

Difficult Channel Instance Generator for VLSI Physical Design Automation using Genetic Algorithm

Swagata Saha Sau^{1*} and Rajat Kumar Pal²

¹Department of Computer Science, Sammilani Mahavidyalaya, Baghajatin, Kolkata – 700075, West Bengal, India
swagatasahasau@gmail.com

²Department of Computer Science and Engineering, University of Calcutta, Kolkata –700009, West Bengal, India

Abstract

Objectives: The wire length minimization of Channel routing problem is NP-hard. There are several heuristic algorithms available in the literature to get the feasible routing solutions using some limited instances. Here we want to generate all possible random channel instances based on Genetic Algorithm (GA). **Methods/ Statistical Analysis:** The present paper is described in three phases. In the first phase, we generate fixed size initial population based on some strategies and define the fitness value of each individual by the total number of vertical and horizontal constraints present in the channel. The best individual is obtained among the population based on height fitness value. In the second phase, we select two individuals based on Roulette Wheel Selection strategies and get two offspring's using single point crossover. We continue this process until the number of offspring's reached to the size of the population. To keep the population size fixed we apply reduction methods among the initial population along with all offspring are based on minimum fitness value. In the third phase, we randomly choose some channels for mutation and we find the best individuals among current population. This methodology will be continuing until and unless we reach the goal. **Findings:** The proposed method works efficiently for complex problems arise in VLSI physical design automation and it gives acceptable results in terms of different channel instances. **Applications/Improvement:** The newly designed method for different difficult channel instances generation techniques will help to judge any newly developed algorithm in the field of VLSI physical design automation.

Keywords: Channel Instance, Channel Routing Problem, Genetic Algorithm

1. Introduction

The Channel Routing Problem (CRP) of the area and wire length minimization is recognized a problem in VLSI physical design automation. We know that minimization of area is vital for cost optimization point of view as well as for reducing impurities present on a wafer. The wire length minimization problem in channel routing is also a very significant practical problem as it also reduces routing cost, but the problem is more imperative as a high-performance factor of computing a routing solution.¹⁻³ More wire length means more delay and also more congestion of wire segments that may produce more heat and cause for signal interference.

Channel routing problem for area minimization is proved as NP-complete and that for wire length

minimization is proved as NP-hard.^{4,5} Hence several heuristic algorithms have been developed to get the feasible routing solutions of channel routing problem.^{4,6-12} A very few channel instances are available to judge these heuristic algorithms. Deutsch's Channel Example (DDE) was used extensively as benchmark for testing new algorithms, but this channel instance is not sufficient in recent research domain.⁴ In real life, channels are complex where some heuristic algorithms may not satisfy. Novelty of a heuristic algorithm is judged better of randomly generated instances of the problem.

1.1 Basic Definitions

Generally, for a chip design, a channel is a rectangular routing region that is formed when two blocks are placed

*Author for correspondence

sideways on a chip floor and a routing region is sandwiched in between. Terminals to be connected inside a channel are placed on the periphery of the blocks.^{4,5,7-11} Hence, a channel is a rectangular routing region that has two sets of fixed terminals located horizontally on two reverse sides of the channel and the other two sides are two open ends that may or may not contain any terminal of net and those are not fixed. A set of terminals that need to be electrically joined by wires is called a *net* and all these terminals are assigned a single number. If a net contains more than two terminals, is called a *multi-terminal net*. Vacant terminals are assigned the number zero that not to be connected. Usually, a channel is represented by *channel specification* (or *net list*) that contains two vectors of equal length of net numbers (including vacant terminals).

For the reserved multi-layer V_1H_1 no-dogleg Manhattan channel routing model, alternating layers (vertical (V) and horizontal (H)) are used for vertical and horizontal wire connection of net, respectively^{4,5}. In Manhattan routing, the model allows only rectilinear wiring for all necessary interconnections. In no-dogleg routing, the single horizontal wire is assigned to track for all individual net. This routing model is most practical as it is the simplest and modular routing model adopted in designing a maximum of marketable chips, whose performance is also within the limit of tolerance.

Now we formally define the two constraints present in general CRP. These constraints are termed as horizontal constraints and vertical constraints.^{4,5,8-11} Two nets are said to be horizontally constrained, if their intervals overlap when they are assigned to the same track. As per Figure 1, we may observe that the interval I_4 overlaps with I_2 , I_3 and I_5 but does not overlap with I_1 . Hence we may conclude that I_1 and I_4 may be assigned to the same track but not I_4 with either of I_2 , I_3 or I_5 .

Vertical constraints in the channel determine the order of placement of nets from top to bottom or vice-versa in the channel.^{4,5,8-11} Now in Figure 1 the first column

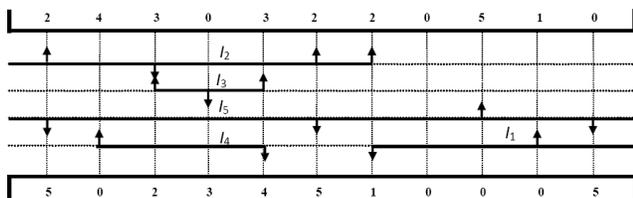


Figure 1. An example of a channel instance that contains five nets.

of the channel, we may observe that the interval I_2 has to be assigned to a track above the track where I_5 is assigned otherwise it makes a short circuit. If the channel has no vertical constraints that mean one side of each column of a channel has non-terminal. In this case, any net come to assigned to any track.

Existing benchmarks in literature represent an extremely small subset of real problems, they may not represent the complexity that exists in the majority of today's and the future's designs.⁴ It is possible to design an algorithm that works well for known benchmarks, but not for other examples. As the number of transistors on a chip has increased considerably and testing on the traditional benchmarks may not be sufficient for evaluating the performance of a newly devised channel routing algorithm, so it is an important issue to generate (difficult) random channel instances.

In 1995, Chao and Harper proposed a random channel routing generator which can generate difficult channel routing instances of any arbitrary size and that can be routed without dogleg.¹³ Their algorithm is known as the *CRP_generator*.

To be a valid channel specification that can be routed without doglegs, the VCG must be a-cyclic. To achieve this authors randomly generate a -cyclic VCG for the channel instances. Thus, it has the advantage of using a minimum number of via. The authors generate the channel instances which has both horizontal and vertical constraints. As their proposed algorithm can generate instances of arbitrary size, hence they claim that the channel becomes intractable as n increases. Instead of some advantages, there are some restrictions and or demerits of their proposed algorithm some which are as follows.

- Their channel routing generator generates the channel instances in such a way that the vertical constraint graphs of those are acyclic which is not matched with real life situations.
- The authors generate CRPs in which each net only have terminals either on the top or bottom side, but in real life problem, the channels may have some trivial nets.
- Floating terminals are not considered in their channel instances.
- The authors consider the nets which have either no vertical constraints or have only a single vertical constraint as the channel has only two-terminal nets, but, these types of restriction in instances may not reach the channel instances in real life problem.

- The vertical constraint between net n_i and net n_j occurs in one column in their channel instances, but it may occur in multiple columns in the channel.
- Channel Instances are not fully randomly generated.

In 2010, Banerjee, Dey, and Dutta proposed a difficult channel routing generator which generates cycle-free difficult channel instances using the Genetic Algorithm (GA).¹⁴ As the channel instances are cycle free with respect to Vertical Constraint Graph (VCG) of the channels, and then these instances can be solved without dogleg. The generated channel instances have both horizontal and vertical constraints. The authors claim that their random channels are difficult to route as those channels have higher vertical and horizontal constraints. Deutsch's Channel Example (DDE) was used extensively as a benchmark for testing new algorithms, but the authors showed that their channel instances containing same channel length and a same number of nets like DDE are more difficult than DDE (though DDE has one floating terminal which is not considered in their instances). They define the term "difficult" as the maximum constraints (both vertical and horizontal) present in the channel and they fix it as the fitness value of the channel. Instead of some advantages, there are some restrictions or demerits of their proposed algorithm which are as follows.

- Their channel routing generator generates the channel instances in such a way that the vertical constraint graphs of those channels are acyclic which is not matched with real life problem.
- The authors generate CRPs in which each net only have terminals from either the top or bottom side, but in real situation, the channels may have some trivial nets.
- Floating terminals are not considered in their channel instances.
- The authors consider the nets which have either no vertical constraints or have only a single vertical constraint as two-terminal nets whereas these types of restriction in instances may not reach the channel instances in real life problem.
- The vertical constraint between net n_i and net n_j occurs in one column in their channel instances, but it may occur in multiple columns in the channel to be designed.
- The complexities of the algorithm are not discussed in their paper.

- Parent's selection method of the channels for cross-over process and Reduction (P_c) are not discussed in this paper.

In 2010, develop two algorithms for generating random channel instances for channel routing problem in VLSI physical design.¹⁵ Channel instances are mainly of two types: simple channel instances where channel contains only horizontal constraint and other is general channel instances, where channel contains both horizontal and vertical constraints. The authors developed two algorithms Simple_Random_Channel_Generator and General_Random_Channel_Generator which generate simple and general channel instances, respectively. Here we discuss those algorithms in brief.

The authors have developed an algorithm Simple_Random_Channel_Generator which randomly generates the simple channel instances of n two-terminal nets. The length of the channel is fixed and it is equal to $2n$. Initially all the column positions of the channel is free. The authors developed their algorithm in such a way that the initial and the final column positions of any net are chosen randomly and no column will be selected more than once. For selection of the final column of any net of the channel, the authors use the variable *max_offset* which randomly fix the span of the net. The terminals of the nets are randomly placed on the top and at the bottom of the channel. The algorithm takes $O(n^2)$ time, and $O(n)$ space where n is the number of nets introduced in the channel.

The authors developed another algorithm General_Random_Channel_Generator which randomly generates the general channel instances consisting of n nets, each net is assumed to be k terminals ($2 \leq k \leq 6$) which is randomly generated. If k_i is the number of terminals of net i , then the total number of terminals for n nets is K and $K = k_1 + k_2 + \dots + k_n$. If NT is the number of non-terminals (randomly generated) in the channel, then the total number of terminals is $P = K + NT$ and these are evenly distributed over top and bottom position of the channel specification along the length of the channel in random fashion. The estimated length of the channel is $L = \epsilon P \alpha 2 \mu$. In order to place P terminals properly, they have used three auxiliary lists *NET_LIST*, *TERM_COUNT*, and *TERM_LIST*. *NET_LIST* is the list which is initially filled in with positive integers from 1 to n . *TERM_COUNT* is the list of length n and it stores the number of terminals for the nets. *TERM_LIST* is a list of length $2L$ and initially it stores the positive integer from 1 up to $2L$.

Initially all the top and the bottom terminal positions of the channel are free i.e., all are zeros. In each iteration, the algorithm select a net number (i) randomly and locate its terminals in different columns (c) in C , and again randomly determine the terminal position, either on the top or at the bottom c for net i . At first, the algorithm selects an element p from NET_LIST and removes the element from the list immediately. Now an element k from $TERM_COUNT$ is selected randomly and the entry is deleted from the list. For each of these k terminals, the algorithm generates a random number r and selects the r -th element from $TERM_LIST$. Let the number is s , then assign the terminal of net p at column $c = \text{és} \square 2 \dot{u}$ in random fashion either on the top or at the bottom. The algorithm takes $O(n^2)$ time, and $O(n)$ space where n is the number of nets introduced in the channel. The authors also comment on the removal of cyclic vertical constraints. Instead of some advantages, there are some restrictions or demerits of their proposed algorithms which are as follows.

- In case of generating simple channel instances they imposed the restriction on the nets. They consider only two-terminal nets. But in real problem simple channel instances may have multi-terminal nets.
- Floating terminals are not considered in general channel instances.
- Though the authors have given experimental results considering with up to 15000 numbers of nets, but none of the channel instances is given in their paper as the benchmark for judgement of new heuristic methods in channel routing problems.

Several interdisciplinary research works have been carried out by researchers in the area of VLSI circuit synthesis.¹⁶⁻¹⁹

The remaining paper is organized as follows: We formulate the Problem developed in this paper in Section 2. Section 3 develops the algorithm which generates difficult random channel instance. Some experimental results have been included in Section 4. The paper is concluded in Section 5 with few remarks.

2. Formulation of the Problem

In this paper we generate random difficult channel instances using Genetic Algorithm (GA). A general channel instance is difficult when it has more constraints, i.e., it has maximum number of horizontal as well as vertical constraints. Genetic algorithm is an optimization

technique guided by the principle of natural genetic system and it performs well when the problem can be modelled as optimization one.^{20,21} Another advantage of using GA is that it always gives us results better with time.

Our proposed algorithm generates the random difficult channel instances where the some characteristics are imposed randomly in the channel.

- The channel has both the constraints, i.e., horizontal and vertical constraints and the channel is difficult or complex when a total number of constraints are maximized.
- The nets are two- and multi-terminal nets.
- The channel may have floating terminals along with fixed terminals. The numbers of floating terminal are randomly considered and nets which have floating terminals also choose in random.
- The channel may have some trivial nets.
- The channel may or may not have cyclic vertical constraints. The removal of cyclic vertical constraints is also given in this paper.
- As the difficult channel instances are randomly generated, hence these are very useful to judge the new heuristic algorithms of channel routing problem. As the channel routing generator developed in this paper can randomly generate difficult instances of arbitrary size, it will fully test these algorithms.
- According to Holland's schema theorem, we know that a schema with an above average fitness tends to increase at an exponential rate until it becomes a significant portion of the population. So, we fixed up the fitness function of a channel to be the total number of vertical and horizontal constraints in the channel. We define the fitness function of a channel as above because we can get a difficult channel with maximum fitness value.

We shall analysis the complexity of the proposed algorithm and show that it is polynomial time computable. We create initial population P_c of fixed size (number of individuals = 100 (may be changed as per requirement)) using the algorithm `General_Random_Channel_Generator` but we imposed another parameter on the instances, i.e., consideration of floating terminals (LCS and/or RCS) to increase the horizontal constraints in the channel.^{14,15} The numbers of floating terminal are randomly considered and nets which have floating terminals also choose in random. We apply genetic algorithm to generate new random difficult channel instance. We fixed up the fitness

function (use in GA) of a channel to be the total number of vertical and horizontal constraints in the channel. We define the fitness function of a channel as above because we can get difficult channel with maximum fitness value. We terminate the process when we reach our goal.

3. Algorithm to Generate Difficult Random Channel Instance

The steps of the proposed algorithm are given as follows:

Algorithm Difficult_Random_Channel_Generator

Step 1 [Create initial population (P_c):

We create initial population P_c of fixed size (number of individuals = 100) using the algorithm General_Random_Channel_Generator¹² but we imposed another parameter on the instances, i.e., consideration of floating terminals (LCS and/or RCS) to increase the horizontal constraints in the channel. The numbers of floating terminal are randomly considered and nets which have floating terminals also choose in random.

Step 2:[Fitness Calculation]

We fixed up the fitness function of a channel to be the total number of vertical and horizontal constraints in the channel. We define the fitness function of a channel as above because we can get difficult channel with maximum fitness value.

Step 3 [Best individual of P_c]:

Find the best individual of P_c i.e., the channel whose fitness value is maximum (if maximum fitness values occur for more than one channel, then consider one which has more vertical constraints using the function *Best_individual*.)

$$P_{best} = \text{Best_individual}(P_c)$$

Step 4:

For generation = 1 to *max_generation*

Step 4.1: $P_n = \varphi$

Step 4.2 [Selection of parents for crossover]:

For *offspring* = 1 to *max_decendent*

Step 4.2.1:

Apply Roulette Wheel selection method twice on a population to select two different parents for crossover. According to Darwin's evaluation theory of survival of the fittest, the best ones should survive and create new

offspring. Let P_α and P_β are two parents selected using Roulette Wheel selection method.

Step 4.2.2 [Crossover]:

Let p be the probability of cross over, $0 \leq p \leq 1$, usually, p is between 0.6 and 0.9. So we expect that on an average 60% to 90% of the chromosomes will undergo crossover. A random number r is generated in the interval $[0,1]$. If $r \leq p$ then crossover is performed on P_α and P_β , else two offspring's Y_1 and Y_2 are identical to their parents P_α and P_β . Apply single site crossover between P_α and P_β , for that a cross-site is selected randomly. To find cross site or crossing position, we first find the C_{min} , which is the minimum channel length among P_α and P_β . Now generate a random number k between 1 to C_{min} to find the cross site. Now two offsprings Y_1 and Y_2 are produced. Y_1 is produced by taking column 1 to column k from P_α and $k + 1$ to last column of P_β . Another offspring Y_2 is produced by taking column 1 to column from P_β and column $k + 1$ to last column from P_α .

$$\text{Step 4.2.3: } P_n = P_n \cup \{ Y_1, Y_2 \}$$

Step 4.3: Calculate fitness value of P_n using the fitness function *Fitness_calculation*(P_n)

Step 4.4 [Reduction]:

Find the new population P_c of the same size (100) using the reduction method. $P_c = \text{Reduction}(P_c \cup P_n)$

Reduction function generates new P_c after eliminating channels from $P_c \cup P_n$ according to their minimum fitness value.

Step 4.5 [Mutation]:

Mutation is obtain using the function *Mutation* on P_c , which first randomly choose some channel from P_c for mutation and then mutation any channel (selected for mutation) is performed. For mutation of any channel it selects any two random numbers between 1 and length of the channel and swaps these two columns which may increase horizontal constraints.

Step 4.6 [Fitness Calculation]:

Calculate the fitness value of all the channels of P_c using fitness function.

Step 4.7: Find the best individual of P_c using the function

$$\text{Best_individual on } P_c \cup P_{best} \\ \text{So } P_{best} = \text{Best_individual}(P_c \cup P_{best})$$

Step 5: Exit

Now we shall discuss the removal of cyclic vertical constraints present in the channel to be designed. Some

model allows cyclic vertical constraints but we can obtain feasible routing solution in two-layer VH and three-layer HVH no-dogleg Manhattan model provided VCG is cycle free. There are different algorithms in literature those are helpful to detect the cycle of VCG. But depth first search (DFS) is suitable for this as it can find back edge in $O(n + e)$ times, where n is the number of nets in the generated channel instance and e is the number of edges in the VCG. If we remove the back edge, say (v_i, v_j) from VCG then VCG is cycle free.

In the instance their corresponding net n_i and n_j in one column split into two different columns as n_i on the top and 0 is at the bottom of one column and 0 is on the top and n_j is at the bottom of another column. We need one additional column to remove one cycle in VCG. Hence channel length is estimated as the sum of channel length and number of back edges in the VCG.

Here, we shall discuss about the time complexity of the proposed algorithm. Step 1 of algorithm *Difficult_Random_Channel_Generator* takes $O(n^2)$ time to create initial population P_c . Step 2 takes constant time to calculate the fitness function of channels in P_c as the population size is fixed. Step 3 takes constant time to find the best individual. Let $max_generation$ be t . Hence step 4 execute t times. Step 4.2.1, 4.2.2, and 4.2.3 takes constant time as the population size is fixed. Hence step 4 takes $O(t)$ times. So the algorithm *Difficult_Random_Channel_Generator* takes $O(n^2 + t)$ times where n is the number of nets and t is the $max_generation$ or the number of iteration we require to generate a difficult random general channel instance.

4. Experimental Results

The proposed methods for generating difficult random channel instances are tested for some problems in VLSI physical design automation. It gives acceptable results. Figure 2, Top represents the top channel specification and Bottom represents bottom channel specification. Left Connection Set (LCS) and Right Connection Set (RCS) represents the set of floating terminals which enter into the channel either from the left or from the right or from both ends. The fixed size initial population is generated based on algorithm¹⁵ along with some more constraints. Here we consider the size of the initial population as 100. The objective function can be considered as the fitness function as it is a maximization problem. The fitness function of each chromosome or solution is defined as a

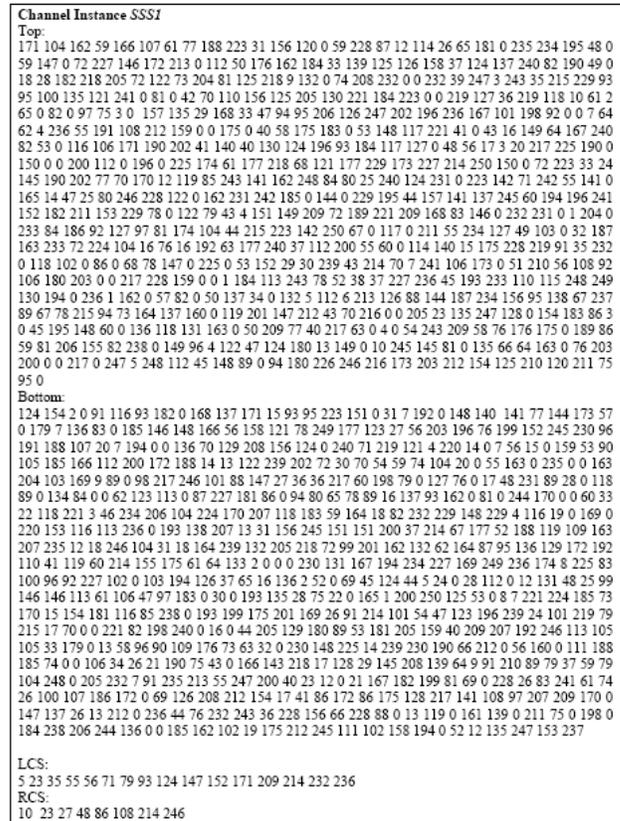


Figure 2. Random difficult channel instance SSS1 generated using algorithm *Difficult_Random_Channel_Generator*.

total number of horizontal and vertical constraints present in the channel. We select two individuals for single point crossover based on Roulette Wheel Selection strategies from the initial population. To perform the Roulette Wheel Selection we calculate the cumulative fitness function of each chromosome. Let the cumulative fitness function of the last chromosome be R . Then we generate a random number in $(0, R)$. For a particular performance of the random experiment (to generate the random number), r be the outcome. Then we select the corresponding chromosome whose cumulative fitness value is less than or equal to r . After selecting two chromosomes, the single point crossover is performed the mating pool. The experimental result gives the remarkable results in terms of different difficult random channel instances.

5. Conclusion

The Channel routing problem for area minimization is a NP-complete problem and at the same the time wire

length minimization problem of it is an NP hard problem. Several heuristic algorithms have been proposed by different researchers to get the feasible routing solutions of the channel routing problem. Some of the existing heuristic methods developed by different researchers are well appreciated; however, a very few channel instances are available to judge these heuristic algorithms. So, it is not sufficient to meet the current requirement. Under such circumstances, difficult channel instance generation technique has come up as a most effective solution. Hence, we are able to suggest a new technique for different channel instant generation using Genetic Algorithm (GA). The conclusions of the proposed method on the basis of our findings along with the existing facts are summarized as follows:

- Different difficult channel instances are generated by the proposed technique for VLSI physical design automation problem.
- The proposed technique is easy to generate the different random difficult channel instances.
- The newly generated instance also satisfies the Holland's observation.
- Here, we considered the single point crossover, and two-point mutation, but we observed that other methods of crossover, as mutation, also give acceptable results.
- Here, we considered the Roulette Wheel Selection procedure, but we observed that other methods like Tournament Selection or Steady State Selection method give remarkable results.
- Random difficult channel instances generated using the proposed algorithm help us to judge the newly developed algorithms in the field of VLSI physical design automation.

Hence, the newly suggested method for different difficult channel instant generation technique is highly encouraged.

6. References

1. Kavya G, Thulasibai V. VLSI Implementation of Telemonitoring System for High Risk Cardiac Patients. *Indian Journal of Science and Technology*. 2014 Jan; 7(5):1–6.
2. Malathy K, Justus RB. A Novel VLSI Based Radix-2 Single Path Delay Commutator (R2SDC) FFT Architecture Design. *Indian Journal of Science and Technology*. 2016 Mar; 9(11):1–9. Crossref
3. Manikandan J, Manikandan M. VLSI based Pipelined Architecture for Radix-8 Combined SDF-SDC FFT. *Indian Journal of Science and Technology*. 2016 Jul; 9(28):1–5. Crossref
4. Pal RK. *Multi-Layer Channel Routing: Complexity and Algorithms*, Narosa Publishing House, New Delhi (Also published from CRC Press, Boca Raton, USA and Alpha Science International Ltd). UK; 2000.
5. Alpert CJ, Mehta DP, Sapatnekar SS. *Handbook of Algorithms for Physical Design Automation*. London, New York. CRC Press; 2009.
6. Hong C, Kim Y. The Efficient Hybrid Approach to Channel Routing Problem. *International Journal of Advanced Science and Technology*. 2012;42:61–8.
7. Wagner MDF, Wagner F. Routing Through a Dense Channel with Minimum Total Wire Length. *Proceeding of Second Annual ACM-SIAM Symposium*. 1991; 15(2):475–82.
8. Sau SS, Pal RK. An Efficient High Performance Parallel Algorithm to Yield Reduced Wire Length VLSI Circuits. *Proceeding of 5th International Conference on Computers and Devices for Communication (CODEC)*. Kolkata; 2012. Crossref
9. Sau SS, Pal RK. A Re-router for Optimizing Wire Length in Two and Four-Layer No-Dogleg Channel Routing. *Proceedings of 18th International Symposium on VLSI Design and Test. VDAT*. Publisher: IEEE 978-1-4799-4006-6/14/2014 IEEE. 2014. p. 1–6. Crossref
10. Sau SS, Pal RK. An Efficient Algorithm for Reducing Wire Length in Three-Layer Channel Routing. *Applied Computation and Security Systems. Series Advances in Intelligent Systems and Computing*. Springer India. 2011; 2:145–56.
11. Sau SS, Pal RK. A Re-router for Reducing Wire Length in Multi-Layer No-Dogleg Channel Routing. *International Journal of Computer Trends and Technology (IJCTT)*. 2016; 38(2):1–17.
12. Somogyi KA, Recski A. On the Complexity of the Channel Routing Problem in the Dogleg-free Multi-layer Manhattan Model. *ACTA Polytechnica Hungarica*. 2004; 1(2):1–9.
13. Chao HY, Harper MP. A Difficult Channel Routing Generator. *ECE Technical Reports Paper*. 1995. p. 1–16.
14. Banerjee S, Dey MT, Dutta S. Difficult Channel Generation Using Genetic Algorithm. *International Journal of Artificial Intelligence and Applications (IJAIA)*. 2010; 1(4):145–57. Crossref
15. Pal A, Mandal TN, Pal RK, Kundu D, Datta AK. Algorithms for Generating Random Channel Instances for Channel Routing Problem. *International Journal of Applied Research on Information Technology and Computing*. 2010; 1(1):106–29. Crossref
16. Bremiga GG, Malleswari M, Enoch S. An Improved VLSI Algorithm for Modular Operation in Cryptography. *Indian*

- Journal of Science and Technology. 2016 Aug; 9(30):1-7. Crossref
17. Ramalakshmi P, Saravanan S. Multiple Scan Base Partitioning Technique to Increase the Throughput in VLSI Testing. Indian Journal of Science and Technology. 2016 Aug; 9(29):1-5. Crossref
 18. Nivedha N, Muthaiah R. A VLSI Architecture of Root Raised Cosine Filter using Efficient Algorithm. Indian Journal of Science and Technology. 2016 Aug; 9(29):1-5. Crossref
 19. Premson Y, Sakthivel R. VLSI Architecture for Image Contrast Enhancement using Modified Adaptive Gamma Correction with Weighting Distribution. Indian Journal of Science and Technology. 2016 Feb; 9(5):1-7. Crossref
 20. David E, Goldberg G. Genetic Algorithms. Pearson Education India. Fourth Impression. ISBN: 978-81-7758-829-3. 2009.
 21. Affenzeller M, Wagner S, Winkler S, Beham A. Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications (Numerical Insights). Chapman & Hall/CRC; 2009. Crossref