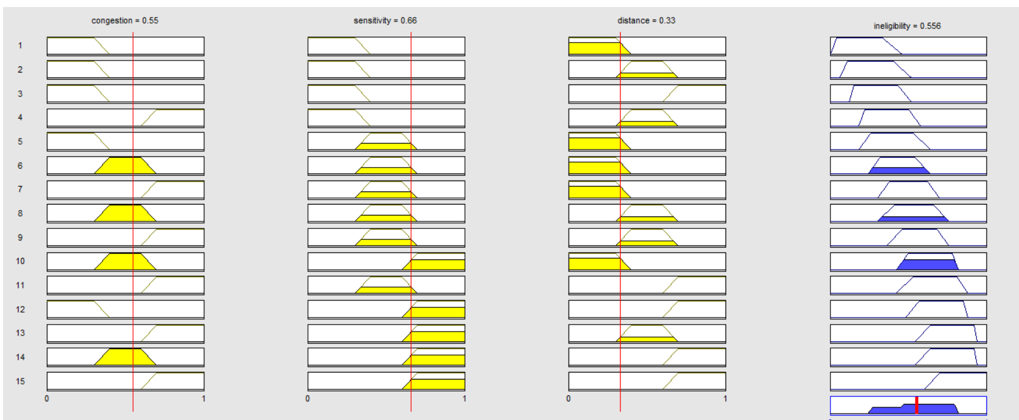
The entire implementation is designed using C, Java, MATLAB 7.14.0, MCR, Matlab Builder JA, Matlab fuzzy toolbox, Swing, and GUIDE. Proposed algorithm has been implemented in two parts. First, a GUI in GUIDE is designed to get a visual graphical path for feasibility study analysis of the pioneering approach. The test cases are defined by us for this case. Next, this technique is enhanced for some general IBM benchmarks using C, Java, MATLAB 7.14.0, MCR, Matlab Builder JA, and Swing. The corresponding fuzzy expert systems are implemented using MATLAB fuzzy toolbox in both the cases. The experiments were performed on a standard desktop environment with an Intel chip running at 2.30 GHz. The five IBM-PLACE 2.0 benchmark suits (for Fixed-die Placement) are used.

4.4.2 Implementation

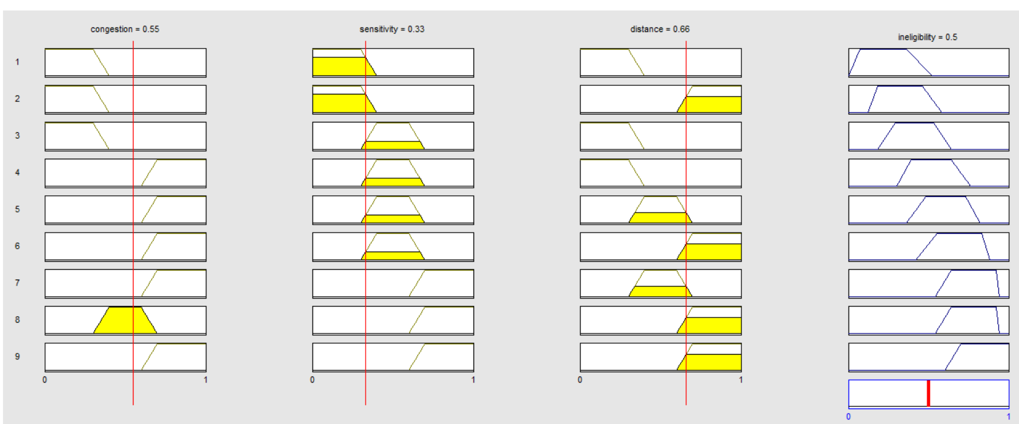
My implemented tool have three modes, i.e. the routing can be of three objective function driven. The three modes are thermal aware, congestion aware and distance aware. The taken α, β, and γ values for the three different modes are depicted in Table.4.3. The Fig.4.11 shows a snapshot of the implemented fuzzy expert system using fuzzy toolbox in MATLAB by selecting thermal mode. Here three different sub-figures shows the three steps with three rule bases. For the same crisp input set [0.55 0.66 0.33] the three steps are producing three different crisp output. The different activated fuzzy sets for premise part by three crisp inputs are the highlighted trapezoids, which fall on the crisp line in first three grids and for consequent part those highlighted trapezoids are on the crisp output line in last grid. From the snap shots of Figure 4.11 it is clear that using original rule base the expert system produce most accurate output.   The surface view of change of ineligibility factor with different input parameters in the fuzzy expert system implemented with second order minimized rule base is shown in Figure 4.12. Here changing of ineligibility factor is very sharp and hedges are quite proper and distinct which means the degree of reliability is less here. For first order minimized rule base it is shown in Figure 4.13. Here changing is lesser sharp and tends to be smooth which achieves some better degree of reliability. Using original rule base the surface view is shown in Figure



(a) Original rule base



(b) First order minimized rule base

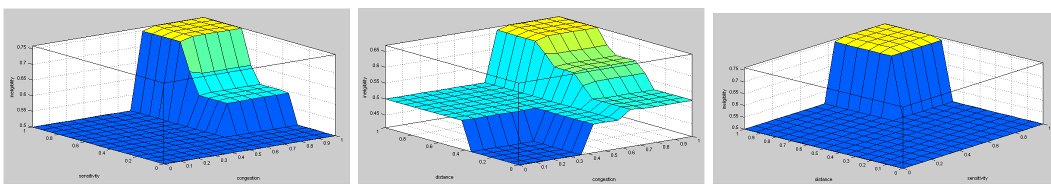


(c) Second order minimized rule base

Figure 4.11: Fuzzification and defuzzification with corresponding rule bases

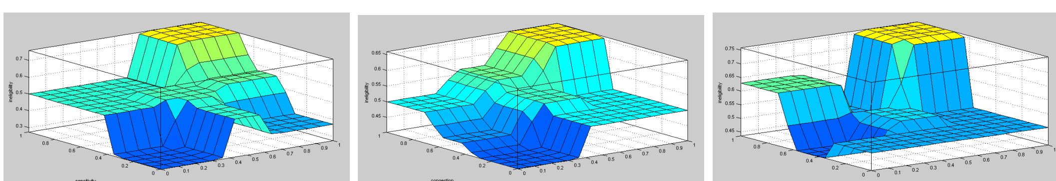
	α	β	γ
Thermal Aware	0.6	0.2	0.2
Congestion Aware	0.2	0.6	0.2
Distance Aware	0.6	0.2	0.6

Table 4.3: Values of three cost factor coefficient for three modes



(a) Ineligibility factor with sensitivity and congestion (b) Ineligibility factor with congestion and distance (c) Ineligibility factor with sensitivity and distance

Figure 4.12: Change of ineligibility factor with sensitivity, congestion and distance for second order minimised rule base



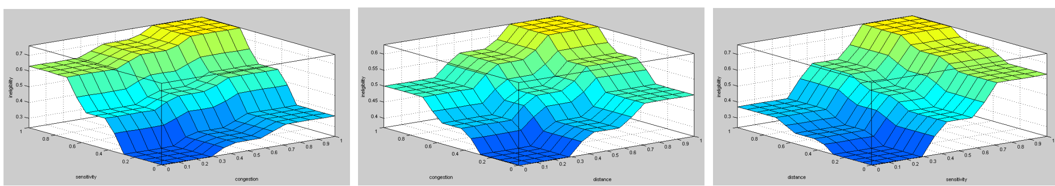
(a) Ineligibility factor with sensitivity and congestion (b) Ineligibility factor with congestion and distance (c) Ineligibility factor with sensitivity and distance

Figure 4.13: Change of ineligibility factor with sensitivity, congestion and distance for first order minimised rule base

Circuit Benchmark	# Nets	# Layer	Vertical Capacity (per layer)	Horizontal Capacity (per layer)	# Intra-layer Two Pin Nets	Routability Achieved (%)	Time Required for Guiding Info Gen(sec)	Time Required for Routing(sec)
ibm01	11507	4	80	80	1453	92.22	87.0	0.0
ibm01	11507	6	80	80	940	94.89	97.0	0.0
ibm02	18429	4	80	80	2514	88.02	157.0	1.0
ibm02	18429	6	80	80	1615	91.64	171.0	0.0
ibm07	44394	4	80	80	6216	79.472	241.0	2.0
ibm08	47944	4	80	80	7151	76.49	259.0	2.0
ibm08	47944	6	80	80	4981	82.89	513.0	1.0
ibm09	50393	6	80	80	4918	79.91	588.0	2.0
ibm09	50393	8	80	80	3643	83.72	249.0	0.0

Table 4.4: Experimental results for two-pin intra-layer nets in IBM benchmark circuits

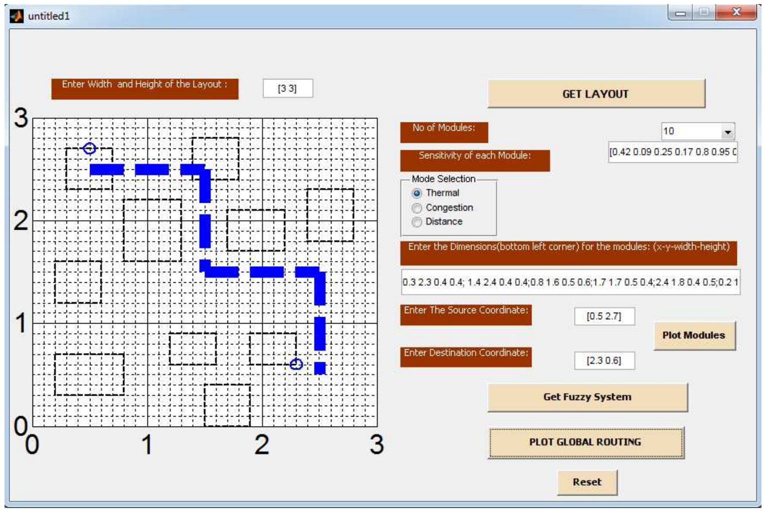
4.14. Here changing of ineligibility factor is quite smooth compared to the two minimized cases so providing the best degree of reliability. The proposed technique is a different conceptual



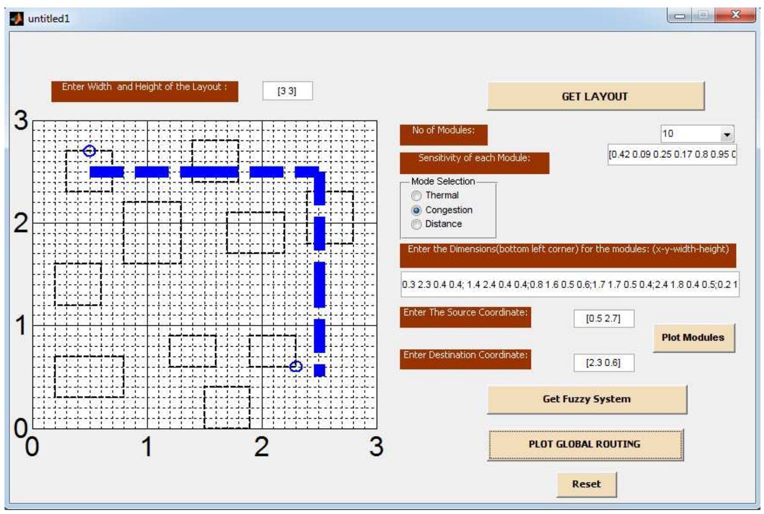
(a) Ineligibility factor with sensitivity and congestion (b) Ineligibility factor with congestion and distance (c) Ineligibility factor with sensitivity and distance

Figure 4.14: Change of ineligibility factor with sensitivity, congestion and distance for original rule base

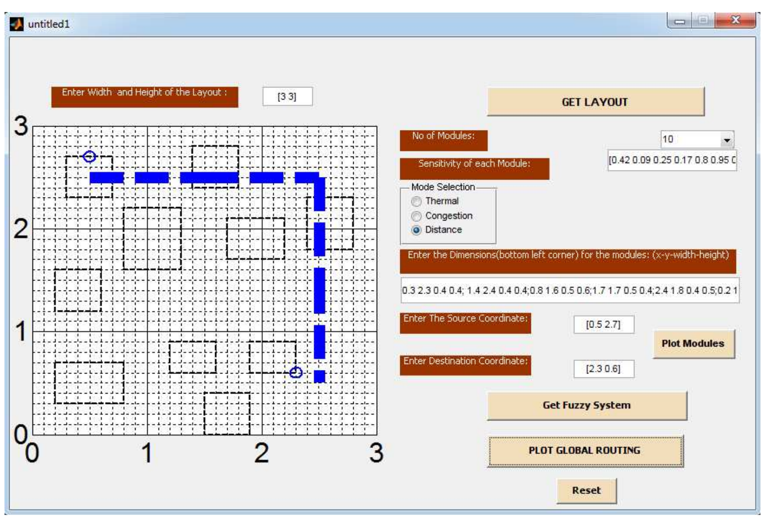
attempt to find a new research area of global routing, so the main concentration only upon the feasibility study of the proposed approach. In the implemented tool one test case is taken for three modes separately and is completely user controlled where user can change each and every parameter according to her specification and this is completely dynamic implementation. The Figure 4.15 shows the snapshots of the tool for the three modes where the routing decision for each sub-region is determined by the implemented fuzzy expert system. In the snapshots the global routing path only selects the favoured sub-regions according to the algorithm not the geometric path. To signify the rout-ability for large circuits, the algorithm is implemented for IBM benchmarks taking two pin nets only. In Table 4.4 several experimental results are stated. The implementation is heading to minimize the time complexity for 3D ICs in fuzzified way. So the placement information have been taken regarding to 3 dimensional space for those benchmarks. The 5_{th} column is showing the percentage of two pin intra-later nets present in the 3D placement for a circuit. From the experimental results it is clear that for routing the negligible time is required. The main concern of this algorithm is the time required for guiding information generation phase. The rout-ability is achieved in range of 80% - 95% , which is quite encouraging result for a pioneering approach. With more number of device layers in 3D placement, the number of intra-layer two pin nets are decreasing but rout-ability is increasing. This characteristic is the beauty of the proposed technique, as it is signifying the concept of 3D ICs. This thesis is aimed at only feasibility study of the proposed approach, not to compare it with other full-proved routers. The implementation is being enriched consequently.



(a) Routing example for thermal aware case



(b) Routing example for congestion aware case



(c) Routing example for distance aware case

Figure 4.15: Snapshots of the designed GUI

4.5 Conclusion

This approach highlights the paramount aspect of global routing problem with proposed pioneering approach using fuzzy logic by achieving a degree of reliability to handle efficiently today's high end design complexity within some comparable time. The generation of guiding information will give $O(m * n)$ time complexity in the proposed method, where m is the number of subregions in x direction and n is the number of subregions in y direction of the layout. The total global routing procedure involves in the same time complexity in addition with the complexity to process fuzzy expert system. The total methodology is proposed and verified successfully for global routing problem in 2 dimensional integrated circuits as well as 3 dimensional one for 2 pin nets only. It may also be considered as a new type of guided global routing approach. Different types of comparative analysis, applications in Steiner tree, multi-pin and critical net routing and enhancement to the implementation for other recent benchmark files may be considered as some possible extensions of this proposed work.

Chapter 5

A Fuzzified Approach for Multi-terminal Inter-layer Net Routing For Standard Cell

5.1 Introduction

In this chapter, the application of fuzzified two-pin intra-layer routing is extended for multi-pin inter-layer routing using the same fuzzy logic approach. In the previous chapter the proposed pioneering technique in the field of global routing in physical design. To escape the boundary limits of deterministic way the possibility based fuzzy logic concept was invented. By using this several applications had found there way to give output in a feasible time and with a better degree of reliability. The implementation of multi-terminal inter-layer routing for standard cells is done by generating a fuzzy expert system and using a fuzzified approach of autometic cluster determination. Success of the implementation has ensured to the enhancement research further for better result and flexibility.

5.2 Problem Formulation

The total problem formulation for global routing can be stated as below. Here problem is formulated to fulfil certain objective function satisfying some specific constraints.

5.2.1 Problem Statement

Let $P = p_1, p_2, p_3, ..., p_m$ be a set of pins of m pin net distributed across multiple routing layers. Let $M = m_1, m_2, m_3, ..., m_k$ be a set of modules spread over the routing layers, where, each m_i has its bottom left coordinates (x_i, y_i). The sensitivity, congestion ratio for each module will either be provided or be determined according to the algorithm and those will be in the range of 0 to 1. The two cost factor coefficients viz. α and β will also be provided, where $\alpha + \beta = 1$. The weighted cost factor will be generated from the sensitivity and congestion information incorporated with the α and β values. The objective of the problem is to generate *a fully 3 dimensional steiner tree determining the global routing path with minimum wire-length depending upon the weighted cost function maintaining the two constraints.*

5.2.2 Problem Description

Concept of Thermal Sensitivity and Congestion Ratio

Here also each routing layer is represented as a grid structure. The size of the grid can be user defined. For standard cell structure the height of each cell is fixed and it is the width which varies for different cells. Each cell have its own sensitivity value. The sensitivity corresponds to the temperature. If the sum of the sensitivity of the grids of an area is very high then that heat dissipation of that area will be very high and that area will be considered as hot spot. There are several already proposed placement technique which can effectively decrease the generation of hot spot. Now during routing avoiding those hot spots is also a great challenge. So the preferable routing region from a source will be a region which have less sensitivity value and less congested . The congestion increases with the increasing number of pre routed net. So routing in a netlist becomes critical for the latter nets. The thermal window concept from [37] is taken here to divide the total routing layer into different sub regions. The subregions are selected unit for the total algorithm. The process of dividing the standard cell layout structure in subregions is shown in Figure 5.1. The routing eligibility is inversely proportional to the thermal sensitivity and congestion ratio of each subregion. That is why ineligibility factor is considered as the linguistic variable in consequent part.

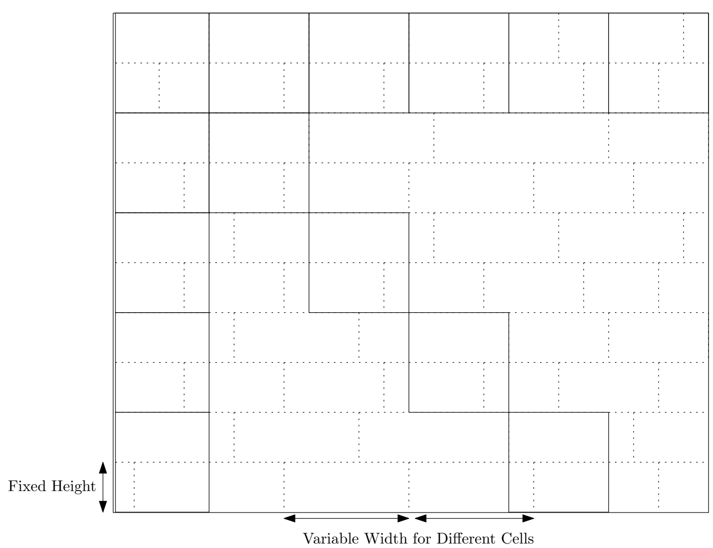


Figure 5.1: Standard cell structure for each layer of a 3D IC

Fuzzification of Thermal Sensitivity and Congestion Ratio

In fuzzy logic concept thermal sensitivity information(s_r) and congestion ratio(o_r) all are the linguistic variables. All these variables vary within [0,1]. In the premise part the derived six fuzzy sets from the two linguistic variables are shown in Table 5.1.

Thermal Sensitivity Ratio	Weakly Sensitive(WS) Moderately Sensitive(MS) Highly Sensitive(HS)
Congestion Ratio	Weakly Congested(WC) Moderately Congested(MC) Highly Congested(HC)

Table 5.1: Derived fuzzy sets from linguistic variables

Fuzzification of Ineligibility Factor

In the consequent part the linguistic variable is ineligibility criteria (μ_r). Here a number of fuzzy sets may be present. One rule base is used in this approach. For the rule base there are total 9 i.e. all possible fuzzy sets($i^{th}I$, where, $i = 1 \dots 9$) for this particular linguistic variable, is used.

5.3 Proposed Solution

5.3.1 Grade of Membership Function

Here also the membership functions for different fuzzy sets of both premise and consequent part are trapezoidal in nature as in the previous chapter. The significance and representation of the membership functions are also same as two pin net connection approach.

5.3.2 Proposed Algorithm for Generation of Guiding Information

My algorithm is divided into two main part distinctly. First part corresponds to generation of relevant information regarding to a specific unit of layout called subregion and generating a fuzzy expert system regarding the connection of a two pins in a single layer. This part is executed after placement phase during VLSI physical design for each layer separately. The procedure works here as guided routing which will be further fed to the fuzzy expert system to take decision during global routing between a source and destination. In Algorithm 7 the procedure *Generate_Guiding_Info()* is generating the guiding information as mentioned.This algorithm is called from *Generate_Routing_Path( )* with a particular layer number. The during the execution of this algorithm the total layout is divided into number of subregions for that particular layer. The grid size of the layout is scalable and can be controlled by the user. Then for each subregion normalized weighted average sensitivity ratio, normalized congestion ratio is generated. Mean and variance for all subregion is determined, which is used for determining the boundary values of membership functions for corresponding fuzzy sets.

5.3.3 Description of Algorithm

Regarding to the standard sell structure the implementation of this algorithm is quite simpler that the previous one. Here the congestion estimation and the total area estimation for each subregion can be got easily from $ObstacleInfo$ and $ParticulatPlacementInfo$ for the

Generate_Guiding_Info( )

Input : Set of sensitivity information for each module ($SInfo$), The total placement information ($PlacementInfo$), Obstacle information ($ObstacleInfo$), Parameter values ($ParamInfo$), Layer Number $layerNo$
Output: Generated guiding information ($GuidingInfo$)

---

begin
$ParticularPlaceInfo = PlacementInfo$ for $layerNo$;
$Width$ = Width($ParticularPlacementInfo$);
$Length$ = Length($ParticularPlacementInfo$);
$NoOfTerminals$ = Total_Terminals($ParticularPlacementInfo$);
$SubregionInfo$ = Get_Subregion_Information($ParticularPlacementInfo$, $Width$, $Length$);
SRN = number of $Subregion$s in the layout from $SubregionInfo$ of layer $layerNo$;
$Subregion$ = one subregion for $SubregionInfo$;
for $i \leq SRN$ **do**
$TotalArea = Area\ (Subregion_i)$;
$ObstacleInSub$ = Get_Obstacle_Per_Subregion($ObstacleInfo$, $Subregion_i$, $layerNo$);
$AreaCovered$ = area covered by $ObstacleInfo$ in $Subregion_i$ for $layerNo$;
$WeightedSensitivity$ = area covered by $ObstacleInSub \times SInfo$ for $layerNo$;
$CongestionRatio = \frac{AreaCovered}{TotalArea}$;
$SensitivityRatio = \frac{WeightedSensitivity}{TotalArea}$;
$IneligibilityFactor =$
$SensitivityRatio \times ParamInfo \rightarrow \alpha + CongestionRatio \times ParamInfo \rightarrow \beta$;
Add $CongestionRatio$, $SensitivityRatio$ and $IneligibilityFactor$ to $GuidingInfo$;
end
$MeanSRatio$ = mean of sensitivity $\forall\, Subregions$;
$VarSRatio$ = variance of sensitivity $\forall\, Subregions$;
$MeanCRatio$ = mean of congestion $\forall\, Subregions$;
$VarCRatio$ = variance of congestion $\forall\, Subregions$;
Add $MeanSRatio$, $VarSRatio$, $MeanCRatio$, $VarCRatio$ to $GuidingInfo$;
end

Algorithm 7: Algorithm for generation of information required for guidance during routing for multi-pin net specific layer. After the subregion structure information is got in $GubregionInfo$ the average $CongestionRatio(o_r)$ and $SensitivityRatio(s_r)$ is determined for each $Subregion$. Next the mean(s') variance is (v_s) of thermal sensitivity for the particular layer is determined by the equation 5.1 and 5.2, where N = total no of sub region and s_r is the average sensitivity information of r^{th} subregion.

$$s' = \frac{1}{N}\sum_{r=1}^{N}(s_r) \tag{5.1}$$

$$v_s = \frac{1}{N}\sum_{r=1}^{N}(s_r - s')^2 \tag{5.2}$$

The standard deviation for thermal sensitivity of the total region(d_s) = $\sqrt{v_s}$. Hence r^{th} sub region will be recognised as Highly Sensitive(HS) or Moderately Sensitive(MS) or Weakly Sensitive(WS). The boundary values for the classification is as follows.

1. $s_r \geq s' + d_s$, the sub region is HS
2. $s' - d_s < s_r < s' + d_s$, the sub region is MS
3. $0 < s_r \leq s' - d_s$, the sub region is WS

Consequently the mean(o') and variance(v_o of) of congestion ratio information is stated in equation 5.3 and 5.4, where N = total no of sub region and o_r is the average congestion information of r^{th} subregion.

$$o' = \frac{1}{N}\sum_{r=1}^{N}(o_r) \tag{5.3}$$

$$v_o = \frac{1}{N}\sum_{r=1}^{N}(o_r - o')^2 \tag{5.4}$$

Similarly the standard deviation for congestion information of the total region(d_o) = $\sqrt{v_o}$. Hence r^{th} sub region will be recognised as Highly Congested(HC) or Moderately Congested(MC) or Weakly Congested(WC). Next the classification is also same as the previous case.

1. $o_r \geq o' + d_o$, the sub region is HS
2. $o' - d_o < o_r < o' + d_o$, the sub region is MS
3. $0 < o_r \leq o' - d_o$, the sub region is WS

So before routing is started, a prior information is generated related to each subregion, which will guide the global routing procedure further. And the total guiding information in dynamic in nature, i.e. the specified boundary values will be determined during the execution of the routing procedure.

5.3.4 Proposed Rule Base

In the part of implementation only one rule base is considered. As described in previous chapter proposed rule base follows Mamdani [57] model, though it is best fitted to TS model [58] . Here

1.	If x is HS and y is HC then w is $1^{st}I$
2.	If x is HS and y is MC then w is $2^{th}I$
3.	If x is HS and y is WC then w is $3^{th}I$
4.	If x is MS and y is HC then w is $4^{th}I$
5.	If x is MS and y is MC then w is $5^{th}I$
6.	If x is MS and y is WC then w is $6^{th}I$
7.	If x is WS and y is HC then w is $7^{th}I$
8.	If x is WS and y is MC then w is $8^{th}I$
9.	If x is WS and y is WC then w is $9^{th}I$

Table 5.2: The proposed original rule base for inter-layer net routing

the working function $(f(.))$ according to TS model is stated in equation 5.5, where $\alpha + \beta = 1$. And the rule structure is IF s_r is A_j and o_r is B_k then $z = f(.)$.

$$f(.) = \frac{(\alpha \times s_r + \beta \times o_r)}{(\alpha + \beta)} \tag{5.5}$$

The conversion between TS model to Mamdani model is also same as describe in previous chapter, which is as follows,

$$\mu_r = f(\alpha, s_r, \beta, o_r) = \frac{(\alpha \times s_r + \beta \times o_r)}{(\alpha + \beta)} \tag{5.6}$$

where, $\alpha+\beta = 1$. The preferable value for α, β will be determined according to the requirement of objective function. Definition of the fuzzy sets for each rule in the consequent part is generated by putting the lower and upper limits for each fuzzy sets of the premise part in above function to determine the lower and upper limits respectively.In this way the rule base consists of 9 rules with 9 fuzzy sets corresponding to a linguistic variable of consequent part and 6 fuzzy sets corresponding to two linguistic variables of premise part. The rule base considers all possible rules with all fuzzy sets of premise part. The membership function's characteristics of trapezoidal fuzzy sets in the consequent part for the rule base is shown in Figure 5.2. The original rule base is stated in Table 5.2.

5.3.5 Proposed Fuzzy Expert System

In Algorithm 8 the *Fuzzy_Expert_System( )* procedure will construct different fuzzy expert systems for different layers with same proposed rule base but different membership functions. Where the fuzzy sets for antecedent part and consequent part will be got from the generated guiding information. This fuzzy expert system will produce a crisp output for the consequent part depending upon the inserted values in the antecedent part and parameter values. The total fuzziifcation and defuzzification is done inside the expert system according to the membership function and the proposed rule base.

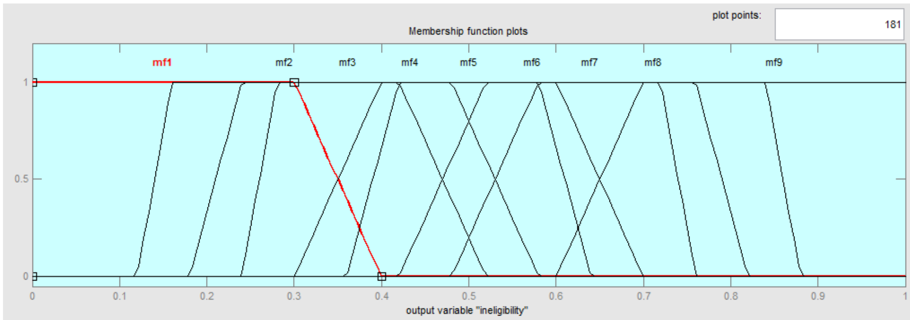


Figure 5.2: Plotted membership functions of 9 fuzzy sets of consequent part in rule base

Fuzzy_Expert_System( )

Input: Information of premise and consequent part($GeneratedInfo$)

---

begin

  build membership functions for premise part from $GeneratedInfo$;
  build membership functions for consequent part from $GeneratedInfo$;
  build the $RuleBase$ with 9 rules;

end

Algorithm 8: Generate the fuzzy expert system with different fuzzy sets for helping in inter-layer multi-pin net connection

5.3.6 Proposed Fuzzified Approach for Intra-layer Net using Fuzzy Expert System

In the Algorithm 9 the *Fuzzified_Intra_Layer_Routing( )* procedure will generate the favoured subregions($Subregion$) for the connection between two pins and a boolean value($Routability$) which will tell whether the destination is reachable or not from the source. When destination is a set of terminals($ClusterInfo$) then this algorithm finds favoured path for each terminals belong to the destination set from the specific source. Here decision making is done using the generated fuzzy expert systems depending upon different layers. The details about this procedure was elaborated in the previous chapter.

Fuzzified_Intra_Layer_Routing( )

Input : Source Coordinate(Src),Destination Information($DestInfo$), Guiding Information($GuidingInfo$), Parameter value($ParamInfo$)
Output: Favoured $Subregions$, Rout-ability ensured or not($Routability$)

---

begin
    $MDSrc$ = Manhattan distance between Src and $Dest$ containing subregions;
    $MD = MDSrc$;
    $TSrc = Src$;
    $Routability = TRUE$;
    Fuzzy_Expert_System ($GuidingInfo$);
    while $MD > 0$ **do**
        Explore four $Neighbours$;
          $NeighInfo$ = sensitivity, congestion ratio information of each $Neighbour$;
          $IneligibilityFactor$ = defuzzified value by centroid method using $RuleBase$;
        Select $Neighbour$ with min $IneligibilityFactor$;
        $TSrc = Neighbour$;
        MD = Manhattan distance between $TSrc$ and $Dest$ containing subregion;
    end
    if *!Rout-ability* **then**
        $Routability = FALSE$;
    end
end

Algorithm 9: Get the sequence of subregions as path of global routing for a intra-layer connection

5.3.7 Used Technique for 2D to 3D Placement

In the modern technology also, the concept of implementing full 3 dimension in case of Integrated Chips is till not achieved. The benchmarks available for 3D ICs are only for multiple metal layers, any benchmark is still not generated for multiple device layers. Due to the unavailability of the input benchmark , leads me to use some 2D to 3D placer tool for the feasibility study of proposed approach. There are already some efficient tool like 3-D MAGIC and PR3D [52] by designed by MIT researchers. There are also different other placer tools. But due to the unavailability and absence of matching with the approach the output of one thermal aware 3D placement algorithm which is stated in [42], [43], is being used here. In this approach one heuristics was stated for the conversion of 2 dimensional node information to the 3 dimensional placement information.

5.3.8 Proposed Algorithm for Global Routing : The Three Step Approach for Multi-terminal Inter-layer Net Routing

Coming to the second part of the algorithm the procedure *Generate_Routing_Path( )* generates the total routing path in terms of sub regions and the total percentage of rout-ability for a netlist of the same circuit in Algorithm 10. It takes the output of 2D to 3D placer tool for a circuit as it's input and produce rout-ability percentage as output for the netlist of the same circuit. The three main steps of proposed algorithm is stated as pseudo code in Algorithm10 and also described lucidly after that. At the starting of the algorithm the generation of subregion information and guiding information is got by the two functions Get_Subregion_Information() and *Generate_Guiding_Info()*. Then it heads towards the routing procedure. The three steps defined below will be called repeatedly for each net having inter-layer connection.

Automatic Cluster Determination and Connection within Clusters

This step is mainly required to process a large net to a smaller one, i.e. by making clusters of terminals with nearer proximity will give the larger net a smaller look. Which is actually required when the number of nodes to be connected in the netlist reaches millionth order. For a inter-layer net if there are a large number of nodes in a single or pair of layers then for those layers one clustering technique is applied. As the total procedure is fuzzified and aimed at to escape the bound of deterministic approach, here one fuzzy clustering approach was used. For better degree of reliability the automatic number of fuzzy cluster determination technique using Simulated Annealing proposed in [62], is being selected. The center of the generated cluster the work as pseudo-terminal for that layout and can be used further for routing. Now for each terminal(t) of l^{th} layer, need to calculate it's residing subregion(r) and its combined sensitivity-congestion ratio(sc_{rl}) from generated guiding information. The combined value will be according to equation 5.7, where the α and β are coming from $ParamInfo$. For the implementation the selected values are, α as 0.4 and β as 0.6 for congestion aware routing, and α as 0.6 and β as 0.4 for thermal aware routing.

$$sc_{rl} = \frac{s_{rl} \times \alpha + c_{rl} \times \beta}{\alpha + \beta} \tag{5.7}$$

The distance measure between the two terminals will be Manhattan Distance(MD) measurement in terms of subregions. Here the automatically determined clusters will be of different sizes. Suppose for l^{th} layer the automatically determined number of clusters is N_l and the vertex of i^{th} cluster in l^{th} layer is v_{il}, where $i \in \{1...N_l\}$. Now for each cluster the *Fuzzified_Intra_Layer_Routing()* procedure is called which will connect all the terminals of the cluster with the center vertex of cluster. Here for a large net also number of cluster will be small so number of inserted pseudo-terminal will not hamper the cost much.

Generate_Routing_Path( )

Input : Set of sensitivity information for each node ($SInfo$), The total placement information ($PlacementInfo$), Obstacle information ($ObstacleInfo$), Netlist($NetlistInfo$), Parameter values ($ParamInfo$)
Output: Routing path in terms of subregions ($RoutedSubregions$), Percentage of rout-ability ensured($TotalRoutability$)

```
begin
    TotalWidth = Width(PlacementInfo);
    TotalLength = Length(PlacementInfo);
    NoOfLayers = Layers(PlacementInfo);
    NoOfTerminals = Total_Terminals(PlacementInfo);
    SubregionInfo = Get_Subregion_Information(PlacementInfo, TotalWidth, TotalLength);
    SRN = number of Subregions in the layout from SubregionInfo;
    for l ≤ NoOfLayers do
        GuidingInfo_l = Generate_Guiding_Info(SInfo, PlacementInfo, ObstacleInfo, l, ParamInfo);
        for j ≤ NoOfTerminals do
            r = Subregion(Terminal_j);
            ComSC_rl =
              (GuidingInfo_l → s_rl × ParamInfo → α + GuidingInfo_l → c_rl × ParamInfo → β)
              / (ParamInfo → α + ParamInfo → β);
        end
    end
    for n ≤ end(NetlistInfo) do
        if Inter_Layer_Net(NetlistInfo_n) then
            layerInfo_n = Layer_Info(NetlistInfo_n);
            for l ∀ layerInfo_n do
                ClusterInfo_nl = Automatic_Cluster_Determination(NetlistInfo_nl);
                RoutedSubregions = RoutedSubregions +
                Fuzzified_Intra_Layer_Routing(ClusterInfo_nl → center, ClusterInfo_nl,
                GuidingInfo, ParamInfo);
                if Routability then
                    TotalRoutability = TotalRoutability + 1;
                end
                noCls_n = Cluster_Number(ClusterInfo_nl);
                for c ≤ noCls_n do
                    ComSCPerCluster_lc = Σ_{∀r∈c} ComSC_rl;
                end
            end
            xBkBone = (Σ_{l=1}^{layerInfo_n} Σ_{i=1}^{noCls_n} ComSCPerCluster_il × ClusterInfo_il → x)
                      / (Σ_{l=1}^{layerInfo_n} Σ_{i=1}^{noCls_n} ComSCPerCluster_il);
            yBkBone = (Σ_{l=1}^{layerInfo_n} Σ_{i=1}^{noCls_n} ComSCPerCluster_il × ClusterInfo_il → y)
                      / (Σ_{l=1}^{layerInfo_n} Σ_{i=1}^{noCls_n} ComSCPerCluster_il);
            for l ∀ layerInfo_n do
                if !obstacle(xBkBone_l, yBkBone_l) then
                    PseudoTermInfo = Plot_Pseudo_Terminal(xBkBone_l, yBkBone_l);
                else
                    PseudoTermInfo = Plot_Pseudo_Terminal_Avoiding_Obstacles(xBkBone_l,
                    yBkBone_l, ComSCPerClusterInfo);
                end
                RoutedSubregions = RoutedSubregions +
                Fuzzified_Intra_Layer_Routing(ClusterInfo_nl → center, PseudoTermInfo,
                GuidingInfo, ParamInfo) + Subregion(PseudoTermInfo);
                if Routability then
                    TotalRoutability = TotalRoutability + 1;
                end
            end
        end
    end
end
```

Algorithm 10: Algorithm for generation of routing path in terms of subregions

Backbone Tree Construction and Determination of Pseudo Terminals

For the inter-layer connection through electrical via the construction of backbone tree is needed. The points where the vias will be inserted in a particular layer will depend upon the generated cluster's or terminals behaviour. The position of via insertion point should be inclined to the direction of larger and more sensitive or congested clusters or terminals. So here as a backbone tree determination strategy the combined sensitivity-congestion information for each clusters or terminals and the size of the clusters are used, which is stated in equation 5.8, 5.9 and 5.10. In equation 5.8 the total summation of sc for each cluster is determined, where $c_{il} \rightarrow i^{th}$ cluster in l^{th} layer. And the

$$sc_{il} = \sum_{\forall r \in i} sc_{rl} \tag{5.8}$$

The x coordinate (x_b) of backbone for all layers for a particular net is determined in equation 5.9 and the y coordinate(y_b) of the same is determined in equation 5.10. Where $vx_{il} \rightarrow$ x coordinate of the center of cluster i in l^{th} layer, $N \rightarrow$ total layer number and $N_l \rightarrow$ automatically determined cluster number in the l^{th} layer and $vy_{il} \rightarrow$ y coordinate of the center of cluster i in l^{th} layer. Then the backbone may be drawn through (x_b,y_b) point intersecting all the layers.

$$x_b = \frac{\sum_{l=1}^{N}\sum_{i=1}^{N_l}(sc_{il} \times vx_{il})}{\sum_{l=1}^{N}\sum_{i=1}^{N_l} sc_{il}} \tag{5.9}$$

$$y_b = \frac{\sum_{l=1}^{N}\sum_{i=1}^{N_l}(sc_{il} \times vy_{il})}{\sum_{l=1}^{N}\sum_{i=1}^{N_l} sc_{il}} \tag{5.10}$$

Next coming to the determination of final position of pseudo-terminals stage. In one layer , if there is no obstacle in (x_b, y_b) point, that point will work as a pseudo terminal and will participate in routing of that layer and one electrical via will be inserted into that point according to the function Plot_Pseudo_Terminal(). But if the point(x_b, y_b) falls on any obstacle in any layer then need to determine the pseudo terminal point on that device layer by the function Plot_Pseudo_Terminal_Avoiding_Obstacles(). For that here one obstacle avoidance strategy have been taken, which will also consider the cluster size and combined sensitivity-congestion ratio information as impacting factor during it's avoidance technique.

Obstacle Avoidance Scheme during Determination of Pseudo-Terminal: The position of pseudo terminal avoiding the obstacle will depends upon the of the total weight factor of the clusters falling on the four possible sides of the obstacles. Here the position of the pseudo terminal will be skewed to the more congested and sensitive and large clusters so that congestion does not increase more to that side. By injecting the pseudo terminal the vertical via position is actually inserted so it will help the electrical signal to pass quickly from the more sensitive and congested regions to other layer which may be favourable for routing. One example of determination of pseudo terminal in obstacle avoidance scheme is shown in Figure 5.3. The scheme is stated bellow.

1. The weight factor related to cluster size and combiner sensitivity-congestion ratio for i^{th} cluster in l^{th} layer is w_{il}, stated in equation 5.11, where n_{il} = number of terminals in i^{th} cluster in l^{th} layer and sc_{ikl} = the average sensitivity-congestion ratio of the

corresponding subregion where the k^{th} terminal belongs in i^{th} cluster in l^{th} layer.

$$w_{il} = n_{il} \times \sum_{k=1}^{n_{il}} sc_{ikl} \tag{5.11}$$

2. Next suppose the bottom left corner of the obstacle is (x_o,y_o) and length and width of the obstacle is h_o and w_o respectively. The coordinate of the four corner points of the obstacle anticlockwise will be (x_o,y_o),($x_o + w_o$,y_o), ($x_o + w_o$,$y_o + h_o$) and (x_o,$y_o + h_o$).

3. Need to calculate the four components (m_1, m_2, m_3, m_4), where

   (a) $m_1 = (x_o + w_o - x_b) \times \sum_{vx_{il} > x_b} w_{il}$

   (b) $m_2 = (x_b - x_o) \times \sum_{vx_{il} \leq x_b} w_{il}$

   (c) $m_3 = (y_o + h_o - y_b) \times \sum_{vy_{il} > y_b} w_{il}$

   (d) $m_4 = (y_b - y_o) \times \sum_{vy_{il} \leq y_b} w_{il}$

4. After determine the maximum value(m) among the four components according to equation 5.12, the final pseudo-terminal($x_{b'}$, $y_{b'}$) position will be determined.

$$m = max\{\, m_1, m_2, m_3, m_4 \,\} \tag{5.12}$$

   So here four conditions may arise

   (a) if $m = m_1$ then $x_{b'} = x_o + w_o$ and $y_{b'} = y_b$

   (b) if $m = m_2$ then $x_{b'} = x_o$ and $y_{b'} = y_b$

   (c) if $m = m_3$ then $x_{b'} = x_b$ and $y_{b'} = y_o + h_o$

   (d) if $m = m_4$ then $x_{b'} = x_b$ and $y_{b'} = y_o$

5. Ultimately the determined point($x_{b'}$, $y_{b'}$) will work as pseudo-terminal for that particular layer.

As a result of obstacle avoidance scheme the upper and lower layer of the obstacle containing layer, need to add two pseudo terminals and some extra wire as shown in Figure 5.3. But this overhead will not affect much because till now three dimensional technology supports maximum upto 6 layers or 8 layers. Now through those determined speudo terminals the backbone tree will be drawn intersecting all the layers in between or participating into the connection for that particular net. This total step by step approach for avoiding obstacle is actually stated inside Plot_Pseudo_Terminal_Avoiding_Obstacles() function.

Connecting Clusters to Backbone Tree

Connection between each cluster center with the pseudo terminal caused by the insertion of backbone tree is done by *Fuzzified_Intra_Layer_Routing()* procedure again. And all the routed subregion information is generated and returned to the user. If routing for all the nets is not ensured for a particular netlist then in Algorithm 10 the *Generate_Routing_Path( )* procedure will return the percentage of rout-ability achieved. In such cases the total technique can again be applied after some rip up and re route technique.

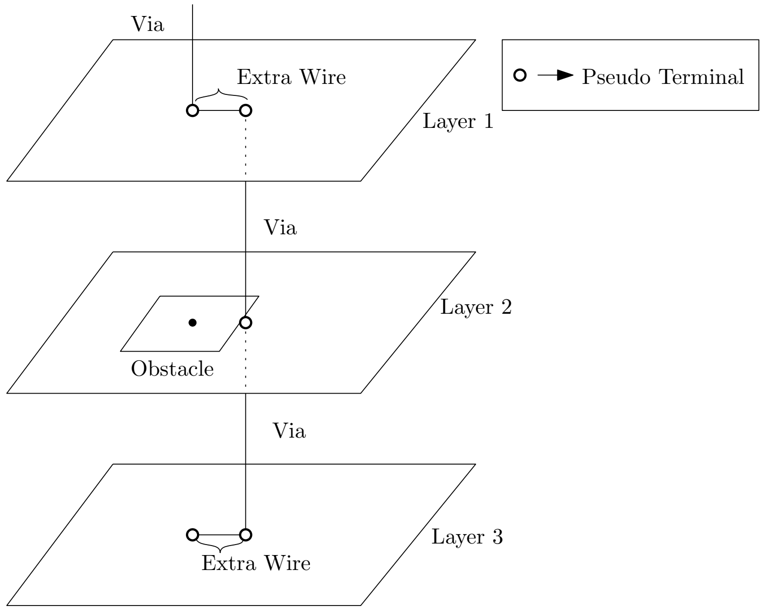


Figure 5.3: Insertion of pseudo terminal in presence of obstacle during backbone tree construction

5.4 Experimental Results

5.4.1 Framework

Proposed algorithm has been implemented in C, Java and MATLAB 7.14.0, MCR, and Matlab Builder JA. The GUI is designed in GUIDE and the fuzzy expert system is implemented using MATLAB fuzzy toolbox. The experiments were performed on a standard desktop environment with an Intel chip running at 2.30 GHz.The eight IBM-PLACE 2.0 benchmark suits (for Fixed-die Placement) are used.

5.4.2 Implementation

The Figure5.4 shows a snapshot of the implemented fuzzy expert system using fuzzy toolbox in MATLAB . For the crisp input set [0.638 0.641] the proposed rule base is producing a crisp output value of 0.631. The different activated fuzzy sets for premise part by two crisp inputs are the highlighted trapezoids, which fall on the crisp line in first two grids and for consequent part those highlighted trapezoids are on the crisp output line in last grid. From the snap shots of Figure 5.5 it is clear that using rule base the expert system is producing quite accurate output as changing of ineligibility factor is quite smooth here.

To signify the rout-ability for large circuits, the proposed algorithm is implemented IBM benchmarks for standard cells only. In Table 5.3 several experimental results are stated. The implementation is heading to minimize the time complexity for 3D ICs in fuzzified way. So the placement information regarding to 3 dimensional space for those benchmarks is taken, after transforming the 2D placement information to 3D placement information by a 2D to 3D placer tool. From the experimental results it is clear that for routing much less time is required. The main concern of the algorithm is the time required for guiding information generation phase. The rout-ability is achieved in range of 80% - 95% , which is quite encouraging result for a pioneering approach. With more number of device layers in 3D placement the rout-ability is increasing for larger circuits. This characteristic is the beauty of this proposed technique, as it is signifying the concept of 3D ICs. This chapter is aimed at only feasibility study of the proposed approach, not to compare it with other full-proved routers. The implementation is being enriched consequently.

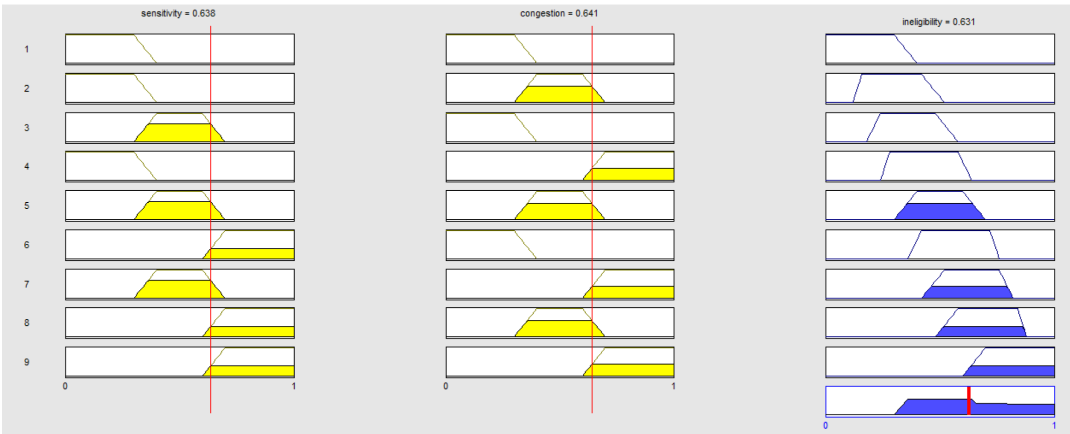
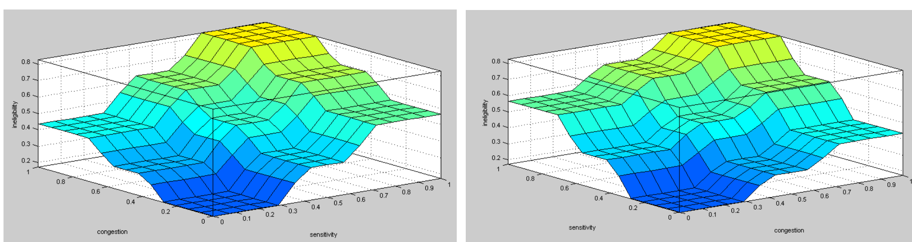


Figure 5.4: The fuzzification and defuzzification with respect to rule base



(a) Ineligibility factor with congestion and sensitivity

(b) Ineligibility factor with sensitivity and congestion

Figure 5.5: Change of ineligibility factor with sensitivity, congestion for proposed rule base

5.5 Conclusion

My approach is ensuring rout-ability for multi-pin inter-layer nets for 3 dimensional ICs by extending the pioneering approach of fuzzified technique to connect two pins within a single layer. Here also the generation of guiding information will give $O(l * m * n)$ time complexity in the proposed method, where l is the total number of layers, m is the number of subregions in x direction and n is the number of subregions in y direction of the layout. The total global routing procedure for connecting the two-pin, multi-pin as well as critical nets requires that much of complexity in addition with the time required to process fuzzy expert system for

Circuit Benchmark	# Layer	Vertical Capacity (per layer)	Horizontal Capacity (per layer)	Routability Achieved (%)	Time Required for Guiding Info Gen(sec)	Time Required for Routing(sec)
ibm01	4	80	80	87.51	121.0	7.0
ibm01	6	80	80	90.29	85.0	6.0
ibm02	4	80	80	78.54	196.0	18.0
ibm02	6	80	80	87.14	96.0	10.0
ibm07	4	80	80	70.0	457.0	100.0
ibm08	4	80	80	68.35	560.0	83.0
ibm08	6	80	80	75.27	524.0	76.0
ibm09	6	80	80	71.80	494.0	118.0
ibm09	8	80	80	78.72	596.0	103.0

Table 5.3: Experimental results for all nets connections in IBM benchmark circuits

each layer. The total methodology is proposed and verified successfully for global routing problem for two-pin, multi-pin and critical nets for 2 dimensional as well as 3 dimensional Integrated Circuits. It may also be considered as a new type of guided global routing approach for standard cells. The total procedure is tested for IBM-PLACE 2.0 benchmark suits (for Fixed-die Placement). Different types of comparative analysis, applications in other standard benchmark files and enhancement to the implementation for mixed sized cell placement may be considered as some possible extensions of this proposed work.

Chapter 6

Conclusion And Future Direction

6.1 Conclusion

3-D IC has a great potential for improving circuit performance and degree of integration. So performing a good routing in the 3D ICs will facilitate the performance. Though there are already a number of thesis and published paper upon this but still it can be made better by some new routing approaches to provide more flexibility.

Throughout the project primary concentration was on stating a fuzzy logic control system for global routing by sensitivity analysis for 3D ICs and its feasibility study. And another one is a algorithm for obstacle aware multi net global routing in 3D ICs. A tool having three options for congestion aware, thermal aware, and distance aware global routing for 2-pin intra-layer net is developed, so that the user can get three types of multi objective routing strategy for the same specification. Then in our fuzzy logic based multi-pin inter-layer net routing technique the larger nets will be routed. So for a total benchmark circuit all the 2-pin nets situated in same layer, will be routed in fuzzified 2-pin intra-layer net routing technique. And the critical and larger nets will be routed according to the fuzzified multi-pin inter-layer net routing technique. Thus by merging these two approaches the total fuzzified method for global routing is constructed.

The proposed deterministic obstacle aware approach consists of shaping and avoiding non-uniform obstacles and constructing a backbone tree with group steiner tree and cluster determination approach. To the best of our knowledge the proposed fuzzified technique is till now a pioneering work in global routing in VLSI physical design and in this project work it's feasibility is proved for ensuring the reliability and escaping the bound of deterministic approach.

6.2 Future Aspect

The next target of this work will be to merge and compare the main two proposed technique i.e. thermal aware fuzzified approach and obstacle aware deterministic approach. The designed tool can be enhanced in a way, so that the user can get two options to determine her objective function, whether obstacle aware or thermal aware. By this way the user can compare the fuzzified and deterministic approach for the same input circuit in terms of timing complexity and reliability. Thus success of our comparison to prove that proposed pioneering fuzzified approach is faster in time and have comparable reliability with deterministic one then that will open up a new area of research in global routing of 3D ICs.

Bibliography

[1] M. Setnes, R. Babuska, U. Kaymak, and H. W. N. Lemke, "Similarity Measures in Fuzzy Rule Base Simplification," vol. B-28, 1998.

[2] N. Sherwani, *Algorithms for VLSI Physical Design Automation*. Kluwer Academic, Dordrecht, 1999.

[3] S. M. Sait and H. Youssef, *VLSI Physical Design Automation*. World Scientific,India, 1999.

[4] G. Moore, "Cramming More Components onto Integrated Circuits," in *Electronics 38*, pp. 114–117, 1965.

[5] C. T. Lin and C. S. G. Lee, *Neural Fuzzy systems:A Neuro-Fuzzy Synergism to Intelligent Systems*. Prentice-Hall, London, 1996.

[6] M. R. Garey and D. S. Johnson, "The Rectilinear Steiner tree problem is NP complete," pp. 826–834, 1977.

[7] C. Y. Lee, "An Algorithim for path connection and its applications," in *IRE Transactions on Electronic Computers*, pp. 346–365, 1961.

[8] J. L. Ganley and J. P. Cohoon, "Routing a Multi-Terminal Critical Net: Steiner Tree Construction in the Presence of Obstacle," in *IEEE International Symposium on Circuits and Systems*, pp. 113–116, 1994.

[9] D. W. Hightower, "A solution to the line-routing problem on the continuous plane," in *Proceedings of the Sixth Design Automation Workshop*, pp. 1–24, 1969.

[10] J. Liu, Y. Zhao, E. Shragowitz, and G. Karypis, "A Polynomial Time Approximation Scheme for Rectilinear Steiner Minimum Tree Construction in the Presence of Obstacles," in *Proceedings of Electronics, Circuits and Systems*, vol. 2, pp. 781–784, 2002.

[11] Y. Hu, Z. Feng, T. Jing, X. Hong, Y. Y. Ge, X. Hu, and G. Yan, "FORst:A 3-step heuristic for obstacle-avoiding rectilinear Steiner minimal tree construction," in *Proceedings of JICS*, pp. 107–116, 2004.

[12] Z. Shen, C. Chu, and Y. Li, "Efficient Rectilinear Steiner Tree Construction with Rectilinear Blockages," in *Proceedings ICCD*, pp. 38–44, 2005.

[13] H. Wu, T. Jing, X. Hong, Z. Feng, X. Hu, and G. Yan, "An-OARSMan: Obstacle-avoiding Routing Tree Construction with Good Length Performance," in *Proceedings ASP-DAC*, pp. 7–12, 2005.

[14] J. Long and S. O. M. H. Zhou "EBOARST: An efficient edgebased obstacle avoiding-rectilinear Steiner tree construction algorithm," in *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, pp. 2169–2182, Dec 2008.

[15] L. Li and E. F. Y. Young, "Obstacle-avoiding rectilinear Steiner tree construction," in *Proceedings of ICCAD*, pp. 523–528, 2008.

[16] G. Ajwani, C. Chu, and W. K. Mak, "FOARS: FLUTE Based Obstacle-Avoiding Rectilinear Steiner Tree Construction," in *Proceedings of Int. Symp. Phys. Des.*, pp. 27–34, 2010.

[17] L. Li, Z. Qian, and E. F. Y. Young, "Generation of optimal obstacle avoiding rectilinear Steiner minimum tree," in *Proceedings of ICCAD*, pp. 21–25, 2009.

[18] T. Huang and E. F. Y. Young, "Obstacle-avoiding Rectilinear Steiner Minimum Tree Construction: An Optimal Approach," in *Proceedings of Int. Conf. Computer-Aided Des.*, pp. 610–613, 2010.

[19] Y. Shen, Q.Liu, and W.Guo, "Obstacle-avoiding Rectilinear Steiner Minimum Tree Construction Based on Discrete Particle Swarm Optimization," in *Proceedings of Natural Computation (ICNC)*, vol. 4, pp. 2179–2183, July 2011.

[20] J.R.Minz, M.Pathak, and S.K.Lim, "Net and pin distribution for 3D package global routing," in *Proceedings of Design, Automation and Test in Europe*, vol. 2, p. 21410, 2004.

[21] J.Rajkumar, Minz, E.Wong, M.Pathak, and S.K.Lim, "Placement and Routing for 3-D System-On-Package Designs," in *Proceedings of Components and Packaging Technologies, IEEE Transactions*, vol. 29, pp. 644–657, Sep 2006.

[22] J. Minz, E. Wong, and S. Lim, "Thermal and Crosstalk-Aware Physical Design For 3D System-On-Package," in *Proceedings of Electronic Components and Technology Conference*, pp. 824 – 831, Jun 2005.

[23] P.Ghosal, H.Rahaman, S.Das, A. Das, and P.Dasgupta, "Obstacle Aware Routing in 3D Integrated Circuits," in *Lecture Notes in Computer Science*, pp. 451–460, 2012.

[24] P. Ghosal, S. Das, and A. Das, "A Novel Algorithm for Obstacle Aware RMST Construction during Routing in 3D ICs," in *Advances in Computing and Information Technology*, vol. 2, pp. 649–658, 2012.

[25] P. Ghosal, A. Das, and S. Das, "A New Class of Obstacle Aware Steiner Routing in 3D Integrated Circuits," in *Advances in Computing and Information Technology,*, vol. 3, pp. 657–666, 2012.

[26] Y.Shen, Q.Liu, and W. Guo, "Obstacle Aware RMST Generation Using Non-Manhattan Routing For 3D ICs," in *Proceedings of Natural Computation (ICNC)*, vol. 4, pp. 2179–2183, July 2011.

[27] G. Reich and P. Widmayer, "Beyond Steiners problem: A VLSI oriented generalization," in *Graph-Theoretic Concepts in Computer Science WG89,Lecture Notes in Computer Science*, vol. 411, (Berlin,New York), pp. 196–210, Springer-Verlag, 1990.

[28] N. Garg, G. Konjevod, and R. Ravi, "A polylogarithmic approximation algorithm for the group steiner tree problem," pp. 66–84, 2000.

[29] C. Chekuri, G.Even, and G. Kortsarz, "A greedy approximation algorithm for the group Steiner problem ," in *Discrete Applied Mathematics*, vol. 154, pp. Pages 15–34, Jan 2006.

[30] R. Khandekar, G. Kortsarz, and Z. Nutov, "Approximating fault-tolerant group-Steiner problems," in *Theoretical Computer Science*, vol. 416, pp. 55–64, Jan 2012.

[31] A.Biswas, P. Bhowmik, and B.B.Bhattacharya, "Construction of isothetic covers of a digital object : A combinational approach," vol. 21, p. 295310, May 2010.

[32] B. Goplen and S. S. Sapatnekar, "Thermal via placement in 3D ICs," in *Proceedings of the International Symposium on Physical Design*, pp. 167–174, 2005.

[33] J. Minz, E. Wong, and S. K. Lim, "Thermal and Crosstalk-Aware Physical Design For 3D System-On-Package," in *Proceedings of Electronic Components and Technology Conference*, pp. 824–831, 2005.

[34] L. A. Zadeh, "Fuzzy Sets," 1965.

[35] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," vol. 3, pp. 28– 44, 1973.

[36] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," 1974.

[37] P. Ghosal, T. Samanta, H. Rahaman, and P. Dasgupta, "Thermal-aware Placement of Standard Cells and Gate Arrays:Studies and Observations," in *Symposium on VLSI, 2008. ISVLSI '08. IEEE Computer Society Annual*, pp. 369 – 374, Apr 2008.

[38] S. M. Sait and H. Y. H. Ali, "Fuzzy Simulated Evolution Algorithm for Multi-objective Optimization of VLSI Placement," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 1, 1999.

[39] T. Zhang, Y. Zhan, and S. S. Sapatnekar, "Temperature-aware routing in 3D ICs," in *Proceedings of the Asia-South Pacific Design Automation Conference*, pp. 309–314, 2006.

[40] T. Zhang, Y. Zhan, S. S. S. Cong, and Y. Zhang, "Thermal-driven multilevel routing for 3-D ICs," in *Proceedings of the Asia-South Pacific Design Automation Conference*, pp. 121–126, 2005.

[41] P. Ghosal, H. Rahaman, and P. Dasgupta, "Uniform Thermal Distributions in Placement of Standard Cells and Gate Arrays: Algorithms and Results," in *Proceedings of the 13th IEEE/VSI VLSI Design and Test Symposium (VDAT 2009)*, pp. 69–78, 2009.

[42] P. Ghosal, H. Rahaman, and P. Dasgupta, "Minimizing Thermal Disparities During Placement in 3D ICs," in *Proceedings of The 13th IEEE International Conference on Computational Science and Engineering (CSE - 2010)*, pp. 160–167, 2010.

[43] P. Ghosal, H. Rahaman, and P. Dasgupta, "Cell Level Thermal Placement in 3D ICs," in *Proceedings of IEEE INDICON 2010*, pp. 1–4, 2010.

[44] P. Ghosal, H. Rahaman, and P. Dasgupta, "An Experimental Study of Interconnection Length in 3D and 2D VLSI," in *Proceedings of IEEE Latin American Symposium on Circuits and Systems*, 2010.

[45] P. Ghosal, H. Rahaman, and P. Dasgupta, "Thermal Aware Placement in 3D ICs," in *Proceedings of 2nd International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom 2010)*, pp. 66–70, 2010.

[46] M. Pathak and S. K. Lim, "Thermal-aware Steiner Routing for 3D Stacked ICs," in *Proceedings of ICCAD*, pp. 205–211, 2007.

[47] M. Pathak and S. K. Lim, "Performance and Thermal-Aware Steiner Routing for 3-D Stacked ICs," in *Proceedings of Computer-Aided Design of Integrated Circuits and Systems*, pp. 1373–1386, 2009.

[48] C. J. Alpert, D. P. Mehta, and S. S. Sapatnekar, *Handbook of Algorithms for Physical design Automation*. Auerbach Publications Boston, MA, USA, 2008.

[49] C. J. Chang, P. J. Huang, T. C. Chen, and C. N. J. Liu, "ILP-based inter-die routing for 3D ICs," in *Proceedings of ASP-DAC*, pp. 330–335, 2011.

[50] T. H. Wu, A. D. Jeffrey, and T. Linderoth, "GRIP: Scalable 3D Global Routing Using Integer Programming," in *Proceedings of DAC-2009*, pp. 320–325, 2009.

[51] M. Cho, K. Lu, K. Yuan, and D. Z. Pan, "BoxRouter 2.0: A Hybrid and Robust Global Router with Layer Assignment for Routability," in *Proceedings of ICCAD*, 2007.

[52] S. Das, A. Chandhakasan, and R. Reif, "Design Tools for 3-D Integrated Circuits," in *Proceedings of ASP-DAC '03*, pp. 53–56, 2003.

[53] M. D. Moffitt, "Global routing revisited," in *Proceedings of the 2009 International Conference on Computer-Aided Design, ICCAD 2009*, pp. 805–808, 2009.

[54] L. X. Wang, "Fuzzy systems are universal approximators," in *Proceedings of IEEE Int. Conf. Fuzzy Systems*, (San Diego, CA), 1992.

[55] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Symposium on Math, Statistics and Probability, Berkeley, CA: University of California Press*, pp. 281– 297, 1967.

[56] C. Chu, "FLUTE: Fast Lookup Table Based Wirelength Estimation Technique," in *Proceedings of ICCAD-2004*, pp. 696–701, 2004.

[57] E. H. Mamdani, "Application of fuzzy algorithms for control of a simple dynamic plant," in *Proceedings of Inst. Electr. Eng.*, pp. 1585–1588, 1974.

[58] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," vol. SMC-15, pp. 116–132, 1985.

[59] B. G. Song, R. J. M. II, S. Oh, P. Arabshahi, T. P. Caudell, and J. J. Choi, "Adaptive membership function fusion and annihilation in fuzzy if-then rules," in *Proceedings of FUZZ-IEEE/IFES 93*, (San Diego, CA), pp. 961–967, 1993.

[60] D. Dubois and H. Prade, "Fuzzy Sets and Systems: Theory and Applications," 1980.

[61] G. J. Klir and B. Youan, "Fuzzy Sets and Fuzzy Logic: Theory and Applications. Englewood Cliffs," 1995.

[62] S. Bandyopadhyay, "Automatic Determination of the Number of Fuzzy Clusters Using Simulated Annealing with Variable Representation," in *Proceedings of ISMIS 2005*, pp. 549–602, 2005.