FuzzRoute: A Thermally Efficient Congestion-Free Global Routing Method for Three-Dimensional Integrated Circuits

DEBASHRI ROY and PRASUN GHOSAL, Indian Institute of Engineering Science and Technology
SARAJU MOHANTY, University of North Texas

The high density of interconnects, closer proximity of modules, and routing phase are pivotal during the layout of a performance-centric *three-dimensional integrated circuit* (3D IC). Heuristic-based approaches are typically used to handle such NP-complete problems of global routing in 3D ICs. To overcome the inherent limitations of deterministic approaches, a novel methodology for multi-objective global routing based on fuzzy logic has been proposed in this article. The guiding information generated after the placement phase is used during routing with the help of a fuzzy expert system to achieve thermally efficient and congestion-free routing. A complete global routing solution is designed based on the proposed algorithms and the results are compared with selected fully established global routers, namely Labyrinth, FastRoute3.0, NTHU-R, BoxRouter 2.0, FGR, NTHU-Route2.0, FastRoute4.0, NCTU-GR, MGR, and NCTU-GR2.0. Experiments are performed over ISPD 1998 and 2008 benchmarks. The proposed router, called *FuzzRoute*, achieves balanced superiority in terms of routability, runtime, and wirelength over others. The improvements on routing time for Labyrinth, BoxRouter 2.0, and FGR are 91.81%, 86.87%, and 32.16%, respectively, for ISPD 1998 benchmarks. It may be noted that, though FastRoute3.0 achieves fastest runtime, it fails to generate congestion-free solutions for all benchmarks, which is overcome by the proposed FuzzRoute of the current article. It also shows wirelength improvements of 17.35%, 2.88%, 2.44%, 2.83%, and 2.10%, respectively, over others for ISPD 1998 benchmarks. For ISPD 2008 benchmark circuits it also provides 2.5%, 2.6%, 1 %, 1.1%, and 0.3% lesser wirelength and averagely runs 1.68×, 6.42×, 2.21×, 0.76×, and 1.54× faster than NTHU-Route2.0, FastRoute4.0, NCTU-GR, MGR, and NCTU-GR2.0, respectively.

Categories and Subject Descriptors: B.7.2 [**Integrated Circuits**]: Design Aids—*Placement and Routing*

General Terms: Design, Algorithms, Performance, Experimentation

Additional Key Words and Phrases: VLSI layout design, global routing, fuzzy expert system, fuzzified global routing

ACM Reference Format:
Debashri Roy, Prasun Ghosal, and Saraju Mohanty. 2015. FuzzRoute: A thermally efficient congestion-free global routing method for three-dimensional integrated circuits. ACM Trans. Des. Autom. Electron. Syst. 21, 1, Article 1 (November 2015), 38 pages.
DOI: http://dx.doi.org/10.1145/2767127

1. INTRODUCTION

Nanoscale technology permits us to integrate systems with billions of transistors on a single chip. Layout design plays a pivotal role in the design cycle by transforming the circuit description into geometric description. Recent researches on global routing

One initial work of this approach has been reported in IEEE Annual Symposium on VLSI, 2014 [Roy et al. 2014b].

Authors' addresses: D. Roy and P. Ghosal (corresponding author), Department of Information Technology, Indian Institute of Engineering Science and Technology, P.O. Botanic Graden, Shalimar, Howrah, West Bengal 711103, India; email: prasung@gmail.com; S. Mohanty, Department of Computer Science and Engineering, University of North Texas, 1155 Union Circle no. 311366, Denton, TX 76203-5017.

1084-4309/2015/11-ART01 $15.00
DOI: http://dx.doi.org/10.1145/2767127

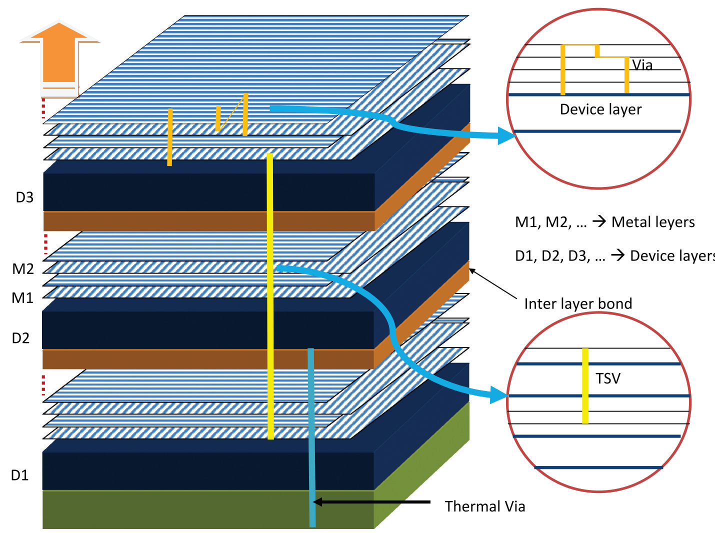


Fig. 1. 3D integration structure.

are aimed at optimization of different multi-objective functions related to performance and congestion, thermal issues, proper insertion of thermal vias [Goplen and Sapatnekar 2005], sensitivity, wirelength, critical, paths or crosstalk [Minz et al. 2005], etc. Consideration of the net ordering problem in OTC (*over the cell*) routing is a big challenge to reach a polynomial-time solution. Some other metaheuristics, like Simulated Annealing- and Genetic Algorithm-based approaches, are also influencing modern trends. But, to the best of our knowledge, no existing complete fuzzified method for global routing (proven to be an improved way out of the problems with deterministic approaches) is reported for large-scale problems in global routing of three-dimensional integrated circuits.

1.1. Global Routing: Trade-Offs and Challenges

Global routing plays a very important role in VLSI physical design to achieve a faster response from any integrated circuits. Approximate connection paths among the nodes of any net are determined here that lead the global routing problem to be an NP-complete one. An optimal connectivity for a circuit depends on several constrains imposed during the global routing step and thereby specifying a special performance feature of that IC.

Addition of a third dimension has allowed researchers to enrich performance of ICs in a better degree, but have made the problem of global routing more complex. Consideration of third dimension during routing decision-making is a witty and complex measure to perform in every global routing algorithm. One basic structure of three-dimensional global routing has been demonstrated pictorially in Figure 1.

1.2. Fuzzy Logic and FLC

Fuzzy logic is a form of multi-valued logic or probabilistic logic that deals with approximate reasoning rather than fixed and exact. In contrast to traditional crisp logic [True/False], they can have varying values that range in degree between 0 and 1.

Linguistic variables are the input or output variables of the system whose values are words or sentences from a natural language instead of numerical values. A membership

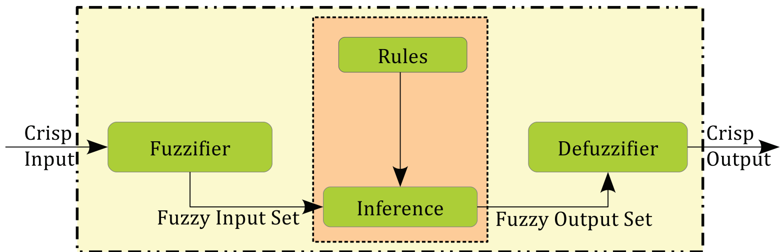


Fig. 2. A fuzzy logic expert system.

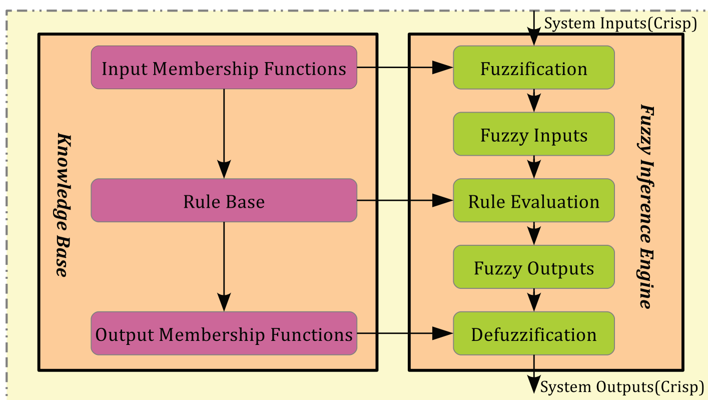


Fig. 3. A fuzzy logic expert system.

function is used to quantify a linguistic term and in fuzzification and defuzzification procedures.

A *fuzzy logic control* (FLC) system may be defined as the nonlinear mapping of an input dataset to a scalar output data. An FLC consists of four main parts: (a) fuzzifier; (b) rules; (c) inference engine; and (d) defuzzifier. These components and the general architecture of an FLC is shown in Figure 2.

The process of fuzzy logic is explained in Figure 3. First, a crisp set of input data are gathered and converted to a fuzzy set using fuzzy linguistic variables, fuzzy linguistic terms, and membership functions. This step is known as fuzzification. Afterwards, an inference is made based on a set of rules. Lastly, the resulting fuzzy output is mapped to a crisp output using the membership functions in a defuzzification step.

During the past decade, fuzzy logic control [Zadeh 1973] has been considered as one of the most promising research areas in the application of industrial process control to medical diagnosis and securities trading [Pedryz and Gomide 2007; Lughofer 2011]. The main idea behind FLC is to incorporate the *expert experience* of a human interface in designing a controller. Several possible ways of FLC implementation are demonstrated in Abonyi [2003] and Babuska [1998].

The rest of the article is organized as follows. Section 2 summarizes the novelties of our proposed work, followed by Section 3 providing current state-of-the-art and motivation behind this work. In Section 4, a formulation of the overall problem is presented. Description of approaches for prerouting guiding information generation, as well as the proposed fuzzy logic expert system are described in Section 5. The proposed heuristic for three-dimensional routing is explained in Section 6. Next, the overall proposed routing scheme is presented at a glimpse in Section 7. The proposed solution approach for fuzzified global routing for all the three categorized types of net (that may be considered as the backbone of this work) is covered in Section 8. Experimental results, like a feasibility study for the global routing model and various types of comparison studies for a complete fuzzified global router, are given in Section 9. Section 10 concludes the article by giving important extensions and directions of our initiative.

2. NOVEL CONTRIBUTIONS OF THE CURRENT ARTICLE

A complete fuzzified global router is reported using the fuzzy logic concept for routing all the nets in a netlist in this article. Novelties of this approach lie in many folds, including providing the solution within a feasible time and with a better degree of reliability. A standard cell-based design style is used for testing with benchmark circuits. However, the proposed method can easily be extended to mixed cell design also.

An efficient fuzzified expert system has been designed for thermal- and congestion-aware global routing in routing space for a fully 3D IC structure. The concept of thermal sensitivity is adapted from Ghosal et al. [2008]. This article corroborates the overall models and procedures of the proposed fuzzified global routing in three-dimensional space. To the best of our knowledge, the proposed fuzzified approach for multi-pin global routing in 3D ICs is the first of its kind. The novelties listed next provide an overview of our contributions.

—*Nondeterministic.* This approach provides a way out of the alternative to standard deterministic or heuristic-based approaches to overcome their inherent limitations in handling complex design problems.
—*P solution to NP.* Our proposed approach ensures getting a polynomial-time solution for this NP-complete problem. Time complexity analysis of the proposed algorithms is reported in Section 9.7.
—*Exhibiting feasibility.* Depending upon the nature of the complexity and size of the problem, it is validated as providing a feasible fuzzified global routing model between a source and destination. The overall feasibility analysis with fuzzy expert system is presented in Section 9.2.
—*Dynamic and portable.* It ensures consideration of the dynamic fuzzy expert system and making it portable for all circuits. The fuzzy expert system works dynamically by applying different boundary values for different fuzzy sets depending upon the input benchmark circuit. Details are lucidly depicted in Sections 5.3 and 5.6.
—*Superiority.* It gets better results with respect to time and reliability than the other established global routers. Comparison with some of those is provided in Tables IV and V in Section 9.5.
—*Robust and extensible.* It exhibits the robustness property by providing comparable results with varying parameter values. Table II in Section 9.4 presents comparable results for different sets of parameter values. Due to the property of fuzzy-logic-based multi-objective optimization formulation used in this work, the proposed tool is also open and extensible to greater numbers of constraints with the primary constraints, namely thermal and congestion, used in this work. This may be easily achieved by simply tweaking the formulation of the problem by incorporating other parameters.
—*Adaptability.* It exhibits a good adaptive property by one proposed obstacle avoidance mechanism. The scheme is described lucidly in Section 8.4.
—*Completeness.* A sequence of global routing paths is given in terms of subregions generated for all two-pin and multi-pin intra- as well as inter-layer nets in a netlist. Extensive analysis on recent ISPD'98 and '08 benchmarks is presented in Section 9.3 as validation of the proposed technique.

3. STATE-OF-THE-ART AND MOTIVATION

3.1. Some Existing Global Routing Techniques

In high-performance VLSI circuits, the on-chip power densities play a dominant role, due to the increased scaling of technology and increasing number of components, frequencies and bandwidths. Consumed power, usually converted into dissipated heat, affects the performance and reliability of a chip. Generation of hotspots is a critical issue in the VLSI physical design phase. Several works have been reported [Zhang et al. 2005, 2006; Ghosal et al. 2009, 2010a, 2010b, 2010c, 2010d] have experimented on thermal-aware placement and routing for 2D as well as 3D integrated circuits. Pathak and Lim [2009] have presented a novel algorithm on 3D Steiner routing by an NLP-based approach for thermal-aware global routing in 3D stacked ICs. The complexity of a Steiner-tree-based approach becomes very high with multi-net.

In Kastner et al. [2002], a concept of pattern routing was developed to guide subsequent maze routing. A global routing was proposed in Zhang et al. [2008], where a fast maze routing was presented by introducing the virtual capacity concept. A history-based cost-function-driven multi-source and sink maze routing was proposed in Gao et al. [2008], and negotiated congestion routing was proposed in Roy and Markov [2008]. The bounded maze routing concept was used in Dai et al. [2012] and Liu et al. [2013] for 3D global routing to achieve significant wirelength and congestion on the most recent benchmarks.

In the case of inter-die routing, one ILP-based technique has been introduced by Chang et al. [2011]. Similarly, an integer-programming-based approach was proposed by Wu et al. [2009]. Their proposed approach optimizes wirelength and via cost without going through a layer assignment phase. In Cho et al. [2009], 2D-to-3D mapping was done by a layer assignment powered by progressive via- or blockage-aware integer linear programming. All these are global routing approaches in three-dimensional space. In Das et al. [2003], authors designed some routing-and-placement-specific tools for 3D ICs.

Among the different global routers reported so far, Chang et al. [2011] and Wu et al. [2009] optimize wirelength and via cost without going through a layer assignment phase, unlike Cho et al. [2009]. Several other global routers were proposed in Zhang et al. [2008], Gao et al. [2008], and Roy and Markov [2008]. Recent global routing benchmarks like ICCAD'09 [Moffitt 2009], ISPD'08 [Nam et al. 2008], and ISPD'11 [Viswanathan et al. 2011] consider only the 3D situation for metal layers. A multiple-device layer, that is, a fully 3D strategy, has not been replicated in benchmark until now.

3.2. Thermal Issues: Its Impact and Modeling

Localized regions of high heat flux, called hotspots, are becoming significant with increased scaling of process technology along with increase in total power dissipation. The problem is quite severe in the case of three-dimensional integrated circuits, due to the close proximity of neighboring modules. The temperature of hotspots is generally above average die temperature. Uniformity of power dissipation is quite desirable to achieve certain optimized chip performance. In some recent works like Ghosal et al. [2008, 2010c, 2010d]; authors have expressed their concern over this issue in achieving an optimized thermal-aware placement in 3D ICs.

Thermal issues have always been an increasingly big concern throughout the layout design of 3D ICs. Recent researches like Gupta et al. [2008] and Lu and Pan [2009] show its present importance during routing. In Gupta et al. [2008], authors have proposed one thermal-aware global routing technique to reduce the probability of failure of chips due to interconnect failures by routing more wires in the colder regions of the chip and less in the hotter regions. Then, a reliability-aware global routing with thermal considerations was reported in Lu and Pan [2009] to reduce the probability of interconnect failures by thermal-driven minimum spanning tree construction and thermal-driven maze routing. Some thermal-aware placement techniques and models are also described later.

Voltage drop is another issue affecting the thermal profile of the chip, though it has not been considered during the present formulation. As its impact is strongly correlated with thermal effects, it may be taken in account in future work.

3.3. Application-Specific Fuzzy Logic Implementation

Fuzzy systems have been attracting research for the past decade, due to their property as an universal approximator [Wang 1992; Castro and Delgado 1996]. Application of fuzzy logic is being implemented extensively in diverse fields like medicine [Kanthi et al. 2013], frequency control [Sabahi et al. 2014], vehicle path planning [Huang et al. 2014], etc. Sait and Ali [1999] had proposed a fuzzy simulated evaluation algorithm for placement. One source-destination-only fuzzified global routing model for VLSI layout design has been reported in Roy and Ghosal [2013]. A 2-pin-only global router has also been presented in Roy et al. [2014a]. But no such notable contribution has been found to apply fuzzy logic in designing a complete global router for 3D ICs which is able to route an entire netlist consisting of two/multi-pin intra/inter-layer nets as well as critical nets.

3.4. Scope and Relevance of Present Work

In the modern era, the design complexity of different problems is increasing in exponential order. Due to their very large problem size, such problems have seemed unsolvable in feasible time, even by some heuristics. So, considering the decision only in a deterministic way leads the complexity of the problem to NP-completeness. Global routing has also been facing such problems recently. In our proposed approach, we have tried to achieve a degree of reliability for each solution of global routing. Since fuzzy systems have already been recognized as universal approximators used to formulate our pioneering work. In the fuzzified approach, the search space is decreased for a particular solution, and so is the time and design complexity. This is possible only for global routing. Detailed routing is not applicable here as fine-tuning is necessary, and this can only be done in a deterministic way.

As the presently proposed technique was not motivated by any subtle shortcoming of a particular established global router, our initial thrust was given to a feasibility study of the proposed method and thereby identify and establish its applicability. The total algorithm was preferred first for implementation over widely accepted benchmark suites (ISPD'98 [Alpert 1998] and ISPD'08 [Nam et al. 2008]).

Here thermal optimization has been done during the placement phase by an efficient thermal placer with certain necessary modifications introduced. Based upon the model and depending upon the thermal influence of each module on the overall thermal profile of the entire layout area, the modules have been categorized in three different classes [Ghosal et al. 2008]. Thermal sensitivity of a module is defined as its effect on the overall thermal scenario. When the circuit is in operation, that is, during the dynamic scenario as the switching occurs, then, depending upon the nature of variation, thermal sensitivity measures to what degree a module is responsible for changing the overall thermal profile of the chip. Depending on the strength of effect, the modules are classified into three different classes of sensitivity. These three classes of modules, namely strongly or highly sensitive, moderately sensitive, and weakly sensitive, obtained by this thermal sensitivity analysis have been used in the next phase of layout, that is, routing as a guiding factor. By these means, an intelligent technique has been developed

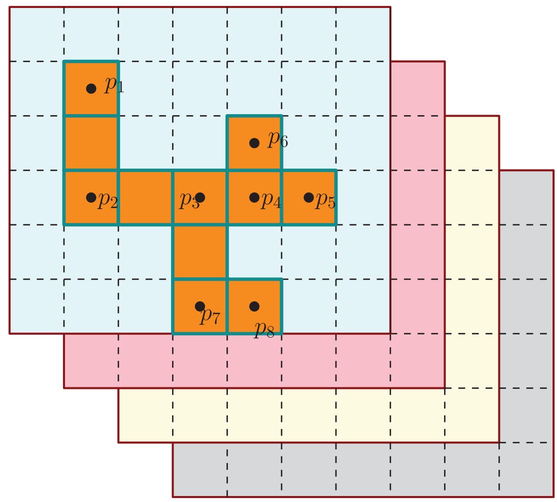


Fig. 4. Geometric description of problem statement.

to route the nets while avoiding hotspots and thermally sensitive areas. Here, no "thermal via" or "thermal wire" insertion has been considered.

4. PROBLEM FORMULATION FOR FUZZIFIED GLOBAL ROUTER

4.1. Description of Problem

Let $P = \{p_1, p_2, p_3, \ldots, p_k\}$ be a set of pins of a k-pin net distributed across L layers and L be the number of layers available. Let $M = \{m_1, m_2, m_3, \ldots, m_r\}$ be a set of modules spread over the routing layer, where (x_i, y_i) are the bottom-left coordinates of module m_i. The *thermal sensitivity* (SI) of a module is in the range of 0 to 1, and the *congestion ratio* (CR) for each module is in the range of 0 to 1. Also, α, β are two cost factor coefficients, where $\alpha + \beta = 1$.

A weighted cost factor is generated from the thermal sensitivity and congestion ratio information incorporated with the α and β values.

4.2. Geometric Description with Example

Standard cell-based design style has been used for the implementation. So, the total routing layer is represented as a grid structure. The routing layout is divided into subregions, where each subregion is composed of certain grids. One subregion is the main routing unit here. In Figure 4, one net is shown on one device layer. Here, $P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8\}$. So, that multi-pin net is spread over eight modules, darkened in that figure. The routing path needs to be determined for M depending on α and β values and some constraints.

4.3. Definition and Formulation of Weighted Cost Factor

Thermal sensitivity and congestion are two considered constraints. The probability of getting selected as the next followed subregion is determined by the routing eligibility for each subregion. The relationship between routing eligibility and the two constraints is represented by a weighted cost factor, namely the. *ineligibility factor* (IF).

The ineligibility factor is inversely proportional to routing eligibility. Formulation of IF may be represented as a standard minimization problem, as stated in (1).

$$
\begin{aligned}
\text{minimize} \quad & IF = SI \times \alpha + CR \times \beta \\
\text{subject to} \quad & \alpha + \beta = 1 \\
& 1 \geq \alpha, \beta \geq 0 \\
& 1 \geq SI \geq 0 \\
& 1 \geq CR \geq 0.
\end{aligned}
\tag{1}
$$

4.4. Objective

The objective is to build a fuzzified global router that is able to route by determining the routing region with minimum wirelength for the total netlist, depending upon the weighted cost function maintaining two constraints, namely thermal sensitivity and congestion ratio for each net.

5. APPROACH FOR PREROUTING INFORMATION GENERATION

Our proposed routing procedure is a fuzzified approach to find one in-between solution of deterministic and heuristic-based approaches. Prior to the routing step, some prerouting information are generated that will help in subsequent routing steps. One main purpose of these globally routed paths will be to avoid the more heated and congested portion of the layout. This constraint-based approach takes the decision during the routing procedure from some prerouting guiding information. This guiding information is generated from a proposed fuzzy expert system and rule base. Every fuzzy expert system has two parts: (a) premise, that is, the input part; and (b) consequent, that is, the output part. Different definitions and assignments of fuzzy terms and the guiding information generation procedure are described next.

5.1. Linguistic Variables: Thermal Sensitivity and Congestion Ratio

Unlike numerical variables that take numerical values, linguistic variables take linguistic values. In any fuzzy-based logic, recognizing proper linguistic variables plays an important role. In this fuzzified global routing procedure, thermal sensitivity and congestion ratio are the two linguistic variables in the premise part. *Ineligibility factor* (IF) is the only linguistic variable in the consequent part. Here, the linguistic values to be taken are: (a) high; (b) moderate; and (c) weak for the premise part.

5.2. Fuzzification of Thermal Sensitivity, Congestion Ratio and Ineligibility Factor

In the fuzzy logic concept, linguistic variables vary within [0,1]. Here, all derived fuzzy sets from three linguistic variables in both the premise and consequent part are shown in Figure 5.

In the consequent part, a total 9 fuzzy sets are present. Only one rule base is sufficient for this purpose. For the rule base, there are total of 9, that is, all possible fuzzy sets ($i^{th}I$, where $i = 1 \ldots 9$), for this particular linguistic variable.

5.3. Grade of Membership Function

The membership function is the characteristic function for fuzzy sets. In this proposed global routing procedure, an overlapping trapezoidal nature is best suited because moderately sensitive information with a higher grade of membership value may also be considered as a highly sensitive information with a lesser grade of membership value. A Gaussian function is not well fitted here because of its different response at each point and only one maximum response. But, our problem specification is that of having a particular response over a range.

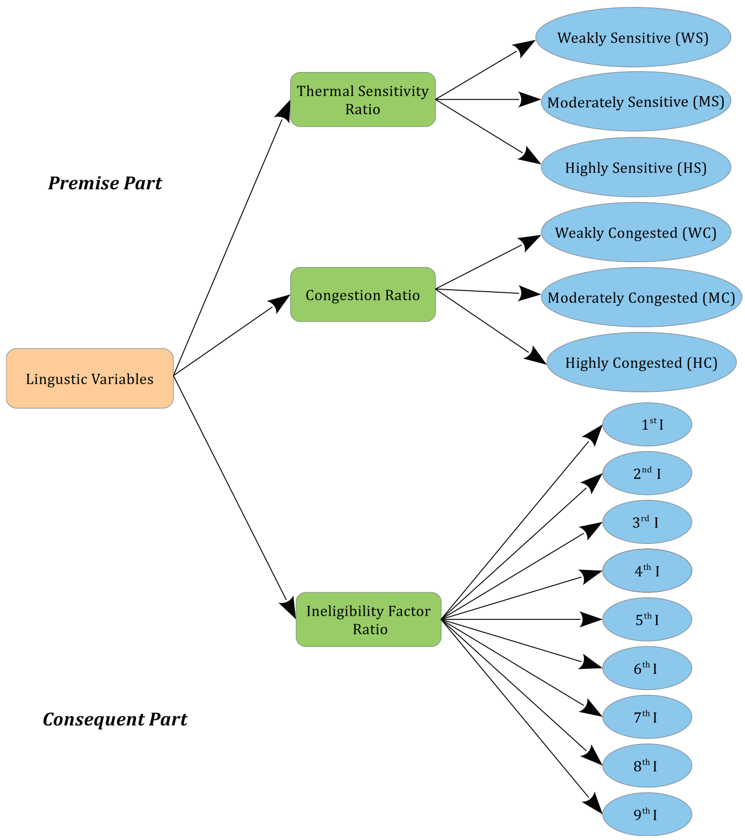


Fig. 5. Derived fuzzy sets from linguistic variables.

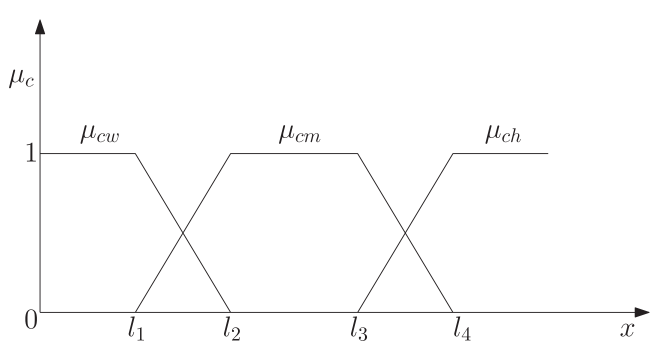


Fig. 6. The graph corresponding to the grade of membership values for sensitivity and congestion ratio.

The grade of membership values (shows in Figure 6) for three fuzzy sets corresponding to a linguistic variable of the premise part will be found according to Eqs. (2), (3), and (4).

$$
\begin{aligned}
\mu_{ch} &= 0 && for\, x < l_3 \\
&= 1 && for\, l_4 \leq x \leq 1 \\
&= (x - l_3)/(l_4 - l_3) && for\, l_3 < x < l_4.
\end{aligned} \tag{2}
$$

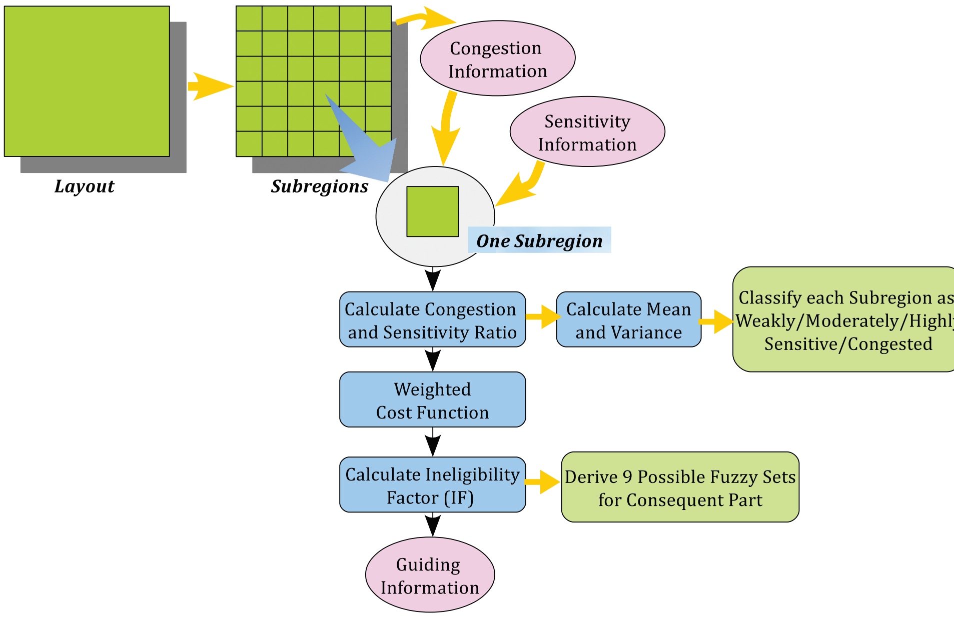


Fig. 7. Graphic representation of guiding information generation.

$$\begin{aligned}\mu_{cm} &= 0 && for\, x < l_1 \,\&\, x > l_4\\ &= 1 && for\, l_2 \le x \le l_3\\ &= (x-l_1)/(l_2-l_1) && for\, l_1 < x < l_2\\ &= (l_4-x)/(l_4-l_3) && for\, l_3 < x < l_4.\end{aligned} \tag{3}$$

$$\begin{aligned}\mu_{cw} &= 0 && for\, x > l_2\\ &= 1 && for\, 0 \le x \le l_1\\ &= (l_2-x)/(l_2-l_1) && for\, l_1 < x < l_2.\end{aligned} \tag{4}$$

The different boundary values l_1, l_2, l_3, l_4 in Eqs. (2)–(4) are dynamic for the premise part and generated in the guiding information generation algorithm. The step-by-step procedure for determining l_1, l_2, l_3, l_4 for thermal sensitivity and congestion information is stated in Section 5.4.1. Similarly, for the consequent part, a total of nine trapezoids are in [0,1] range and the boundary values of each are also dynamically generated depending upon the proposed rule base (described later).

5.4. Guiding Information Generation

The fuzzified global routing approach bifurcates in guiding information generation and in the main routing procedure. The first part corresponds to generation of relevant information regarding a specific unit of layout, called the subregion, and generating a fuzzy expert system. The overall procedure of guiding information generation is presented in Figure 7.

This part is executed immediately after the placement phase for each layer separately. The procedure works here as guided routing which will be further fed to the fuzzy expert system to take a decision during global routing between a source and destination. In Algorithm 1, the procedure *Generate_Guiding_Info()* generates the guiding

ALGORITHM 1: Guiding information generation for global router.

Generate_Guiding_Info( )

Input : S = Set of sensitivity information for each module,
P = Total placement information,
O = Obstacle information,
V = Parameter values,
L = Layer number
Output: GI = Generated guiding information

```
begin
    Layerwise_Place_Info();   /* Particular placement information for L layer from P */
    Num_Ter = Total_Terminals(P);   /* Calculating number of terminals for layer L */
    Get_Subregion_Information(P);
    SRN = Total_Subregion(L);   /* Number of subregions in layer L */
    for i ≤ SRN do
        AC = Area_Covered(O); /* Calculating total area covered by obstacles for i^th subregion of
        layer L */
        WSn = Weighted_Sensitivity(O, S); /* Calculating weighted sensitivity for each subrefor
        i^th subregion of layer L*/
        CR = AC / Total_Area();
        SR = WSn / Total_Area();
        IF = SR × V → α + CR × V → β;
        Add_to_GuidingInfo();   /* Add CR, SR and IF to GI */
    end
    Get_GuidingInfo();   /* Add mean and variance of CR and SR to GI for all subregions*/
end
```

information as mentioned. This algorithm is called from the *Generate_Routing_Path()* with a particular layer number. During execution of this algorithm, the total layout is divided into a 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, a normalized weighted average sensitivity ratio and normalized congestion ratio are generated. Mean and variance for all subregions are used to determine the boundary values of membership functions for the corresponding fuzzy sets.

The boundary values of *highly sensitive* (HS), *moderately sensitive* (MS), *weakly sensitive* (WS), *highly congested* (HC), *moderately congested* (MC), and *weakly congested* (WC) are determined during guiding information generation, as described next.

5.4.1. Description of the Algorithm. Due to the standard cell structure, the implementation of this algorithm is quite simple. The congestion estimation and total area estimation for each subregion may be calculated easily from obstacle information (O) and placement information (P) for that specific layer. After the layer-wise placement information is obtained, the area covered by obstacles (AC) and the weighted sensitivity (WSn) are calculated. Congestion and sensitivity ratio (CR and SR) information are also determined for each subregion.

Classification of sensitivity information. The mean(s') and variance (v_s) of thermal sensitivity for that particular layer are determined by Eqs. (5) and (6), where N = the total number of subregions and s_r = average sensitivity information of the r^{th} subregion.

$$s' = \frac{1}{N}\sum_{r=1}^{N}(s_r). \tag{5}$$

$$v_s = \frac{1}{N}\sum_{r=1}^{N}(s_r - s')^2. \tag{6}$$

The standard deviation for thermal sensitivity of the total region is $(d_s) = \sqrt{v_s}$. Hence, r^{th} subregion will be recognised as *highly sensitive* (HS), *moderately sensitive* (MS), or *weakly sensitive* (WS). The boundary values are found according to Eqs. (2) and (3), and (4) are stated follows.

(1) $l_1 = s' - \frac{3d_s}{2}$
(2) $l_2 = s' - \frac{d_s}{2}$
(3) $l_3 = s' + \frac{d_s}{2}$
(4) $l_4 = s' + \frac{3d_s}{2}$

Now the classifications for each subregion according to thermal sensitivity would be as follows.

(1) $s_r \geq s' + \frac{3d_s}{2}$, the subregion is HS
(2) $s' + d_s < s_r < s' + \frac{3d_s}{2}$, the subregion is HS with higher grade of membership and MS with lower grade of membership
(3) $s' + \frac{d_s}{2} < s_r < s' + d_s$, the subregion is MS with higher grade of membership and HS with lower grade of membership
(4) $s' - \frac{d_s}{2} < s_r \leq s' + \frac{d_s}{2}$, the subregion is MS
(5) $s' - d_s < s_r < s' - \frac{d_s}{2}$, the subregion is MS with higher grade of membership and WS with lower grade of membership
(6) $s' - \frac{3d_s}{2} < s_r < s' - d_s$, the subregion is WS with higher grade of membership and MS with lower grade of membership
(7) $0 < s_r \leq s' - \frac{3d_s}{2}$, the subregion is WS

So, coarsely, the classification can be presented as follows.

(1) $s_r \geq s' + d_s$, the subregion is HS
(2) $s' - d_s < s_r < s' + d_s$, the subregion is MS
(3) $0 < s_r \leq s' - d_s$, the subregion is WS

Classification of congestion ratio information. Consequently, the mean(o') and variance(v_o) of congestion ratio information are stated in Eqs.(7) and (8), where N = the total number of subregions and o_r = the average congestion information of the r^{th} subregion.

$$o' = \frac{1}{N}\sum_{r=1}^{N}(o_r). \tag{7}$$

$$v_o = \frac{1}{N}\sum_{r=1}^{N}(o_r - o')^2. \tag{8}$$

Similarly, the standard deviation for congestion information of the total region(d_o) = $\sqrt{v_o}$. Hence the r^{th} subregion will be recognized as *highly congested* (HC), or *moderately congested* (MC), or *weakly congested* (WC). Here also, the boundary values are according to Eqs. (2)–(4) and would be as follows.

(1) $l_1 = o' - \frac{3d_o}{2}$.
(2) $l_2 = o' - \frac{d_o}{2}$.
(3) $l_3 = o' + \frac{d_o}{2}$.
(4) $l_4 = o' + \frac{3d_o}{2}$.

Now the classifications for each subregion according to congestion ratio information are as follows.

(1) $o_r \geq o' + \frac{3d_o}{2}$, the subregion is HC.
(2) $o' + d_o < o_r < o' + \frac{3d_o}{2}$, the subregion is HC with higher grade of membership and MC with lower grade of membership.
(3) $o' + \frac{d_o}{2} < o_r < o' + d_o$, the subregion is MC with higher grade of membership and HC with lower grade of membership.
(4) $o' - \frac{d_o}{2} < o_r \leq o' + \frac{d_o}{2}$, the subregion is MC.
(5) $o' - d_o < o_r < o' - \frac{d_o}{2}$, the subregion is MC with higher grade of membership and WC with lower grade of membership.
(6) $o' - \frac{3d_o}{2} < o_r < o' - d_o$, the subregion is WC with higher grade of membership and MC with lower grade of membership.
(7) $0 < o_r \leq o' - \frac{3d_o}{2}$, the subregion is WC.

Here, coarse classification will be as follows.

(1) $o_r \geq o' + d_o$, the subregion is HC.
(2) $o' - d_o < o_r < o' + d_o$, the subregion is MC.
(3) $0 < o_r \leq o' - d_o$, the subregion is WC.

So, before routing is started, prior information is generated related to each subregion that guides the global routing procedure further. And the total guiding information is dynamic in nature, that is, the specified boundary values are determined during the execution of the routing procedure. Different dynamic fuzzy sets with different boundary values may be produced per layer present in the three- dimensional placement.

5.5. Proposed Rule Base

A rule base is an important factor during constructing a fuzzy expert system to define behavior of the system. There exist several rule models for this purpose. A TS model is computationally efficient and works well in optimization and adaptive techniques. The consequent part of the rules are not fuzzy, so less intuitive. The Mamdani model is computationally less efficient, but intuitive and well suited for human input. So, it has widespread acceptance. In the expert system this model is used to generate a rule base for total fuzzification of the process. The ineligibility weight factor is determined by a mathematical function. So, a conversation between the mathematical function to fuzzy sets is required.

Here, the TS model rule structure is IF s_r is A_j and o_r is B_k THEN $z = f(.)$. The representation of mathematical function $f(.)$ follows Eq. (9), where $\alpha + \beta = 1$.

$$f(.) = \frac{(\alpha \times s_r + \beta \times o_r)}{(\alpha + \beta)}. \tag{9}$$

The ineligibility weight factor (μ_r) of the consequent part depends upon the sensitivity ratio (s_r) and congestion ratio (o_r). The formula for stating the corresponding weight factor is formulated in Eq. (10).

$$\mu_r = f(\alpha, s_r, \beta, o_r) = \frac{(\alpha \times s_r + \beta \times o_r)}{(\alpha + \beta)}. \tag{10}$$

Here $\alpha + \beta = 1$. The preferable values for α, β are determined according to the requirement of the objective function. Definition of the fuzzy set for each rule in the consequent part is generated by putting the lower and upper limits for each fuzzy set of the premise part in the preceding function to determine the lower and upper limits, respectively. In this way the proposed rule base can be represented according to the Mamdani model. The proposed rule base consists of 9 rules with 9 fuzzy sets corresponding to a linguistic variable of the consequent part and 6 fuzzy sets corresponding to two linguistic variables of the premise part. The rule base considers all possible rules with all fuzzy sets of the premise part. The procedure of converting the TS model to the Mamdani is represented in Figure 8 and Eq. (11) for better understanding. The membership function's characteristics of trapezoidal fuzzy sets in the consequent part for the rule base are shown in Figure 9. The proposed rule base is stated in Table I.

$$\begin{aligned} a &= \frac{a_1 \times \alpha + a_2 \times \beta}{\alpha + \beta}, \\ b &= \frac{b_1 \times \alpha + b_2 \times \beta}{\alpha + \beta}, \\ c &= \frac{c_1 \times \alpha + c_2 \times \beta}{\alpha + \beta}, \\ d &= \frac{d_1 \times \alpha + d_2 \times \beta}{\alpha + \beta}. \end{aligned} \tag{11}$$

5.6. Proposed Fuzzy Expert System

In Algorithm 2, the *Fuzzy_Expert_System()* procedure constructs different fuzzy expert systems for different layers with same proposed rule base but different membership functions. Here, fuzzy sets for the antecedent or premise part and consequent part

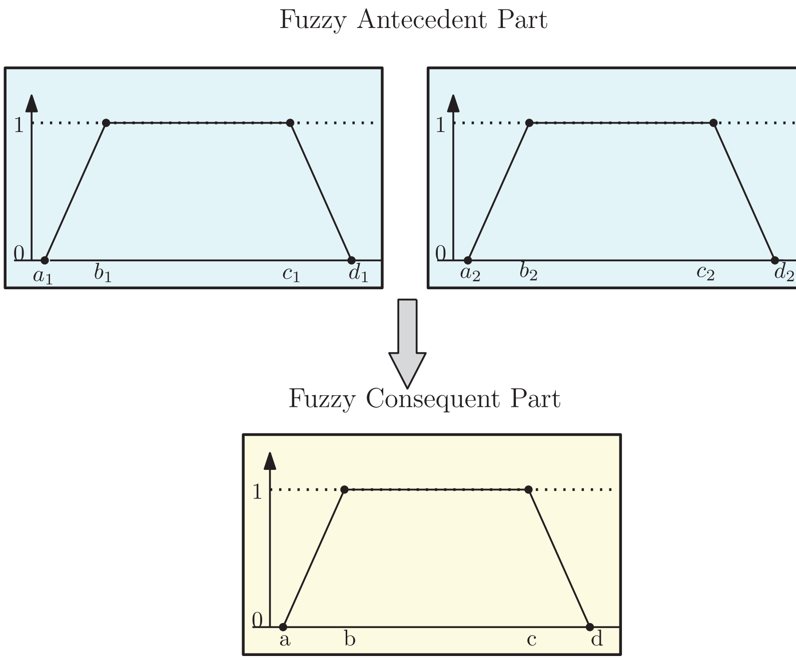


Fig. 8. Conversion procedure from TS model to Mamdani model.

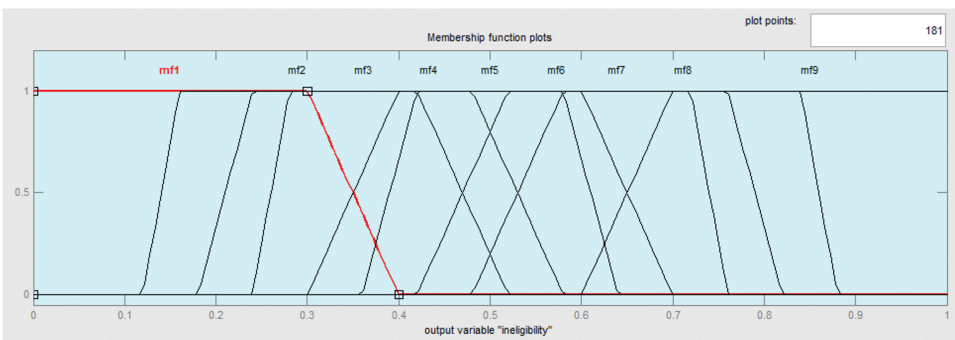


Fig. 9. Plotted membership functions of 9 fuzzy sets of the consequent part in the rule base.

Table I. The Proposed Original Rule Base for Inter-Layer Net Routing

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

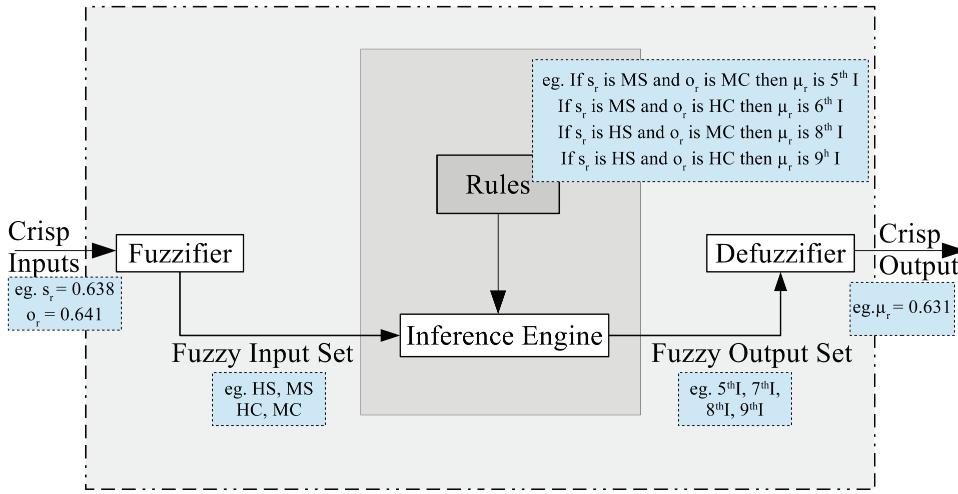


Fig. 10. Modeling of fuzzy expert system for three-dimensional structure.

ALGORITHM 2: Modeling the fuzzy expert system.

Fuzzy_Expert_System( )

Input: GI = Information of premise and consequent part)

begin
Mem_Fun_Premise(); /* Build membership functions for premise part from *GI* */
Mem_Fun_Consequent(); /* Build membership functions for consequent part from *GI* */
Gen_Rule_Base(); Build a rule base with 9 rules */
end

are obtained from the generated guiding information. This fuzzy expert system produces a crisp output for the consequent part, depending upon the inserted values in the antecedent part and parameter values. Here the proposed fuzzy expert system works dynamically by considering different boundary values for different fuzzy sets for different circuits. The total fuzzifcation and defuzzification are done inside the expert system according to the membership function and the proposed rule base. The block diagram of the proposed expert system is presented in Figure 10.

6. PROPOSED 3D PLACEMENT TECHNIQUE

6.1. Thermal-Aware Modeling

Thermal effects have prominent impact on performance and reliability of a chip. Due to closer proximity of the modules in a 3D multilayered structure, 3D ICs are significantly dependent on thermal parameters, that is, power dissipation issues. In their relevant work, Tsai and Kang [2000] have successfully addressed this issue in the case of standard cell placement and have come up with an efficient placement technique to reduce hotspots (generated by high heat flux in localized regions) during placement without compromising traditional design metrics, such as area and wirelength. Moreover, in their paper they have first successfully pointed out that it is the power density (and not the power dissipation) that may be treated as the most effective measurement parameter to account for thermal effects.

6.1.1. Analytical Die Temperature Model. As power density of the modules is a very effective parameter [Tsai and Kang 2000] in thermal profiling of a placed layer compared to power dissipation by the module itself, it was really necessary to figure out a directly measurable parameter that can be taken as a measure of this effect and originates as a direct after-effect of this issue. In another pioneering work, Im and Banerjee [2000] have successfully modeled this thermal effect and have shown that the die temperature is solely dependent upon the power density and that the change in this temperature is linearly dependent on this. Therefore, the change in die temperature may be considered as a directly measurable parameter to characterize the thermal profile of the die.

A simple analytical model has been proposed to estimate the change in temperature in each active layer of 3D chips. The temperature rise (above the ambient temperature) of the j^{th} active layer in an n-layer 3D chip may be expressed as

$$\triangle T_j = \sum_{i=1}^{j} \left[R_i \left(\sum_{k=i}^{n} \frac{P_k}{A} \right) \right]. \tag{12}$$

where n is the total number of active layers, R_i represents thermal resistance between the i^{th} and $(i-1)^{th}$ layers, and P_k is the power dissipation in the k^{th} layer. Assuming identical power dissipation (P) in each layer and identical thermal resistance (R) between layers, the temperature rise of the uppermost (n^{th}) layer in an n-layer 3D chip can be expressed as

$$\triangle T_n = \left(\frac{P}{A} \right) \left[\frac{R}{2} n^2 + \left(R_1 - \frac{R}{2} \right) n \right]. \tag{13}$$

where R_1 is mostly due to the package thermal resistance between the first layer and the heat sink and R is the thermal resistance between the i^{th} and the $(i-1)^{th}$ layers, respectively.

6.1.2. Dynamic Thermal Modeling. In the proposed thermal arrangement during placement, the non-uniformity of thermal conditions of modules was modeled in terms of the intrinsic parameter of placement, called $ThermalBound$, reported in Ghosal et al. [2009]. The probabilistic switching of modules is considered to satisfy a Poisson distribution [Press et al. 2001]. Hotspots are generally workload dependent and their duration may vary with switching activity of the circuits. The Poisson distribution counts the number of discrete occurrences of an event during a specified time interval. In Ghosal et al. [2009], switching of a module is considered as an occurrence of an event where λ denotes the average number of switchings for a set of logic modules. Therefore the probability of a module switching exactly k times is given by

$$\xi(k, \lambda) = \frac{e^{-\lambda} \times \lambda^k}{k!}. \tag{14}$$

where k is a nonnegative integer. It is clear that ξ accounts for the number of switchings of the corresponding module and lies between [0,1]. The actual number of times a module switches for average values λ is given by $\xi(k, \lambda) \times k$. In absence of any probability of switching, a module will switch exactly k times in a period of k cycles.

6.2. Proposed Algorithm

To date design solutions for implementing a pure 3D integrated circuit, are still not readily available. Available benchmarks in 3D also come with multiple metal layers only in spite of multiple device layers. Due to this unavailability, some 2D-to-3D placer tools have been used to study the feasibility of the proposed approach. The output of one

ALGORITHM 3: Generation of full 3D placed circuits from standard benchmark circuits.

Modified_3D_Placement( )

```
Input  : N = Number of cells to be placed,
         L = Number of layers available,
         A_r = Aspect ratio of the chip,
         S_t = sub-matrix order,
         I_t = Maximum iteration limit
Output: Thermally optimized 3D placement

begin
    Generate_3D(); /* Generate 3D matrix of minimum dimension satisfying constraints */
    Allocate_3D(); /* Allocate cells with corresponding power density values */
    while optimization_possible = yes do
        for layer_number = 1 to L do
            Layer_Therm_Opt(); /* Layer wise thermal optimization */
        end
        Inter_Layer_Therm_Opt(); /* Thermal optimization across all the layers */
        if optimized = no then
            Find_Layers(); /* Determine the layers where 2D optimization is necessary */
        else
            Stretch_Layers(); /* Stretching routing regions in each layers to eliminate
            overlapping in cells */
        end
    end
end
```

efficient thermal-aware 3D placer, reported in Ghosal et al. [2010b, 2010c], has been used in the present work with some necessary and suitable modification. An intelligent heuristic has been proposed for this placement migration. In this proposed approach, an efficient overlap elimination technique has been introduced as a post-placement procedure to optimize and improve the placement results. Pseudocode of the proposed placement algorithm has been presented in Algorithm 3.

7. PROPOSED SCHEME FOR GLOBAL ROUTING

The overall flow of the proposed fuzzified global routing tool for 3D ICs may be represented as in the flowchart in Figure 11. This fuzzified routing technique proceeds by designing a fuzzy logic expert system for generating guiding information that helps in decision-making during the actual global routing phase. The placed region of an individual layer is divided into subregions. The proposed heuristic is based on a sensitivity information value for each node (that signifies the thermal response) and congestion information to specify the congestion-driven technique for routing.

For each subregion, the crisp values of sensitivity and congestion information are transformed in two fuzzy sets. These are used to generate a crisp output by passing through a rule-base-based on the Mamdani model and by the defuzzification method. Thus the total fuzzy expert system works for generating the guiding ineligiility factor generation for each subregion. During decision-making of the routing procedure, the subregion with minimum ineligibility factor is favoured.

Our proposed routing procedure does not allow detours and the global routing paths are generated in units of subregions. The unroutable nets will again start from the guiding information generation step during the rip-up-and-reroute procedure. Specific

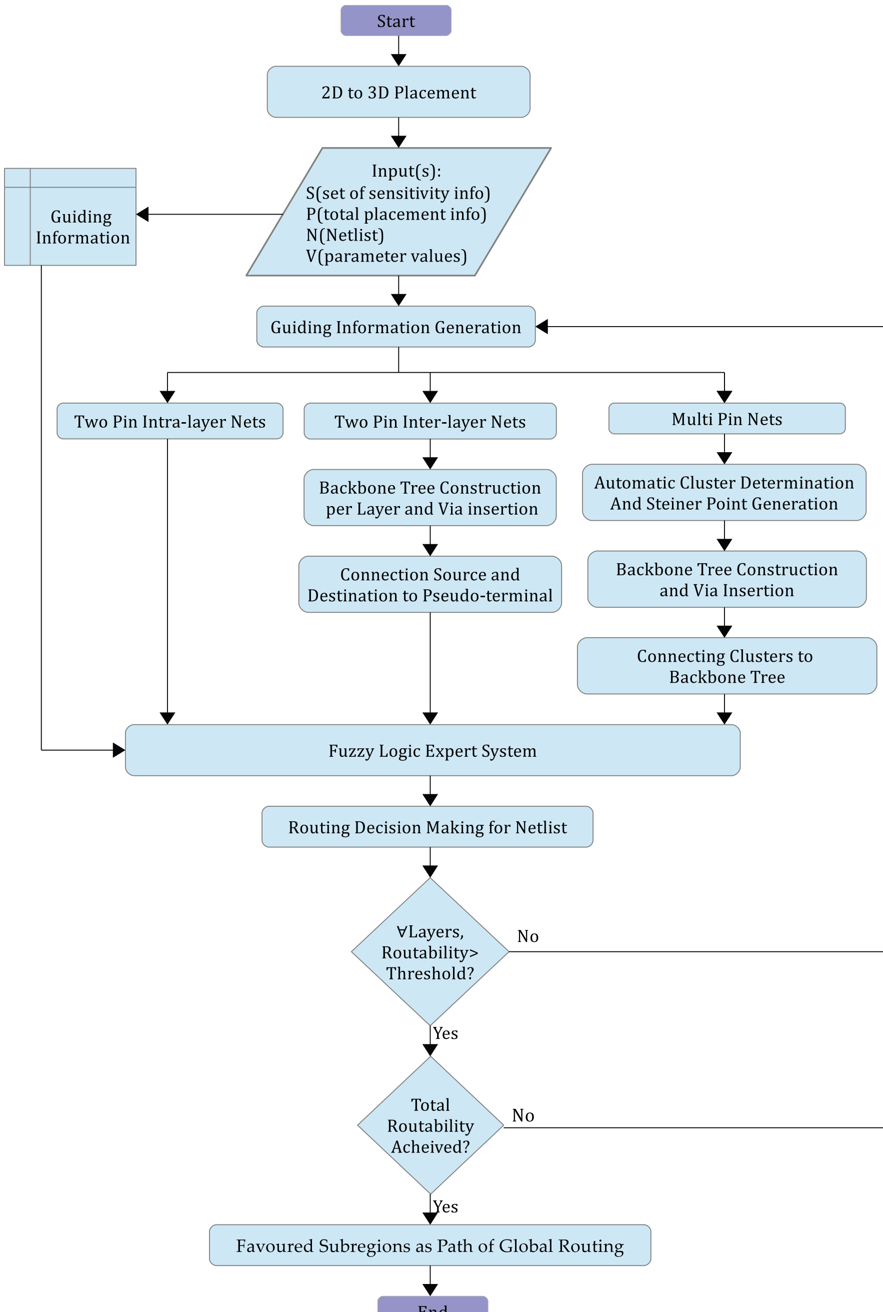


Fig. 11. Overall flow of the proposed fuzzified global router for 3D ICs.

geometric paths are generated during detailed routing. Another aspect of the proposed global routing is that if one subregion is selected during routing of a particular net, then that specific subregion will never be retraced again for the same net. After routing one net, the ineligibility factor gets increased by a factor for the subregions which the

ALGORITHM 4: Global routing path generation for fuzzified router.

Intra_Layer_Routing( )

Input : *Src* = Source Coordinate,
D = Destination Coordinate,
GI = Guiding Information,
V = Parameter value
Output: Favoured subregions(*Subregion*)

```
begin
    MDSrc = MD(Src, D);  /* Manhattan distance between Src and D containing subregions */
    MD = MDSrc;
    PSrc = Src;  /* Setting a pivot subregion */ Fuzzy_Expert_System (GI);
    while MD > 0 do
        Explore_Neighbour();  /* Explore four neighbors around pivot subregion */
          NInfo = Get_Info();  /* Sensitivity, congestion information of each neighbors */
          IF = Defuzz();  /* Defuzzified value by centroid method using rule base */
        if Nbour_Not_Visited() then
              /* Check whether a neighbor is visited or not */
            PSrc = Select_Neighbour();  /* Select neighbor with min IF and set as new pivot */
        end
        MD = MD(PSrc, D);
    end
end
```

routed path has gone through, and either vertical or horizontal capacity gets decreased for that subregion.

Different nets of a particular netlist are processed sequentially for the sake of easy implementation. One prerouted net will contribute to increasing congestion ratio information in this sequential order of global routing.

(1) *Two-pin intra-layer nets.* Here connections are made in a normal fuzzified way according to Algorithm 4.
(2) *Two-pin inter-layer nets.* Multilayered two-pin connections are made by inserting pseudo-terminal points.
  —First, the backbone tree construction goes across multiple layers that specify the pseudo-terminals for each layer and inject vias for inter-layer communication. Determination of pseudo-terminals is based on the sensitivity and congestion ratios of the subregions containing the source and destination terminal.
  —The second step consists of connecting the pseudo-terminal points to the source as well as the destination containing subregions in that particular layer by two-pin intra-layer net connection in a fuzzified way, according to Algorithm 4.
(3) *Multi-pin nets.* Multilayered or single-layered Steiner tree construction is done by insertion of a pseudo-terminal. The three steps are as follows.
  —First, numbers of clusters are generated by an automatic cluster determination technique [Bandyopadhyay 2005] (required for larger nets) by a simulated annealing method. The centers of clusters act as Steiner points during generation of intra-cluster Steiner trees for each layer. Connection between the cluster center and each terminal is done by Algorithm 4.
  —Second, the backbone tree construction is done across multiple layers that specify the pseudo-terminals for each layer and inject vias for inter-layer communication.

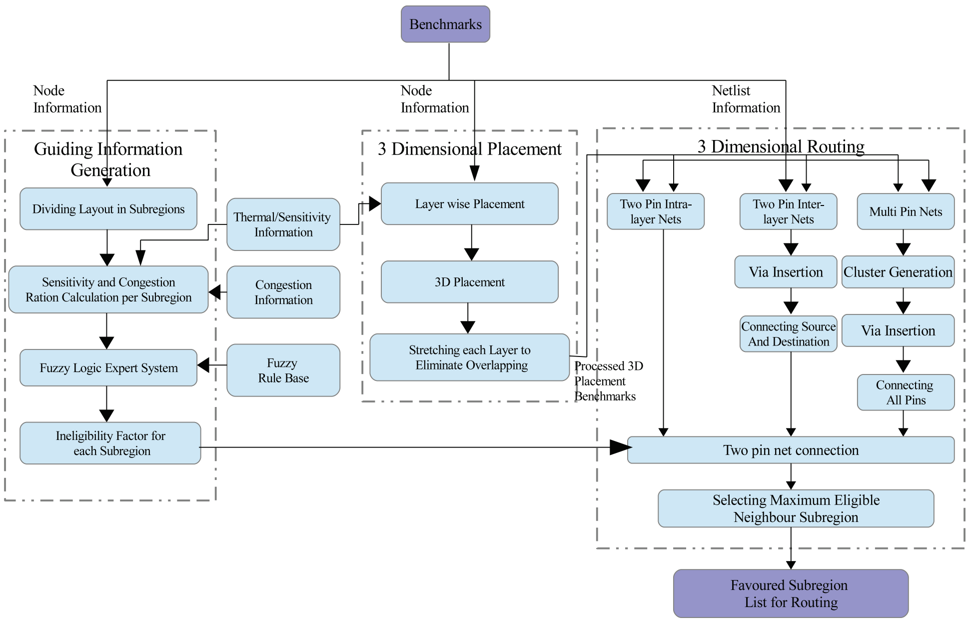


Fig. 12. The proposed thermally efficient congestion-free global routing approach for 3D ICs.

Determination of pseudo-terminals is based on the sensitivity and congestion ratios of the subregions. One obstacle avoidance heuristic is used here during insertion of inter-layer vias.

—The third step connects each cluster center or single terminal with the backbone tree, that is, the pseudo-terminal points in that particular layer by two-pin net connection in a fuzzified way, as stated in Algorithm 4.

The detailed working principle of the total procedure is described later. One obstacle avoidance heuristic is used here during insertion of inter-layer vias.

8. PROPOSED FUZZIFIED APPROACH FOR GLOBAL ROUTING

The proposed procedure *Generate_Routing_Path()* generates a total routing path for all nets in terms of subregions for a netlist of a circuit in Algorithm 5. It takes the output of Algorithm 3 for a circuit as its input and produces a global routing path as output for the netlist of the same circuit. At the start of Algorithm 5, the generation of subregion information, guiding information, and building of the fuzzy expert system is done through two procedures, *Generate_Guiding_Info()* and *Fuzzy_Expert_System()*, per layer. Then it heads towards the routing procedure. The proposed algorithm recognises several types of nets and performs the global routing accordingly. The functional-level block diagram for the total router is presented in Figure 12.

8.1. Two-Pin Intra-Layer Net Routing

In Algorithm 5, the procedure *Generate_Routing_Path()* first recognises all two-pin intra-layer nets and performs global routing in a fuzzified way as stated in Algorithm 4. Here, the *Intra_Layer_Routing()* procedure generates the favoured subregions for connections between two pins. During the decision-making of selection of the next favoured subregion, one subregion is never reselected for the same net routing.

ALGORITHM 5: Algorithm for generation of routing path for a netlist.

Generate_Routing_Path( )

Input : S = Set of sensitivity information for each module,
P = The total placement information,
N = Netlist,
V = Parameter values
Output: Routing path in terms of favoured subregions

```
begin
    L = No_of_Layers(P);
    for layer_number = 1 to L do
        Generate_Guiding_Info(S, P, V); /* Guiding information generation for all layers */
        Fuzzy_Expert_System(); /* Build distinct fuzzy expert systems for all layers */
    end
    for net ∈ N do
        if intra_layer_two_pin_net then
            Intra_Layer_Routing(); /* Generate routing path from source to destination */
        else if inter_layer_two_pin_net then
            Backbone_Tree(); /* Determine pseudo terminal positions by constructing backbone
            tree */
            Pseudo_Terminal_Insertion(); /* Insertion of pseudo terminals and via */
            Connect_All_Pins(); /* Connecting source and destination pins to the pseudo
            terminal points */
        else
            /* For multi-pin intra and inter layer nets */

            Automatic_Cluster_Generation(); /*Generate Automatic number of clusters with
            multiple pins for all layers */
            Backbone_Tree(); /* Determine pseudo terminal positions by constructing backbone
            tree */
            Pseudo_Terminal_Insertion(); /* Insertion of pseudo terminals and via */
            Connect_All_Pins(); /* Connecting all pins of a net to the pseudo terminal points */
        end
    end
end
```

When the destination is a set of terminals, then this algorithm finds a favoured path for each terminal belonging to the destination set from the specific source. Here decision-making is done using the generated fuzzy expert system depending upon different layers. One scenario of two-pin intra-layer net routing is shown pictorially in Figure 13.

8.2. Two-Pin Inter-Layer Net Routing

Next, all two-pin inter-layer nets are sorted out in Algorithm 5. To route all those recognised nets, two steps are followed.

8.2.1. Backbone Tree Construction and Determination of Pseudo-Terminal. The backbone tree performs inter-layer connection during connection between two intra-layer pins. The backbone tree generation process is done by the Backbone_Tree() function in Algorithm 5. Inter-layer vias are the technology to build a backbone tree here. A via insertion procedure can be done in two ways, as shown in Figure 14. The position of a

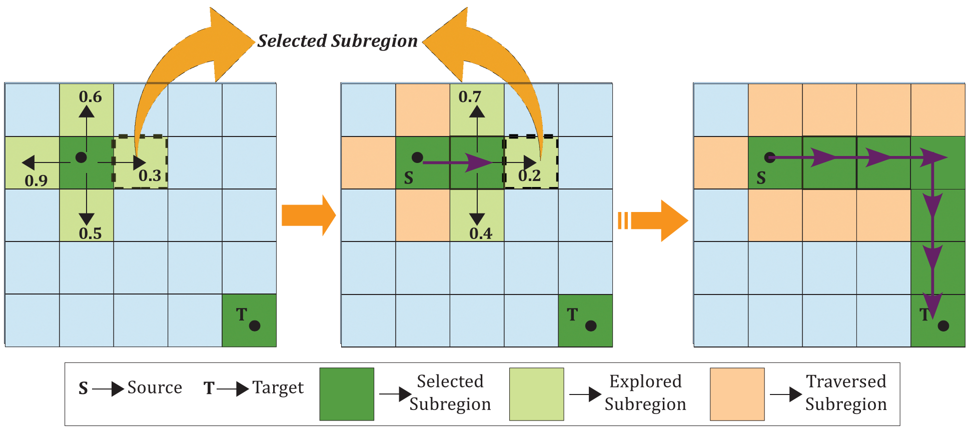


Fig. 13. A scenario of source-to-target intra-layer global routing.

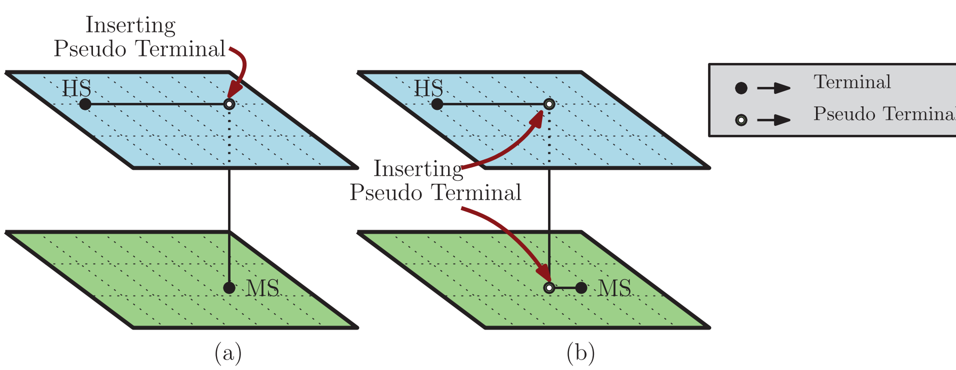


Fig. 14. Insertion of pseudo-terminal during inter-layer two-pin net routing: (a) normal approach; (b) proposed approach.

via insertion point should be inclined to the direction of a more sensitive or congested terminal.

In the figure, the normal approach shows a general method of the via insertion strategy during connection between an inter-layer source and destination. But the significance of our proposed approach lies in providing the more sensitive and congested terminal, giving a closer path for heat dissipation (through the via) while still giving less sensitive and congested ones some favour.

One example case may be taken, as shown in Figure 14, where the main priority is given to the highly sensitive terminal by placing the via position near to it in both approaches. But in the proposed approach, the moderately sensitive terminal also gets some extend of favour, which is not given in the normal approach. So, here in the proposed technique, as a backbone tree determination strategy, the combined sensitivity-congestion information for two terminals is used as stated in Eqs. (15)–(17).

$$sc_{rl} = \frac{s_{rl} \times \alpha + c_{rl} \times \beta}{\alpha + \beta}. \tag{15}$$

The average weight (sc_{rl}) for the r^{th} subregion in the l^{th} layer is determined in Eq. (15) where α, β are the two constraints provided by the user specification, and s_{rl} and c_{rl} are the sensitivity and congestion information of the r^{th} subregion residing on the l^{th}

layer. The x-coordinate (x_b) of the backbone for all layers for a particular two-pin net is determined in Eq. (16) and the y-coordinate(y_b) of the same is determined in Eq. (17). Here, src_i is the subregion containing the source terminal residing on the i^{th} layer and $dest_j$ is same for the destination terminal residing on the j^{th} layer. The sc_{si} and sc_{dj} are the average weight for the source- and destination-containing subregions. Then the backbone may be drawn through the (x_b, y_b) point intersecting all the layers in between.

$$x_b = \frac{src_i \rightarrow x \times sc_{si} + dest_j \rightarrow x \times sc_{dj}}{sc_{si} + sc_{dj}}. \tag{16}$$

$$y_b = \frac{src_i \rightarrow y \times sc_{si} + dest_j \rightarrow y \times sc_{dj}}{sc_{si} + sc_{dj}}. \tag{17}$$

Next, the final positions of pseudo-terminals are determined. In one layer, if there is no obstacle in the (x_b, y_b) point, that point works as a pseudo-terminal and participates in routing of that layer and one electrical via is inserted into that point according to the function Pseudo_Teminal_Insertion(). But if the point(x_b, y_b) falls on any obstacle in any layer, then one obstacle avoidance strategy has been introduced that considers weight factor information as the impacting factor during its avoidance technique.

8.2.2. Connecting Source and Destination to Backbone Tree. Connection between source and destination with the pseudo-terminals caused by the insertion of the backbone tree is done by the *Fuzzified_Intra_Layer_Routing()* procedure again. In the *Connect_All_Pins()* procedure, all routed subregion information is generated and returned to the user.

8.3. Multi-Terminal Net Routing

Lastly, the multi-terminal nets are recognized in Algorithm 5. The intra- and inter-layer multi-pin nets are handled in quite a similar approach during execution of the routing procedure. Multi-pin inter-layer nets may be considered as a generalization of multi-pin intra-layer nets with a single layer. So, the solution approach also takes this generalization concept. The three steps defined next are called repeatedly for each multi-pin net having inter- or intra-layer connection (for intra-layer nets, the layer number will be one).

8.3.1. Automatic Cluster Determination and Connection within Clusters. This step is mainly required to process a large net to a smaller one, that is, by making clusters of terminals with nearer proximity give the larger net a smaller look.

As the total procedure is fuzzified and aimed at escaping the bound of the 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 Bandyopadhyay [2005], has been selected. The center of the generated cluster works as pseudo-terminal for that layout and can be used further for routing.

Now, for each terminal(t) of the l^{th} layer, the corresponding home subregion(r) and its combined sensitivity-congestion ratio (sc_{rl}) from generated guiding information are determined. The combined value is determined according to Eq. (15), where α and β come from parameter values (V). For a typical implementation, the selected values may be 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.

The distance measure between two terminals is the *Manhattan distance* (MD) measurement in terms of subregions. Here automatically determined clusters may be of different sizes. Suppose that for the l^{th} layer the automatically determined number of clusters is N_l and the vertex of the i^{th} cluster in the l^{th} layer is v_{il}, where $i \in \{1...N_l\}$.

Now, for each cluster, the *Intra_Layer_Routing()* procedure is called that connects all the terminals of the cluster with the center vertex of the cluster.

For a large net also, the number of clusters is generally small. So, the number of inserted pseudo-terminals does not affect the cost significantly. Total automatic cluster determination and a connection within the clusters procedure is abstracted within the Automatic_Cluster_Generation() function in Algorithm 5.

8.3.2. Backbone Tree Construction and Determination of Pseudo-Terminals. Construction of backbone trees is mandatory for inter-layer via connections. In connecting points, vias are inserted in a particular layer whose position depends upon the behaviour of the generated cluster or terminals. The position of a via insertion point should be inclined to the direction of larger and more sensitive or congested clusters or terminals.

So, the backbone tree determination strategy combines both sensitivity-congestion information for each cluster or terminal and the size of the clusters (stated in Eqs. (18)–(20)).

In Eq. (18), the total summation of sc for each cluster is determined, where $c_{il} \rightarrow i^{th}$ cluster in the l^{th} layer.

$$sc_{il} = \sum_{\forall r \in i} sc_{rl}. \tag{18}$$

The x-coordinate (x_b) of the backbone for all layers for a particular net is determined in Eq. (19) and the y-coordinate(y_b) in Eq. (20). Here $vx_{il} \rightarrow x$-coordinate of the center of cluster i in the 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 the l^{th} layer. Then the backbone may be drawn through an (x_b,y_b) point intersecting all the layerss. Eqs. (19) and (20) are generalized from Eqs. (16) and(17).

$$x_b = \frac{\sum_{l=1}^{N} \sum_{i=1}^{N_l} (sc_{il} \times vx_{il})}{\sum_{l=1}^{N} \sum_{i=1}^{N_l} sc_{il}}. \tag{19}$$

$$y_b = \frac{\sum_{l=1}^{N} \sum_{i=1}^{N_l} (sc_{il} \times vy_{il})}{\sum_{l=1}^{N} \sum_{i=1}^{N_l} sc_{il}}. \tag{20}$$

The earlier described backbone tree construction procedure is done in the Backbone_Tree() procedure. Here, determination of the final position of the pseudo-terminals stage is done layer-wise. In one layer, if there is no obstacle in the (x_b, y_b) point, then that point works as a pseudo-terminal. That pseudo-terminal also participates in routing of that layer and one electrical via is inserted into that point according to function Pseudo_Teminal_Insertion().

But if the point (x_b, y_b) falls on any obstacle in any layer, then one obstacle avoidance strategy is used that also considers the cluster size and combined sensitivity-congestion ratio information as the impacting factor during its avoidance technique. The avoidance scheme is described later. One example case is shown pictorially in Figure 15 for better understanding. In this figure, it is clear that the insertion point of the pseudo-terminal is inclined to more congested and more sensitive terminals to be connected to minimize heat dissipation as well as wirelength.

8.3.3. Connecting Clusters to Backbone Tree. Connection between each cluster center with the pseudo-terminal caused by the insertion of the backbone tree is done in a Connect_All_Pin() procedure that calls the *Intra_Layer_Routing()* procedure internally.

Ultimately, in Algorithm 5 the *Generate_Routing_Path()* procedure will return favoured subregions as a global routing path.

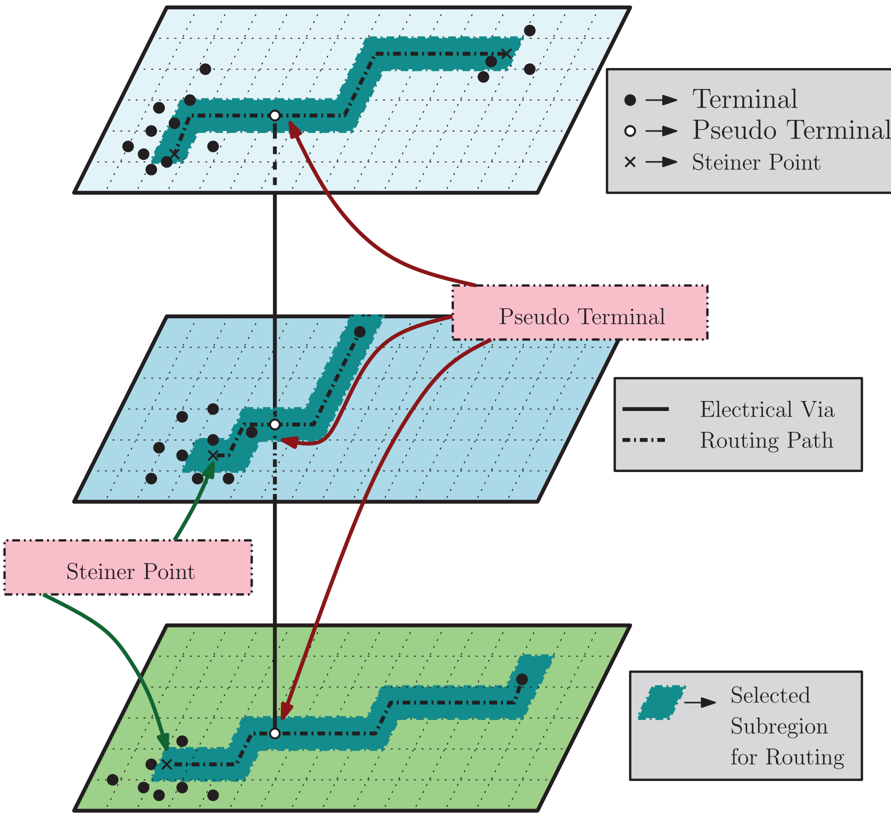


Fig. 15. Insertion of pseudo-terminal and backbone tree construction during multi-terminal inter-layer net routing.

8.4. Obstacle Avoidance Scheme during Determination of Pseudo-Terminal

The position of a pseudo-terminal, after avoiding an obstacle, depends upon the total weight factor of clusters falling on four possible sides of obstacles. Here, the position of the pseudo-terminal is skewed toward more congested, sensitive, and large clusters, so that congestion does not further increase to that side.

By injecting the pseudo-terminal, a vertical via position is actually inserted to help the electrical signal to pass quickly from the more sensitive and congested regions to other layers that may be favourable for routing. One example of a determination of a pseudo-terminal in an obstacle avoidance scheme is shown in Figure 16. The scheme is stated next.

(1) The weight factor related to cluster size and combined sensitivity-congestion ratio for the i^{th} cluster in the l^{th} layer is w_{il} as stated in Eq. (21), where n_{il} = number of terminals in the i^{th} cluster in the l^{th} layer and sc_{ikl} = the average sensitivity-congestion ratio of the corresponding subregion where the k^{th} terminal belongs in the i^{th} cluster in the l^{th} layer.

$$w_{il} = n_{il} \times \sum_{k=1}^{n_{il}} sc_{ikl}. \tag{21}$$

(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 coordinates of the four corner points

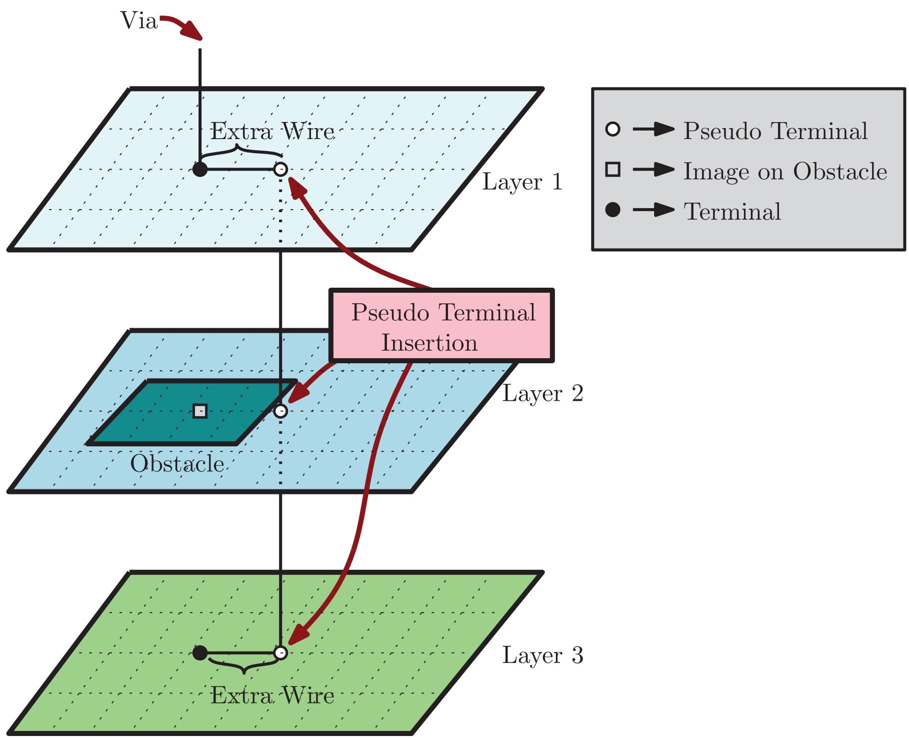


Fig. 16. Insertion of pseudo-terminal in presence of obstacle during backbone tree construction.

of the obstacle counterclockwise are then (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) We 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 determining the maximum value (m) among the four components according to Eq. (22), the final pseudo-terminal $(x_{b'}, y_{b'})$ position will be determined.

$$m = max\{m_1, m_2, m_3, m_4\}. \tag{22}$$

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 $(x_{b'}, y_{b'})$ works as the pseudo-terminal for that particular layer.

As a result of the obstacle avoidance scheme, the upper and lower layer of the obstacle-containing layer require to add two pseudo-terminals and some extra wire as shown in Figure 16. But this overhead does not much affect the out-come, as at present three-dimensional technology supports a maximum of up to 6–8 layers.

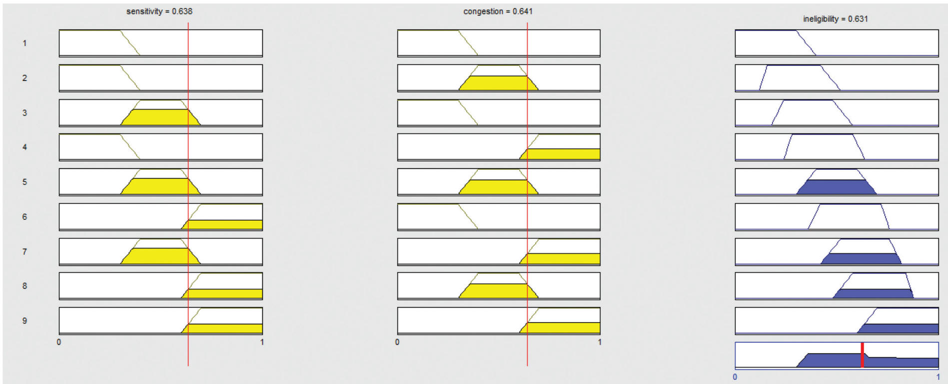


Fig. 17. The fuzzification and defuzzification with respect to rule base.

This total step-by-step approach for avoiding obstacles is done in the Pseudo_Terminal_Insertion() procedure, if required.

9. RESULTS AND DISCUSSION

9.1. Experimental Framework

Proposed algorithms have been implemented in C, Java, MatLab 7.14.0, MCR, MatLab Builder JA, and Swing. The GUI is designed in GUIDE and the fuzzy expert system is implemented using the MatLab fuzzy toolbox. The experiments were performed on a standard desktop environment of 4GB memory with an Intel chip running at 2.30GHz. The ISPD'98 (IBM-PLACE 2.0) benchmark suite [Alpert 1998] (for fixed die placement) is used. Later, the experiments were also implemented for ISPD'07 [Nam et al. 2007] and ISPD'08 [Nam et al. 2008] 3D benchmarks on an 8-core 2.0GHz Intel Xeon-based server with 8GB memory. Two different frameworks have been taken for comparability with other standard routers for different benchmark suites.

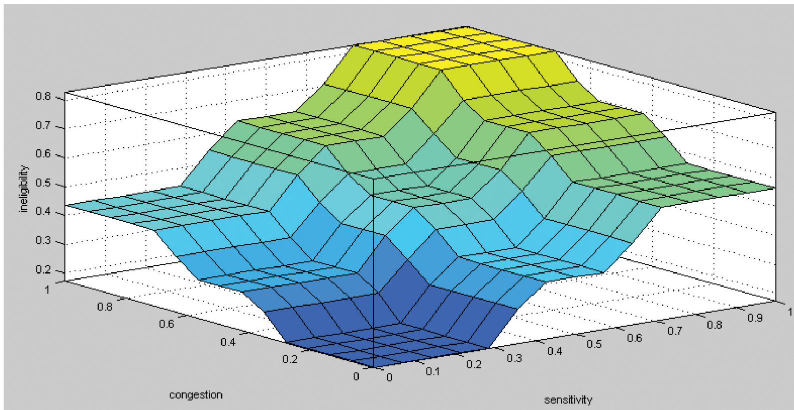
9.2. Feasibility Study of the Proposed Model

Figure 17 shows a snapshot of the implemented fuzzy expert system of a total implementation using the fuzzy toolbox in MatLab. For the crisp input set [0.638 0.641], the proposed rule base produces a crisp output value of 0.631. The different activated fuzzy sets for the premise part by two crisp inputs are the highlighted trapezoids that fall on the crisp line in the first two grids. For the consequent part, those highlighted trapezoids are on the crisp output line in the last grid. From the snapshots of Figure 18, it is clear that, using a rule base, the expert system produces quite accurate output, as the change of ineligibility factor is quite smooth here. The α and β values can actually be user specified, though we have taken $\alpha = 0.5$ and $\beta = 0.5$ in presented experimental result.

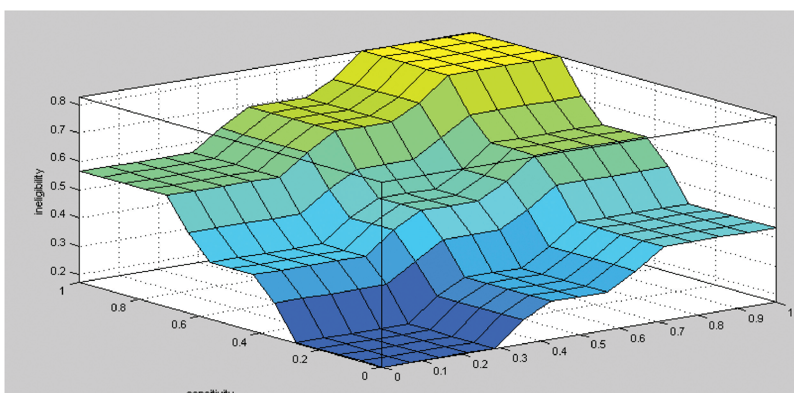
9.3. Implementation

After prior success in a feasibility study of the proposed approach, the total global routing implementation over standard benchmarks is performed.

In Table III, experimental results for IBM benchmarks for standard cells are reported. The implementation aims to minimize the time complexity for 3D ICs in a fuzzified way. 3D placement is generated by the proposed method for these 2D benchmarks. From the experimental results, it is clear that significantly less time is required for routing. The main concern of the algorithm is the time required for the guiding information



(a) ineligibility factor with congestion and sensitivity



(b) ineligibility factor with sensitivity and congestion

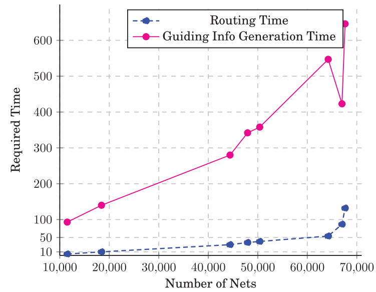
Fig. 18. Change of ineligibility factor with sensitivity, congestion for proposed rule base.

Table II. Variation in Resulting Metrics for IBM 02 (for 6 layers) with Different α and β Values

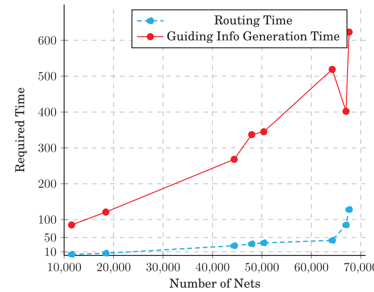
α	β	Guiding Info. Gen Time(sec.)	Routing Time(sec.)	Wirelength
[redacted]				
0.1	0.9	88.0	4.0	167091
0.2	0.8	72.0	10.0	166934
0.3	0.7	83.0	10.0	162913
0.4	0.6	76.0	4.0	168014
0.5	0.5	85.0	4.0	166922
0.6	0.4	91.0	11.0	166242
0.7	0.3	76.0	4.0	185364
0.8	0.2	74.0	10.0	161371
.9	.1	76.0	4.0	166171
[redacted]				

Table III. Experimental Benchmark Statistics for ISPD'98 Benchmarks

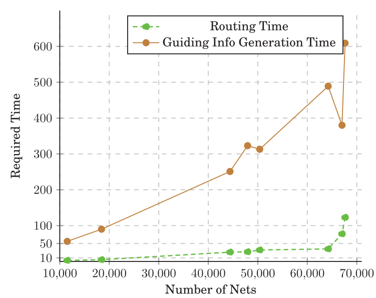
Benchmark	# Net	# Layer	Guiding Info. Gen. Time(sec.)	Routing Time(sec.)
ibm01	11507	3	93.0	4.0
		4	85.0	3.0
		5	56.0	3.0
		6	36.0	3.0
		7	28.0	2.0
		8	16.0	1.0
ibm02	18429	3	140.0	10.0
		4	121.0	6.0
		5	90.0	5.0
		6	85.0	4.0
		7	74.0	4.0
		8	60.0	2.0
ibm07	44394	3	280.0	30.0
		4	268.0	27.0
		5	251.0	26.0
		6	247.0	22.0
		7	240.0	22.0
		8	220.0	20.0
ibm08	47944	3	342.0	36.0
		4	337.0	32.0
		5	323.0	27.0
		6	307.0	26.0
		7	305.0	24.0
		8	297.0	21.0
ibm09	50393	3	358.0	39.0
		4	345.0	35.0
		5	313.0	32.0
		6	294.0	28.0
		7	283.0	23.0
		8	278.0	22.0
ibm10	64227	3	547.0	54.0
		4	519.0	42.0
		5	489.0	35.0
		6	485.0	33.0
		7	480.0	32.0
		8	472.0	32.0
ibm11	67016	3	423.0	87.0
		4	402.0	85.0
		5	380.0	77.0
		6	372.0	74.0
		7	370.0	67.0
		8	367.0	57.0
ibm12	67739	3	646.0	132.0
		4	623.0	128.0
		5	609.0	123.0
		6	605.0	120.0
		7	589.0	102.0
		8	582.0	96.0



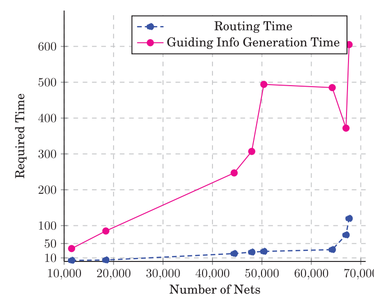
(a) Number of Layers = 3



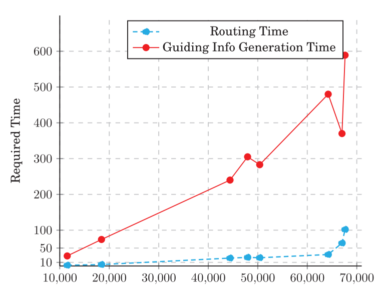
(b) Number of Layers = 4



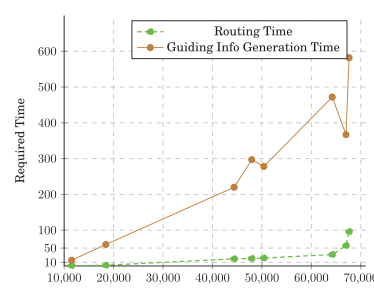
(c) Number of Layers = 5



(d) Number of Layers = 6



(e) Number of Layers = 7



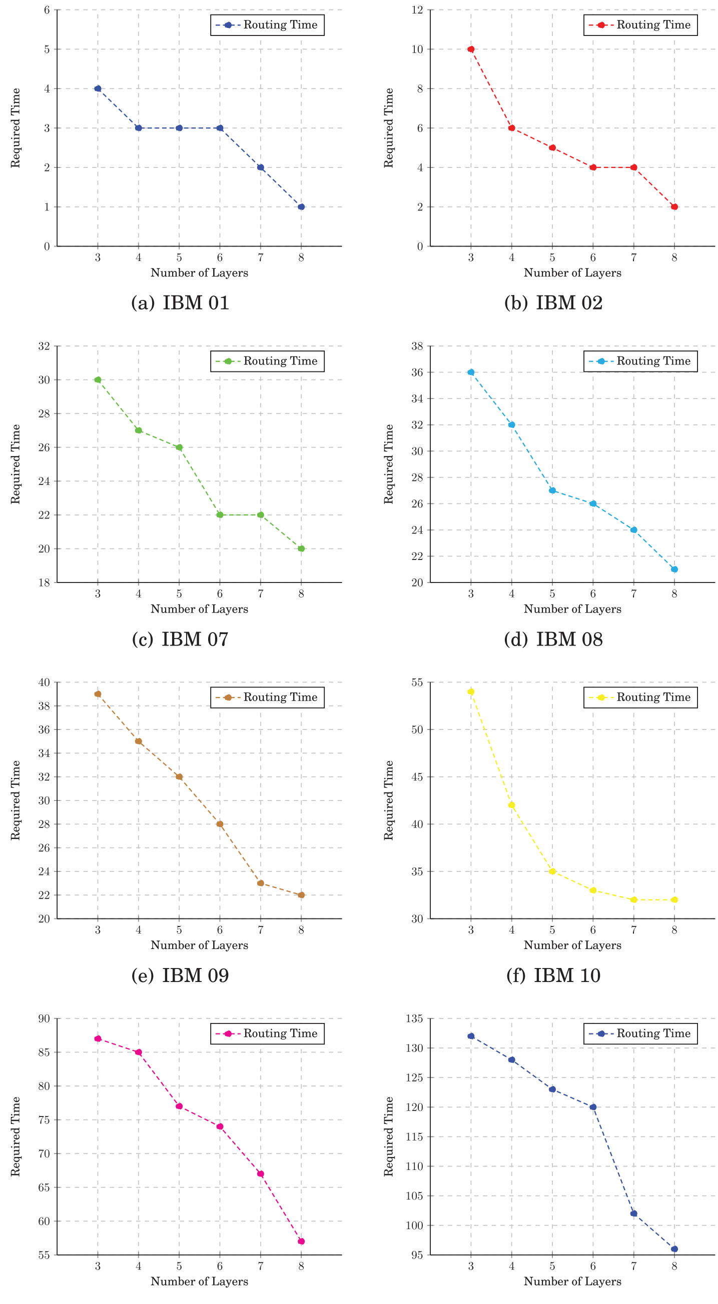
(f) Number of Layers = 8

Fig. 19. Variation in guiding information generation time and routing time for different ISPD'98 benchmark circuits (see Table III) with different numbers of layers.

generation phase. With a greater number of device layers in 3D placement, the routing time decreases for larger circuits. This characteristic conforms to the concept of 3D ICs. This work is aimed at comparing the fuzzified approach with other fool-proof routers based on an heuristics-based approach. The implementation is enriched consequently.

Experimental results for different IBM benchmarks with different device layers are reported in Table III. For different numbers of device layers, variations of guiding information generation time as well as routing time are plotted in Figure 19.

Variation of routing time with an increasing number of device layers for all benchmark circuits has been studied (see Table III) and plotted in Figure 20. Results show that routing time is significantly lower than guiding information generation time. Moreover, it even decreases with an increase in device layers for more availability of routing space in 3D ICs.



(a) IBM 01 (b) IBM 02

(c) IBM 07 (d) IBM 08

(e) IBM 09 (f) IBM 10

(g) IBM 11 (h) IBM 12

Fig. 20. Variation in routing time with available number of layers for different ISPD'98 benchmark circuits (see Table III).

Some important observations may be put in the form following Lemmas 9.1 and 9.2, and Observation 9.1.

LEMMA 9.1. *Better routing time may be achieved with a greater number of layers in 3D ICs.*

PROOF. With an increased number of device layers, routing space increases vertically. So some distant terminals of each net will be nearer and can be connected through vias. This leads to a decrease of routing time for the total netlist. □

LEMMA 9.2. *Time to generate guiding information decreases with an increasing number of device layers in 3D ICs.*

PROOF. An increasing number of device layers makes the layout area smaller and assigns fewer terminals for each layer. So, a lesser number of terminals will make each layer less complicated. Calculating sensitivity and congestion information will be easier for each layer during guiding information generation. Eventually, total guiding information generation time will decrease for a netlist with a greater number of device layers. □

OBSERVATION 9.1. *The rate of increase in routing time is much slower than the rate of increase in guiding information generation time with an increasing number of nets.*

9.4. Variation of Parameter Values

Here, α and β are the two controlling parameters which determine the objective function and therefore the final fuzzy expert system. An exhaustive analysis with variable α and β values for certain benchmarks has been presented in Table II. Two extreme scenarios can be named as follows.

(1) $\alpha = 0.0$ and $\beta = 1.0 \rightarrow$ Totally Congestion-Aware Routing
(2) $\alpha = 1.0$ and $\beta = 0.0 \rightarrow$ Totally Thermal-Aware Routing

It is clear from Table II that guiding information generation time varies within 72–91 seconds, routing time varies within 4–11 seconds, and wirelength varies within 161, 371–172, 575. Apparently, for a certain benchmark the variation in the result is not drastic for a variable set of parameter values. This quality signifies robustness of the proposed global router.

9.5. Comparison for ISPD'98 Benchmarks

The authors have performed the simulation of FuzzRoute in (mentioned in Section 9.1) in a desktop environment to match the experimental framework descriptions with other established global routers to be compared. Tables IV and V show performance of FuzzRoute for ISPD'98 benchmarks. Comparisons are made with some fool-proof published academic global routers, namely Labyrinth [Kastner et al. 2002], FastRoute3.0 [Zhang et al. 2008], NTHU-R [Gao et al. 2008], BoxRouter 2.0 [Cho et al. 2009], and FGR [Roy and Markov 2008]. We cannot run our comparison against Dai et al. [2012] and Liu et al. [2013] because of the unavailability of the results with ISPD'98 benchmarks there.

First, the result shows that FuzzRoute is able to route through all the benchmarks.

Second, it achieves good runtime. It can finish routing all benchmarks within reasonable time on our platform. The improvements on routing time over Labyrinth, FastRoute3.0, NTHU-R, BoxRouter 2.0, and FGR are 91.81%, −10.29%, −34.91%, 86.87%, and 32.16%, respectively. Among all quoted global routers, FastRoute3.0 achieves fastest runtime. But it fails to generate congestion-free solutions for all benchmarks

Table IV. Comparison of CPU Time in Seconds between Published Global Routers and FuzzRoute on ISPD'98 Benchmark Circuits

Benchmark ISPD '98	Labyrinth [Kastner et al 2002]	FastRoute3.0 [Zhang et al. 2008]	NTHU-R [Gao et al. 2008]	BoxRouter 2.0 [Cho et al. 2009]	FGR [Roy and Markov 2008]	FuzzRoute
ibm01	21.2	0.64	4.17	33	10	3.0
ibm02	34.5	0.85	7.44	36	13	4.0
ibm07	228.1	1.68	15.89	86	18	22.0
ibm08	238.7	1.82	13.17	90	18	26.0
ibm09	505	1.67	11.59	273	20	28.0
ibm10	588	3.61	33.72	352	92	33.0
Total	1615.5	10.27	85.98	870	171	116.0
Norm	13.92	0.088	.74	7.5	1.474	1

Table V. Comparison of Wirelength between Published Global Routers and FuzzRoute on ISPD'98 Benchmark Circuits

Benchmark ISPD '98	Labyrinth [Kastner et al. 2002]	FastRoute3.0 [Zhang et al. 2008]	NTHU-R [Gao et al. 2008]	BoxRouter 2.0 [Cho et al. 2009]	FGR [Roy and Markov 2008]	FuzzRoute
ibm01	77K	64221	63321	62659	63332	46276
ibm02	205K	172223	170531	171110	168918	166922
ibm07	449K	369023	366288	365790	366180	357121
ibm08	470K	405935	405169	405634	404714	382855
ibm09	481K	414913	415464	413862	413053	403647
ibm10	680K	582838	580793	590141	578795	595465
Total	2362K	2010K	2001K	2009K	1993K	1952K
Norm	1.21	1.03	1.03	1.03	1.02	1

(e.g., ibm01 and ibm09), whereas FuzzRoute can achieve 100% congestion-free routability for all the circuits in much less time than other specified routers.

Third, in terms of total wirelength, FuzzRoute is not only comparable to others but much better, with improvements of 17.35%, 2.88%, 2.44%, 2.83%, and 2.10% over Labyrinth, FastRoute3.0, NTHU-R, BoxRouter 2.0, and FGR, respectively.

9.6. Comparison for ISPD'07 and '08 3D Benchmarks

Tables VI, VII, and VIII, show performance of FuzzRoute for ISPD'07 and '08 3D benchmark suites [Nam et al. 2007, 2008]. In Tables VI and VII, FuzzRoute identifies 2.5%, 2.6%, 1%, 1.1%, and 0.3% less wirelength and on average runs 1.68×, 6.42×, 2.21×, 0.76×, and 1.54× faster than NTHU-Route 2.0 [Chang et al. 2010], FastRoute 4.0 [Xu et al. 2009], NCTU-GR [Dai et al. 2012], MGR [Xu and Chu 2011], and NCTU-GR 2.0 [Liu et al. 2013], respectively. It is clear from the result set that FuzzRoute can perform the total routing operation much faster than the existing state-of-the-art tools. It result in a slower CPU time compared to MGR, but provides 1% wirelength improvement. Similarly, 1.3% larger wirelength results in much faster execution compared to FastRoute 4.0.

FuzzRoute achieves overflow-free solutions for maximum benchmarks of ISPD'07 and '08. Four of them, namely newblue3, newblue4, newblue7, and bigblue4, generate overflows, represented in Table VIII. FuzzRoute shows comparable results with 0.2%, −0.4%, 2.1%, 0.2%, and 0.3% improvement in the case of total overflow with NTHU-Route 2.0, FastRoute 4.0, NCTU-GR, MGR, and NCTU-GR 2.0, respectively.

Table VI. Comparison of CPU Time in Minute between Published Global Routers and FuzzRoute on ISPD'07/'08 3D Benchmark Circuits

Benchmark ISPD '07/'08	#Layer	NTHU-Route2.0 [Chang et al. 2010]	FastRoute4.0 [Xu et al. 2009]	NCTU-GR [Dai et al. 2012]	MGR [Xu and Chu 2011]	NCTU-GR2.0 [Liu et al. 2013]	FuzzRoute
adaptec1	6	4.86	3.31	3.90	4.93	2.30	1.73
adaptec2	6	1.42	0.95	1.45	1.04	0.64	0.32
adaptec3	6	6.16	3.69	4.88	4.83	2.96	2.12
adaptec4	6	2.08	1.25	2.28	1.41	1.18	3.72
adaptec5	6	11.95	6.70	9.07	7.95	4.97	4.73
bigblue1	6	6.93	4.22	6.35	5.04	3.44	2.97
bigblue2	6	6.47	12.12	11.18	6.00	3.45	1.95
bigblue3	8	3.91	2.06	4.38	2.89	1.78	3.31
bigblue4	8	52.63	93.25	65.37	21.31	63.55	56.29
newblue1	6	4.07	12.01	3.63	4.51	1.93	1.18
newblue2	6	1.17	0.85	0.90	0.80	0.63	0.74
newblue3	6	64.97	15.99	131.43	19.99	63.34	64.80
newblue4	6	52.01	65.23	40.92	15.64	17.48	3.02
newblue5	6	10.88	9.82	15.03	6.54	4.62	2.79
newblue6	6	10.34	8.78	9.67	7.04	4.02	3.82
newblue7	8	50.08	868.74	71.52	21.31	74.53	18.93
Total		289.93	1108.97	381.96	131.23	250.82	172.51
Norm		1.680	6.428	2.214	0.761	1.544	1

From an overall analysis of different dominating matrices of performance characterization for different standard benchmarks, it is clear that FuzzRoute is competitive with state-of-the-art tools, though it can provide an different kind of solution for NP-complete problems in a possibilistic way (implementing fuzzy logic), unlike using heuristics only.

9.7. Time Complexity Analysis

It is verified that the time complexity of generating guiding information is $O(l \times m \times n)$, where l the total number of layers, m the number of subregions in the x direction, and n the number of subregions in y direction of the layout. To perform global routing for each net, the time complexity would be $O(l \times (m + n))$. Hence, the total global routing procedure for connecting two-pin, multi-pin, and critical nets requires that much of the complexity be multiplied by the number of nets, and in addition with the time required to process a fuzzy expert system for each layer.

With respect to a two-pin intra-layer net, the proposed fuzzified approach will result in linear time complexity of $O(m + n)$. So, the proposed approach presents better time complexity than typical maze routing approaches, which is $O(m \times n)$ when $l = 1$. Hence, our proposed fuzzy-logic-based two-pin net routing will give linear time complexity instead of the quadratic time complexity of maze routing. Here, an optimal solution is achieved satisfying certain constraints without any guarantee of finding an optimum solution, whereas maze routing concentrates the solution quality on optimum routing path finding.

10. CONCLUSION AND FUTURE DIRECTION

This article highlights the paramount aspect of the global routing problem in 3D ICs by achieving a prominent degree of reliability and routability with a reasonable time complexity. The total methodology of the designed multi-objective global router has

Table VII. Comparison of Wirelength between Published Global Routers and FuzzRoute on ISPD'07/'08 3D Benchmark Circuits

Benchmark ISPD'07/'08	NTHU-Route2.0 [Chang et al. 2010]	FastRoute4.0 [Xu et al. 2009]	NCTU-GR [Dai et al. 2012]	MGR [Xu and Chu 2011]	NCTU-GR2.0 [Liu et al. 2013]	FuzzRoute
adaptec1	53.49	53.73	53.50	52.28	52.35	52.31
adaptec2	52.31	52.17	51.69	51.69	51.30	50.23
adaptec3	131.11	130.82	130.35	128.92	128.34	128.65
adaptec4	121.73	121.24	120.67	119.96	120.17	120.96
adaptec5	155.55	155.81	154.70	153.23	151.85	152.03
bigblue1	56.35	56.64	56.56	55.82	55.33	54.89
bigblue2	90.59	91.18	89.40	88.92	86.71	86.21
bigblue3	130.76	130.04	129.66	128.75	127.67	126.34
bigblue4	231.04	230.24	223.99	225.73	227.10	226.65
newblue1	46.53	46.33	45.99	45.58	45.62	45.37
newblue2	75.85	75.12	74.88	74.46	74.51	74.44
newblue3	106.49	108.40	104.28	107.22	106.8	105.36
newblue4	130.46	130.46	126.79	128.54	129.27	128.97
newblue5	231.73	230.94	230.31	228.00	225.94	226.78
newblue6	177.01	177.87	176.87	174.86	171.10	170.56
newblue7	353.35	353.38	338.63	349.02	341.90	340.48
Total	2144.35	2144.37	2108.27	2112.98	2095.96	2090.23
Norm	1.025	1.026	1.009	1.011	1.003	1

Table VIII. Comparison of Overflow between Published Global Routers and FuzzRoute on ISPD'07 / '08 3D Benchmark Circuits

Benchmark ISPD '07/'08	NTHU-Route2.0 [Chang et al. 2010]	FastRoute4.0 [Xu et al. 2009]	NCTU-GR [Dai et al. 2012]	MGR [Xu and Chu 2011]	NCTU-GR2.0 [Liu et al. 2013]	FuzzRoute
newblue3	31454	31276	31808	31026	31526	31426
newblue4	138	136	134	136	132	132
newblue7	62	54	114	56	54	56
bigblue4	162	130	164	134	132	134
Total	31816	31596	32220	31352	31844	31748
Norm	1.002	0.995	1.021	0.988	1.003	1

been verified successfully for two-pin, multi-pin, and critical intra- and inter-layer nets for 3D ICs. It may also be considered as a new type of guided global routing approach (using fuzzy logic) for standard cells. The procedure is tested on ISPD'98 and '08 benchmark suites and compared with some well-known global routers. Design of a fool-proof global routing solution for 3D ICs considering other metrics as well, further comparison with other state-of-the-art tools in the ISPD'11 benchmark suite, extension to mixed-size cell placements, and consideration of voltage drop impact as another working constraint may be considered as some possible future extensions of the present work.

ACKNOWLEDGMENTS

The authors are thankful to Dr. S. Das of IIEST Shibpur for his help in using a specific computing resource.

REFERENCES

J. Abonyi. 2003. Fuzzy model identification. In *Fuzzy Model Identification for Control*. Birkhauser, Boston, MA, 87–164.

C. J. Alpert. 1998. The ISPD98 circuit benchmark suite. In *Proceedings of the ACM International Symposium on Physical Design (ISPD'98)*. 80–85.

R. Babuska. 1998. *Fuzzy Modeling for Control*, 1st Ed. Kluwer Academic, Norwell, MA.

S. Bandyopadhyay. 2005. Automatic determination of the number of fuzzy clusters using simulated annealing with variable representation. In *Proceedings of the 15th International Conference on Foundations of Intelligent Systems (ISMIS'05)*. 549–602.

J. L. Castro and M. Delgado. 1996. Fuzzy systems with defuzzification are universal approximators. *IEEE Trans. Syst. Man Cybernet. B Cybernet.* 26, 1, 149–152.

C. J. Chang, P. J. Huang, T. C. Chen, and C. N. J. Liu. 2011. ILP-based inter-die routing for 3D ICs. In *Proceedings of the 16th Asia and South Pacific Design Automation Conference (ASPDAC'11)*. 330–335.

Y.-J. Chang, Y.-T. Lee, J.-R. Gao, P.-C. Wu, and T.-C. Wang. 2010. NTHU-route 2.0: A robust global router for modern designs. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 29, 12, 1931–1944.

M. Cho, K. Lu, K. Yuan, and D. Z. Pan. 2009. BoxRouter 2.0: A hybrid and robust global router with layer assignment for routability. *ACM Trans. Des. Autom. Electron. Syst.* 14, 2, 32:1–32:21.

K.-R. Dai, W.-H. Liu, and Y.-L. Li. 2012. NCTU-GR: Efficient simulated evolution-based rerouting and congestion-relaxed layer assignment on 3D global routing. *IEEE Trans. VLSI Syst.* 20, 3, 459–472.

S. Das, A. Chandhakasan, and R. Reif. 2003. Design tools for 3D integrated circuits. In *Proceedings of the Asia and South Pacific Design Automation Conference the (ASPDAC'03)*. 53–56.

J.-R. Gao, P.-C. Wu, and T.-C. Wang. 2008. A new global router for modern designs. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASPDAC'08)*. 232–237.

P. Ghosal, H. Rahaman, and P. Dasgupta. 2009. 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'09)*. 69–78.

P. Ghosal, H. Rahaman, and P. Dasgupta. 2010a. An experimental study of interconnection length in 3D and 2D VLSI. In *Proceedings of the IEEE Latin American Symposium on Circuits and Systems (LASCAS'10)*.

P. Ghosal, H. Rahaman, and P. Dasgupta. 2010b. Cell level thermal placement in 3D ICs. In *Proceedings of the Annual IEEE India Conference (INDICON'10)*. 1–4.

P. Ghosal, H. Rahaman, and P. Dasgupta. 2010c. Minimizing thermal disparities during placement in 3D ICs. In *Proceedings of the 13th IEEE International Conference on Computational Science and Engineering (CSE'10)*. 160–167.

P. Ghosal, H. Rahaman, and P. Dasgupta. 2010d. Thermal aware placement in 3D ICs. In *Proceedings of 2nd International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom'10)*. 66–70.

P. Ghosal, T. Samanta, H. Rahaman, and P. Dasgupta. 2008. Thermal-aware placement of standard cells and gate arrays: Studies and observations. In *Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'08)*. 369–374.

B. Goplen and S. S. Sapatnekar. 2005. Thermal via placement in 3D ICs. In *Proceedings of the International Symposium on Physical Design (ISPD'05)*. 167–174.

A. Gupta, N. D. Dutt, F. J. Kurdahi, K. S. Khouri, and M. S. Abadir. 2008. Thermal aware global routing of VLSI chips for enhanced reliability. In *Proceedings of the 9th International Symposium on Quality Electronic Design (ISQED'08)*. 470–475.

C. J. Huang, Y. W. Wang, H. M. Chen, H. W. Tsai, J. J. Jian, . L. Cheng, and J. J. Liao. 2014. Application of cellular automata and type-2 fuzzy logic to dynamic vehicle path planning. *Appl. Soft Comput.* 19, 333–342.

S. Im and K. Banerjee. 2000. Full chip thermal analysis of planar (2-D) and vertically integrated (3D) high performance ICs. In *International Electronic Devices Meeting Technical Digest (IEDM'00)*. 727–730.

M. Kanthi, H. S. Mruthyunjaya, and V. I. George. 2013. Fuzzy logic control for active ankle foot orthosis. In *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ'13)*. 1–6.

R. Kastner, E. Bozorgzadeh, and M. Sarrafzadeh. 2002. Pattern routing: Use and theory for increasing predictability and avoiding coupling. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 21, 7, 777–790.

W.-H. Liu, W.-C. Kao, Y.-L. Li, and K.-Y. Chao. 2013. NCTU-GR 2.0: Multithreaded collision-aware global routing with bounded-length maze routing. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 32, 5, 709–722.

K. Lu and D. Z. Pan. 2009. Reliability-aware global routing under thermal considerations. In *Proceedings of the 1st Asia Symposium on Quality Electronic Design (ASQED'09)*. 313–318.

E. Lughofer. 2011. *Evolving Fuzzy Systems - Methodologies, Advanced Concepts and Applications*. Springer.

J. Minz, E. Wong, and S. K. Lim. 2005. Thermal and crosstalk-aware physical design for 3D system-on-package. In *Proceedings of the Electronic Components and Technology Conference (ECTC'05)*. 824–831.

M. D. Moffitt. 2009. Global routing revisited. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD'09)*. 805–808.

G.-J. Nam, C. Sze, and M. Yildiz. 2008. The ISPD global routing benchmark suite. In *Proceedings of the International Symposium on Physical Design (ISPD'08)*. 156–159.

G.-J. Nam, M. Yildiz, D. Z. Pan, and P. H. Madden. 2007. ISPD placement contest updates and ISPD 2007 global routing contest. In *Proceedings of the International Symposium on Physical Design (ISPD'07)*. 167.

M. Pathak and S. K. Lim. 2009. Performance and thermal-aware Steiner routing for 3-D stacked ICs. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 28, 9, 1373–1386.

W. Pedryz and F. Gomide. 2007. *Fuzzy Systems Engineering: Toward Human-Centric Computing*. John Wiley and Sons, Hoboken, NJ.

W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. 2001. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, London.

D. Roy and P. Ghosal. 2013. A fuzzified approach towards global routing in VLSI layout design. In *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ'13)*. 1–8.

D. Roy, P. Ghosal, and N. Das. 2014a. A thermal and congestion driven global router for 3D integrated circuits. In *Proceedings of the Students' Technology Symposium (TechSym'14)*. 303–308.

D. Roy, P. Ghosal, and S. P. Mohanty. 2014b. FuzzRoute: A method for thermally efficient congestion free global routing in 3D ICs. In *Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'14)*. 71–76.

J. A. Roy and I. L. Markov. 2008. High-performance routing at the nanometer scale. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 27, 6, 1066–1077.

K. Sabahi, S. Ghaemi, and S. Pezeshki. 2014. Application of type-2 fuzzy logic system for load frequency control using feedback error learning approaches. *Appl. Soft Comput.* 21, 1–11.

S. M. Sait and H. Y. H. Ali. 1999. Fuzzy simulated evolution algorithm for multi-objective optimization of VLSI placement. In *Proceedings of the Congress on Evolutionary Computation (CEC'99)*.

C.-H. Tsai and S.-M. Kang. 2000. Cell-level placement for improving substrate thermal distribution. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 19, 2, 253–266.

N. Viswanathan, C. J. Alpert, C. Sze, Z. Li, G.-J. Nam, and J. A. Roy. 2011. The ISPD-2011 routability-driven placement contest and benchmark suite. In *Proceedings of the ACM International Symposium on Physical Design (ISPD'11)*. 141–146.

L. X. Wang. 1992. Fuzzy systems are universal approximators. In *Proceedings of the IEEE International Conference of Fuzzy Systems (FUZZ'92)*. 1163–1170.

T. H. Wu, A. D. Jeffrey, and T. Linderoth. 2009. GRIP: Scalable 3D global routing using integer programming. In *Proceedings of the 46^{th} ACM/IEEE Design Automation Conference (DAC'09)*. 320–325.

Y. Xu and C. Chu. 2011. MGR: Multi-level global router. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD'11)*. 250–255.

Y. Xu, Y. Zhang, and C. Chu. 2009. FastRoute 4.0: Global router with efficient via minimization. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASPDAC'09)*. 576–581.

L. A. Zadeh. 1973. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Syst. Man Cybernet.* SMC-3, 1, 28–44.

T. Zhang, Y. Zhan, and S. S. Sapatnekar. 2006. Temperature-aware routing in 3D ICs. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASPDAC'06)*. 309–314.

T. Zhang, Y. Zhan, S. S. Sapatnekar, J. Cong, and Y. Zhang. 2005. Thermal-driven multilevel routing for 3-D ICs. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASPDAC'05)*. 121–126.

Y. Zhang, Y. Xu, and C. Chu. 2008. FastRoute3.0: A fast and high quality global router based on virtual capacity. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD'08)*. 344–349.

Received November 2014; revised April 2015; accepted April 2015