Difficult Channel Generation Using Genetic Algorithm

Siddhartha Banerjee

Department of Computer Scince, Rama Krishna Mission Residential College, Narendrapur

Mitra Tithi Dey and Sayari Dutta

Department of Computer Science, Barrackpore Rastraguru Surendranath College, Barrackpore

Abstract

For channel routing problems, Deutsch's channel examples were used extensively as benchmarks for testing new algorithms. However, it is also important to test the performance of channel routing algorithms on a wider variety of difficult examples. In this paper, we present a random channel routing generator which can generate difficult channel routing instances of arbitrary size. In this paper our goal is to generate those channels using Genetic Algorithm, which have maximum constraints (both vertical and horizontal constraint).

Key words: Channel routing, vertical and horizontal constraint, Genetic algorithm.

1. Introduction

Channel routing plays a central role in the physical design of VLSI chips. To meet the increasing demands of functionality, the number of transistors on a chip today has increased considerably. Most layout systems begin with placing modules on a chip, and then wiring together terminals that should be electrically connected on different modules. An efficient approach for solving the wiring problem is to heuristically partition the chip into a set of rectangular channels, and then route each channel separately. This effectively divides a difficult problem into smaller sub-problems that can be conquered more easily.

For channel routing problems, Deutsch's examples [2] were used extensively as benchmarks for evaluating the performance of new algorithms, especially the so-called Deutsch's Difficult Example (DDE). In this paper, we develop a random channel routing generator which can generate difficult channel routing instances of arbitrary size using Genetic Algorithm. This research is motivated by the following facts:

- Because the benchmarks 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.
- The number of transistors on a chip has increased considerably. Testing on the traditional benchmarks may not be sufficient for evaluating the performance of channel routing algorithms.

DOI : 10.5121/ijaia.2010.1412

- It is possible to design an algorithm that works well for known benchmarks, but not other examples. As our channel routing generator can randomly generate difficult instances of arbitrary size, it will fully test these algorithms.

In Section 2, we introduce the model as well as the major constraints on a CRP. Then in section 3 we introduce the concept of Genetic algorithm. The difficult CRP generator using Genetic Algorithm is discussed in Section 4 and some random examples is given in Section 5.

2. The Restricted Manhattan Model

We consider the two-layer *restricted* Manhattan model [5]. Although the three-layer and multi-layer process is available, the two-layer model is still attractive for the following reasons:

- The yield is higher for the two-layer process. The three-layer and multi-layer process is much more expensive than the two-layer process.
- If the product is time critical on the market, the two-layer model provides a faster way of bringing the product to the market.

A two-layer *channel* is a gridded rectangular area on a chip consisting of a metal layer running *horizontally* and a polysilicon layer running *vertically* (or vice versa). A wire in the horizontal layer is called a *track* and a wire in the vertical layer is called a *column.* There are fixed terminals on the top and bottom sides, and floating terminals on the left and right sides of the channel. Each set of points that need to be electrically connected is called a *net.* A net can connect terminals from the top and bottom of the channel and can exit the channel at the left and right sides. Wires of a net on different layers are connected by *vias.* A *channel* is a routing region bounded by two parallel rows of terminals. Without loss of generality, it is assumed that the two rows are horizontal. The top and the bottom rows are also called *top boundary* and *bottom boundary,* respectively. Each terminal is assigned a number, which represents the net to which that terminal belongs. Terminals numbered zero are called *vacant terminals.* A vacant terminal does not belong to any net and therefore requires no electrical connection. The net list of a channel is the primary input to most of the routing algorithms. The horizontal dimension of the routed channel is called the *channel length* and the vertical dimension of the routed channel is called the *channel height.* The horizontal segment of a net is called a *trunk* and the vertical segments that connect the trunk to the terminals are called its *branches.* A *dogleg* is a vertical segment that is used to maintain the connectivity of the two trunks of a net on two different tracks. An example of a CRP is shown in Figure 1.

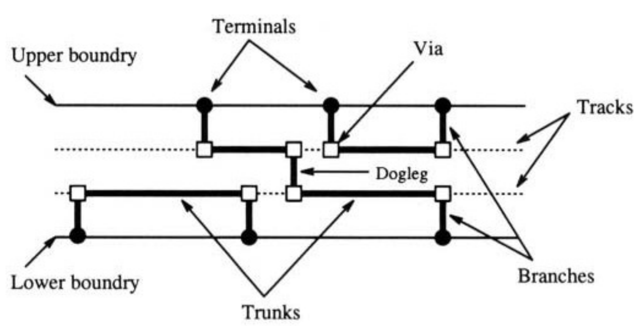

Figure1: An example of CRP.

2.1 Horizontal Constraints and Vertical Constraints

There is a horizontal constraint [5] between two nets if the trunks of these two nets overlap each other when placed on the same track. For a channel C the interval spanned by the net N_i, denoted by I_i is defined by where is the right most terminal of the net and where is the leftmost terminal of the net. Given a channel routing problem, a *horizontal constraint graph* (HCG) is a undirected graph $G_h = (V, E_h)$ where

$V = \{ V_i \mid N_i \in C \}$

$E_h = \{(V_i, V_j) \mid I_i \text{ and } I_j \text{ have non-empty intersection}\}$

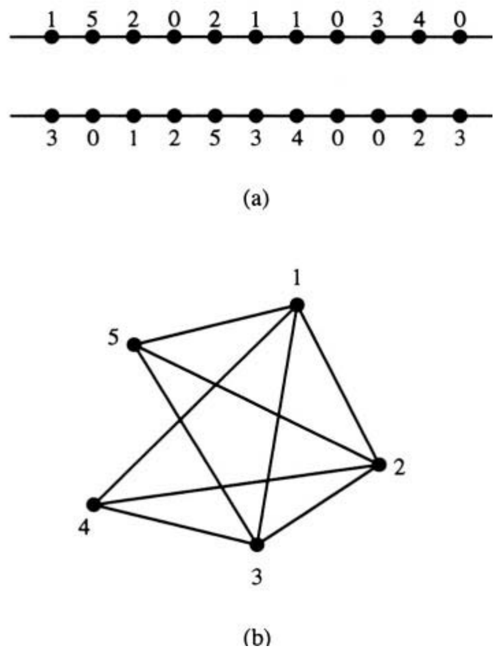

Figure 2: (a) A channel routing problem (b) Horizontal Constraint graph for the channel.

The HCG plays a major role in determining the channel height. In a grid based two-layer model, no two nets which have a horizontal constraint maybe assigned to the same track. As a result, the maximum clique in HCG forms a lower bound for channel height. In the two-layer grid-less model, the summation of widths of nets involved in the maximum clique determine the lower bound.

The constraints that two nets cannot overlap on the vertical layer are called the vertical constraints [5]. If net i connects to the c-th column in the top row and net j connects to the c-th column in the bottom row, $i \neq j$, then net i must be assigned to a track higher than net j. In this case, we say that there is a vertical constraint from i to j. Given a channel routing problem, a *vertical constraint graph (VCG)* is a directed graph $G_v=(V,E_v)$ where,

$V = \{V_i \mid N_i \in C\}$
$E_v = \{(V_i, V_j) \mid N_i \text{ has vertical constraint with } N_j\}$

It is easy to see that a vertical constraint implies a horizontal constraint; however, the converse is not true.

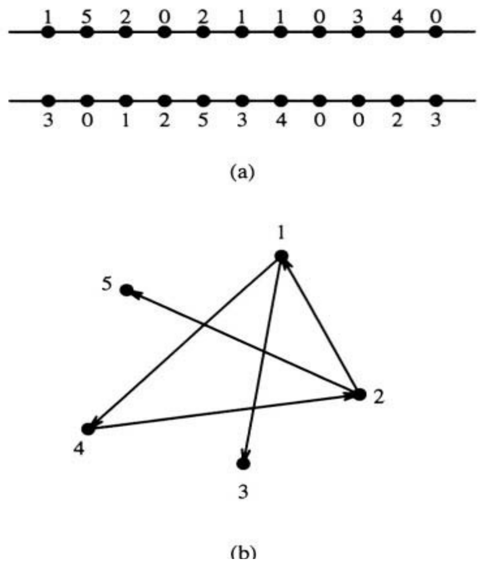

Figure 3: (a) A channel routing problem (b) The Vertical Constraint graph for the channel.

3. Genetic Algorithm

An overview of genetic algorithm presented in this paper is shown in Figure 4. The number of individuals N is kept constant throughout all generations.

```
Create_initial_population (Pc)
Fitness_calculation (Pc)
Pbest = Best_individual (Pc)
For generation = 1 to max_generation
   Pn=φ
   For offspring = 1 to max_decendent
        Pα = selection (Pc)
        Pβ = selection (Pc)
        Pn= Pn U crossover (Pα, Pβ)
    End for
    Fitness_calculation (Pn)
    Pc = Reduction (Pc U Pn)
    Mutation (Pc)
    Fitness_calculation (Pc)
    Pbest = Best_individual (Pbest U Pc)
 End for
```

Figure 4: Outline of the Genetic Algorithm.

4. Algorithm for Difficult CRP generator using Genetic Algorithm

4.1 Initial population creation:

We create an initial population of channels using repetitive call to the make_VC() and create _channel() algorithms.

Make_VC (N, p) is an algorithm that randomly generates the vertical constraint where N is the number of nets and p is the probability ($0 < p < 1$) for an edge from net i to net j to occur in the VCG.

To be a valid channel specification that can be routed without dogleg, the VCG must be cycle free. An algorithm make_VC(N, p), randomly generates the vertical constraints which is represented by a N X N matrix M_k in the k-th call. $M_k[i][j]=1$ denotes that there is a vertical constraint between net i and net j and $M_k[i][j]=0$ denotes there is no vertical constraint between net i and net j. To make VCG acyclic make_VC(N, p) generates $M_k[i][j]=1$ iff $i<j$.

The make _VC (N, p) algorithm is given below. In the algorithm random() is a random number generator which generates a real number in (0, 1).

Step1: for all i,j M_k [i][j]=0

Step2: For i=1 to N-1

For j=i+1 to N

If random()<p then

$M_k[i][j]=1$

Step3. End

For example two call to this algorithm may produces the following two matrices randomly, with N=5.

$$M_1= \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$

(a)

$$M_2= \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

(b)

Figure 5: Two Random Vertical Constraint Matrices generated by make_VC algorithm.

Create_channel() algorithm is used to create a channel from the matrix M_k generated by make_VC() algorithm. TOP_k and $BOTTOM_k$ are two net lists of a channel indicating top and bottom terminal respectively of the k-th channel. The algorithm to create k-th channel from the matrix M_k is given below:

Step1: P=1

Step2: For i = 1 to N-1

For j = i+1 to N

If $M_k[i][j] = 1$ then

$TOP_k[P]=i$

$BOTTOM_k[P]=j$

P=P+1

Step3: end

Thus the channel for the matrix M_1 and M_2 of Figure 1 is given below in Figure 5 and Figure 6 respectively.

TOP_1:	1 1 2 2 3 3
$BOTTOM_1$:	3 4 4 5 4 5

Figure 5: Channel for matrix M_1.

TOP_2:	1 1 2 3 4
$BOTTOM_2$:	2 4 3 5 5

Figure 6: Channel for matrix M_2.

In this way we create the initial population by making several call to make_VC() and create_channel() algorithms.

Assume that the initial population in our algorithm is as in Figure 7.

Channel 1	TOP_1 $BOTTOM_1$	1 1 2 2 3 3 3 4 4 5 4 5
Channel 2	TOP_2 $BOTTOM_2$	1 1 2 3 4 2 4 3 5 5
Channel 3	TOP_3 $BOTTOM_3$	1 2 3 4 2 4 4 5
Channel 4	TOP_4 $BOTTOM_4$	1 1 2 2 3 4 2 5 4 5 5 5

Figure 7: Initial population.

4.2 Fitness calculation:

Since we want to create a difficult channel we fixed up fitness function of a channel to be the total number of vertical and horizontal constraints in the channel. If we generate a channel with maximum fitness value we get a harder channel.

Fitness value of the channel C i.e. F(C) = Number of vertical constraints in channel C + number of horizontal constraint in channel C.

Number of vertical constraints in channel C can be obtained by counting number of columns in TOP or BOTTOM terminal of the channel C, where net number in TOP and BOTTOM are different.

Number of horizontal constraint can be obtained by the following way:

Step1: Find the span S_k of net k. If net k starts from column C_1 and ends in column C_2 then S_k contains all integers from C_1 to C_2 both inclusive.

Step2: Net N_i and N_j are horizontally constraint iff $S_i \cap S_j = \varphi$ if i<j. Count total number of horizontal constraint.

As an example, the fitness value is calculated for channel 1 of Figure 7.

TOP_1	1	1	2	2	3	3
$BOTTOM_1$	3	4	4	5	4	5

Vertical constraint=number of columns in the channel=6

To calculate horizontal constraint we first calculate the span of each net.

S_1={1,2}

S_2={3,4}

S_3={1,2,3,4,5}

S_4={2,3,4,5,6}

S_5={4,5,6}

Now, since $S_1 \cap S_3 \neq \varphi$, $S_1 \cap S_4 \neq \varphi$, $S_2 \cap S_3 \neq \varphi$, $S_2 \cap S_4 \neq \varphi$, $S_2 \cap S_5 \neq \varphi$, $S_3 \cap S_4 \neq \varphi$, $S_3 \cap S_5 \neq \varphi$, $S_4 \cap S_5 \neq \varphi$ horizontal constraint in the channel is 8.

Hence fitness of the channel is =6 + 8=14.

Similarly we calculate the fitness of the all channels in the initial populations of Figure 7 in Figure 8.

Channel 1	TOP_1 $BOTTOM_1$	1 1 2 2 3 3 3 4 4 5 4 5	Fitness=14
Channel 2	TOP_2 $BOTTOM_2$	1 1 2 3 4 2 4 3 5 5	Fitness=12
Channel 3	TOP_3 $BOTTOM_3$	1 2 3 4 2 4 4 5	Fitness=8
Channel 4	TOP_4 $BOTTOM_4$	1 1 2 2 3 4 2 5 4 5 5 5	Fitness=13

Figure 8: Fitness of initial populations of Figure 7.

4.3 Crossover:

To perform crossover following steps are done:

Step1: Select two channels P and Q from the initial population.

Step2: Find the length of channel P and Q i.e. number of columns in P and Q and denote them C_P and C_Q respectively.

Step3: Find minimum between C_P and C_Q and let it C_{min} i.e. $C_{min}=C_P$ if $C_P<C_Q$, $C_{min}=C_Q$ if $C_Q<C_P$ and if $C_P=C_Q$ then C_{min} is either C_P or C_Q.

Step4: Generate a random number between 1 and C_{min} and assume it to be C_{rand}.

Step5: Find two new channels by cross over at column C_{rand}. One channel is generated by taking column 1 to column C_{rand} from P and column C_{rand} + 1 to last column from Q. Another channel is generated from column 1 to column C_{rand} from Q and column C_{rand} + 1 to last column from P.

We illustrate the procedure with an example.

Step1: Chose P and Q from our initial population of Figure 4 as

P:

TOP_1 $BOTTOM_1$	1 1 2 2 3 3 3 4 4 5 4 5

Q:

TOP_2 $BOTTOM_2$	1 1 2 3 4 2 4 3 5 5

Step2: Length of P i.e. C_P=6 and length of Q i.e. C_Q=5.

Step 3: C_{min} = 5

Step4: Let a random number between 1 and 5 be 2, i.e. C_{rand}=2

Step5: Two channels are generated by cross over as:

TOP_1	1	1	2	2	3	3
$BOTTOM_1$	3	4	4	5	4	5

TOP_2	1	1	2	3	4
$BOTTOM_2$	2	4	3	5	5

Then two newly generated channels are

New_1:

1	1	2	3	4
3	4	3	5	5

New_2 :

1	1	2	2	3	3
2	4	4	5	4	5

Now the fitness function of new solutions are calculated and F(New_1)=12 and F(New_2)=13.

4.4 Mutation:

In mutation operation we chose two random columns between 1 and C_{max} (C_{max} is the maximum column of the channel) and swap this two columns since it may increase number of horizontal constraint.

We illustrate the procedure with the channel New_2 generated from crossover operation.

Assume two random numbers between 1 and 6 (hare C_{max}=6) are 2 and 5.Thus new channel (New_3) is generated by swapping column 2 and column 5 in channel New_2 as

New_3

1	3	2	2	1	3
2	4	4	5	4	5

Now the fitness of the channel New_3 is 16.

4.5 Two or multi terminal Channel generation

The final channel obtained by genetic algorithm describe above required some modification such that all the nets are at least two terminal in the channel. There are two cases:

Case 1: Net N has no terminal in the channel. In this case we introduce two columns in the channel one with top terminal N and bottom terminal 0 or vice versa and another with top terminal 0 and bottom terminal N or vice versa. We introduce these two channels in two extreme of the channels such that it increase horizontal constraint.

Case 2: Net N has only one terminal in the channel. In this case we introduce only one column with top terminal N and bottom terminal 0 or vice versa and introduce the column in one of the extreme of the channel so that it is furthest from the column that contains N, because it increases the horizontal constraint.

5. Result

This algorithm generates difficult random channel routing instances of arbitrary size using Genetic Algorithm. Several executions generate random channels with different size. In all cases, we find better solution than well known benchmark channels. The generated random channels are difficult to route since they have higher vertical and horizontal constraints. As an example, we compare our randomly generated channels with well known benchmark channel DDE. DDE has 72 nets and 174 columns. For comparison purpose we generate our random channels with fixed length i.e. 174 columns and 72 nets. We generate five such random channels. The solutions are given with their fitness value in Figure 9. The channel in DDE with its fitness is given in Figure 10. In all cases we find that our algorithm generates channels with higher fitness value, i.e. channels which are difficult to route.

5. Conclusion

In this paper we generate random difficult channel of arbitrary length which are guaranteed to be routed using two layer no dogleg routing algorithms. This algorithm can also be extended such that it generates such difficult channels that cannot be routed using two layer no dogleg routing algorithms and can be routed using two layer dogleg or three layer or multi layer routing algorithms.

Sample	Terminal	Generated channel	Fitness value
1	TOP	46 45 01 02 02 02 03 03 03 44 04 43 05 05 05 06 06 06 42 07 41 08 08 08 09 40 09 10 10 10 11 11 11 39 12 12 38 13 13 14 37 14 15 15 15 16 16 36 17 35 17 35 18 18 19 34 19 34 20 20 33 21 21 22 32 22 23 23 23 24 24 31 25 30 25 26 26 26 27 27 27 28 28 28 29 29 29 30 30 25 31 31 24 32 32 32 33 21 20 34 34 19 35 18 17 16 16 36 37 37 14 13 38 12 12 39 11 10 40 09 08 08 07 42 07 06 43 43 04 04 03 44 02 02 01 01 46 57 47 47 56 55 55 54 54 49 49 50 50 50 51 51 51 52 52 52 53 53 53 49 54 48 48 55 48 56 47 56 57 57 46 58 58 58	1502
	BOTTOM	47 48 15 05 06 08 05 06 09 45 07 47 06 07 08 07 10 16 45 12 51 10 11 13 11 43 14 11 13 15 15 16 18 40 18 19 45 19 20 17 41 21 18 19 20 18 19 37 20 39 22 37 20 22 20 40 26 38 25 27 41 24 25 23 39 28 25 28 29 25 28 32 29 37 33 27 28 29 30 31 33 29 30 33 30 31 32 31 32 32 32 33 25 34 37 39 36 23 25 37 38 23 36 19 21 22 19 42 38 40 18 20 45 18 17 41 16 15 42 13 13 11 14 44 09 10 44 45 08 05 06 49 08 05 12 05 51 61 48 49 58 61 57 63 61 51 53 53 56 57 53 54 60 53 55 59 54 57 59 50 62 51 50 58 49 57 50 59 58 60 52 59 61 63	

2	TOP	01 31 01 31 02 30 30 03 03 04 29 04 29 05 28 06 28 06 07 27 07 08 08 08 09 09 09 26 10 25 11 11 11 12 12 12 13 13 13 14 23 14 15 15 15 16 16 16 17 17 17 18 18 18 19 19 19 20 20 20 21 21 21 22 22 22 23 14 23 24 24 12 25 11 10 26 26 09 27 08 07 06 06 05 29 05 04 03 03 02 02 01 31 32 32 57 57 56 55 34 34 54 35 53 52 52 51 45 37 50 49 49 44 38 39 46 39 40 43 44 41 41 41 42 42 42 43 40 43 44 38 40 38 37 45 36 39 46 47 47 47 48 48 48 45 49 37 45 50 50 51 36 51 46 52 35 53 35 53 54 34 54 55 55 33 56 33 56 33 32 57 58 58 58	1232
	BOTTOM	06 35 12 34 05 40 36 06 07 12 35 17 32 07 36 07 32 11 10 31 13 09 10 11 12 15 16 27 14 31 13 14 19 15 17 18 17 20 24 15 27 20 16 17 19 19 20 22 18 19 20 20 21 23 20 23 26 23 25 26 22 27 28 23 25 30 26 16 28 25 26 17 28 13 15 27 30 15 28 09 11 11 10 12 30 06 16 07 04 06 04 09 38 33 34 60 59 59 58 35 39 60 37 55 56 54 53 54 43 51 62 53 47 43 42 51 45 41 45 49 47 48 52 43 44 45 44 42 46 46 42 43 39 45 54 43 43 53 48 49 54 49 52 56 52 61 46 53 52 53 52 44 55 50 55 40 54 39 56 56 40 61 56 57 36 57 35 61 34 35 61 59 61 63	
3	TOP	01 01 01 02 02 02 03 52 03 51 04 51 05 51 50 06 50 06 49 07 49 08 49 08 09 09 48 48 10 47 11 11 11 12 12 12 13 46 45 14 45 14 44 15 15 16 16 16 43 17 17 18 18 42 19 19 19 41 20 20 21 21 40 22 40 22 23 23 23 24 39 38 25 25 25 26 26 37 27 27 27 28 28 28 29 29 29 30 30 30 31 31 31 32 32 32 33 33 33 34 34 34 35 35 35 36 36 36 37 37 26 38 25 24 24 23 39 40 22 21 21 20 20 19 42 18 18 43 17 16 15 15 45 14 13 46 12 12 47 47 47 48 09 09 08 07 07 06 50 05 05 04 04 03 52 52 53 53 53 54 54 54 55 55 55 56 56 56 57 57 57 58 58 58	1753
	BOTTOM	02 04 05 06 08 13 08 54 10 60 12 59 06 52 56 07 54 11 56 11 55 11 52 14 10 14 54 53 13 51 12 13 16 14 16 17 14 47 53 16 52 25 49 18 21 19 20 21 48 21 24 21 22 48 20 22 27 48 24 25 24 27 46 25 44 28 24 25 27 26 41 41 27 28 31 27 28 46 32 33 36 30 31 33 30 32 33 31 32 34 34 35 36 34 38 39 35 36 39 35 36 37 36 37 38 37 38 41 42 44 29 39 28 31 30 27 44 42 26 28 24 25 21 22 44 23 21 46 20 20 21 17 49 23 16 47 17 14 48 51 52 53 15 10 13 12 08 08 55 08 07 15 11 09 55 59 56 57 59 55 57 60 57 58 61 60 61 62 59 67 69 61 62 65	
Sample	Terminal	Generated channel	Fitness value

4	TOP	01 39 26 38 25 24 03 24 23 22 22 21 20 05 20 19 19 18 17 07 16 08 16 15 14 14 09 10 10 10 11 11 11 12 12 12 13 13 13 09 14 09 15 08 15 08 07 16 17 17 07 18 06 18 06 19 06 05 20 05 21 04 21 04 22 04 23 03 23 03 02 24 02 25 02 26 01 26 01 27 27 58 28 58 29 29 29 57 30 57 31 56 56 32 56 32 55 33 33 34 54 34 35 35 35 53 36 53 37 52 37 38 38 25 27 39 51 40 40 40 50 41 41 42 42 42 43 43 43 44 44 44 45 45 45 46 46 46 47 47 47 48 48 48 49 49 49 50 50 41 40 51 39 38 52 52 53 36 36 54 35 34 33 33 55 56 56 56 30 30 29 28 58 28	1471
	BOTTOM	02 42 32 41 27 29 04 25 26 26 23 26 28 07 21 26 22 20 23 14 20 09 18 17 23 19 13 16 18 20 14 15 17 15 18 20 15 16 17 12 20 11 16 11 18 10 15 23 20 21 09 19 10 23 09 24 08 08 24 06 23 08 27 07 25 06 25 06 29 05 12 31 04 28 03 30 06 34 03 30 31 61 31 59 31 34 35 62 32 60 32 71 65 33 63 36 57 35 36 35 61 37 43 44 48 59 42 57 38 62 47 39 40 29 29 44 54 41 42 43 56 45 48 45 51 55 45 47 52 45 46 49 46 50 52 47 48 55 49 50 53 49 50 51 50 55 57 52 54 44 42 54 42 41 61 62 55 43 39 55 43 36 36 34 59 63 65 71 33 31 34 32 60 30	
5	TOP	56 57 01 02 02 02 03 01 58 01 57 03 56 58 05 54 54 53 07 52 57 08 08 50 49 49 48 48 47 46 11 46 45 45 44 43 13 42 13 42 41 40 15 39 38 16 38 37 37 36 35 18 34 18 19 33 32 32 31 30 30 21 29 29 22 27 23 23 23 24 24 24 25 25 25 26 26 26 27 27 22 28 28 28 22 29 21 30 21 20 31 20 31 20 32 19 33 19 33 34 18 34 35 35 17 36 17 36 17 37 16 38 16 15 39 15 39 40 40 14 41 14 41 14 13 42 43 43 12 44 12 44 12 45 11 11 46 10 47 10 47 10 48 09 49 09 09 50 08 50 51 51 07 52 07 52 53 06 53 06 54 06 55 05 55 05 56 04 51 04 04 58 03 58	2010
	BOTTOM	62 61 06 03 04 11 05 02 62 03 60 18 58 72 08 60 55 58 09 54 59 09 10 52 56 51 51 49 51 53 12 47 53 48 49 49 14 45 18 44 46 48 19 41 43 19 41 41 38 43 43 20 36 26 27 36 37 34 33 33 32 27 40 30 29 35 24 25 26 25 26 31 26 28 30 30 31 32 30 33 32 29 30 31 26 32 28 31 22 27 32 26 35 24 35 37 35 34 37 35 21 37 37 38 24 38 22 46 19 39 21 39 20 22 40 20 43 43 47 21 43 17 49 16 17 46 46 48 17 45 15 50 13 49 15 13 50 14 50 13 52 11 50 14 50 13 12 51 11 53 52 53 15 53 13 55 56 12 60 10 58 09 57 07 62 06 59 13 55 12 09 61 20 72	

Figure 9: Random channels generated by our algorithm with 72 Nets and 174 Columns.

Benchmark Channel	Terminal	Channel	Fitness Value

<table>
<tr><td rowspan="2">DDE</td><td>TOP</td><td>03 05 07 09 05 12 14 15 07 12 14 07 04 13 08 06 15
18 14 08 06 11 22 21 00 18 16 18 16 00 08 06 26 11
00 24 23 25 20 01 29 00 22 03 22 03 00 00 09 02 09 02
00 32 23 33 19 06 08 30 27 34 35 36 37 39 31 39 35 38
31 08 30 37 41 19 06 44 45 00 33 31 33 31 00 27 35 36
48 49 31 39 46 47 50 52 20 53 24 00 47 39 00 24 51 20
52 20 52 23 08 30 50 56 00 00 57 49 19 06 06 19 49 59
00 00 61 50 30 08 55 00 24 64 20 52 00 67 68 63 55 24
52 20 69 24 00 46 62 63 68 00 24 65 20 52 00 70 60
62 54 63 00 24 71 20 52 06 07 00 00 00 00 00 00 00</td><td rowspan="2">1075</td></tr>
<tr><td>BOTTOM</td><td>02 04 06 08 10 11 13 03 09 16 05 17 11 05 14 14 07 12
17 19 01 20 21 23 24 00 16 10 03 11 25 00 26 11 26 11
00 27 28 11 03 09 16 30 27 05 31 01 05 01 20 32 23 24
00 09 01 20 29 23 24 00 03 08 30 38 28 19 06 40 27 35
41 42 06 19 34 43 30 08 31 43 39 46 36 46 47 48 31 00
24 23 45 20 01 51 00 40 39 40 39 00 08 30 50 54 00 00
55 49 19 06 00 47 42 47 42 00 53 58 06 19 49 50 30 08
60 62 59 54 55 54 56 63 55 65 00 66 68 06 06 68 00 60
68 00 46 44 46 44 00 69 00 55 05 08 55 58 00 64 71 00
72 63 72 63 00 57 62 54 70 67 55 61 63 68</td></tr>
</table>

Figure 10: Deutsch's Difficult Example (DDE) without left or right constraint

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley Publishing Company, San Francisco, CA, 1976.

[2] D. N. Deutsch. A dogleg channel router. In *Proceedings of the 13th Design Automation Conference,* pages 425-433, 1976.

[3] S. Haruyama, D. F. Wong, and D. S. Fussell. Topological channel routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* 11(10):1177-1197, October 1992.

[4] R. K. Pal, S. P. Pal., and *A*. Pal. A new lower bound for channel routing. In *Proceedings of the IEEE Region 10 Conference on Computer, Communication, Control and Power Engineering,* pages *507-510,*October *1993.*

[5] Naveed A. Sherwani, Algorithms for VLSI Physical Design Automation, 2nd edition.

[6] Heng Yi Chao and Mary P. Harper, A Difficult Channel Routing Generator, Electrical and Computer Engineering, ECE Technical Reports, Purdue Libraries, 1995.