STOCHASTIC ESTIMATION

OF

CHANNEL ROUTING TRACK DEMAND

John Y. Ngai

Computer Science
California Institute of Technology

5094:TR:84

California Institute of Technology

Department of Computer Science

Stochastic Estimation of Channel Routing Track Demand

John Y. Ngai

5094:TR:84

The work described in this document was sponsored by the Honeywell Research Fellowship, 1983 and was co-sponsored by the Defense Advanced Research Projects Agency, ARPA order number 3771, and monitored by the Office of Naval Research under contract number N00014-79-C-0597

1. Introduction

Channel routing is a special approach to the general interconnect problem where connections are made within a rectangular strip having no obstacles inside. Fixed terminals are located on one pair of opposite sides and flexible terminals on the other pair of opposite sides of the channel. It is an important part of many general layout systems [1][2], where the whole routing plane is divided into many smaller rectangular channels. The problem of finding optimal channel routing patterns has been extensively studied and many good heuristics exist [3][4][5].

In this report we are concerned with a problem different from the usual channel routing problem but related to it. In particular, we are interested in stochastic estimation of track demand for the channel routing problem. The significance of this problem becomes apparent as the size of circuit integration increases and top down methods are employed in the design of VLSI circuits [6][7].

During the top down floorplanning, one is concerned with logic partitioning, placement of circuit modules and global wiring organization. At this stage, the actual circuit modules have not yet been laid out. A stochastic model for channel routing track demand can be employed to guide the floorplanning steps to avoid possible local channel congestions. Stochastic models for routing demand have been investigated by Heller et al. [8] and El Gamal et al. [9][10]. In these models the wire generation distribution is assumed to be Poisson and are concerned with average or asymptotic behaviors of routing demands. While such models are excellent for estimating overall wiring space requirements, they could not be applied to compute estimations for specific channel layouts. In this paper, we will describe models based on Markov Chains which enable us to compute demand distributions for specific channels.

2. Informal Description of Problem

The model of the channel is depicted in Fig. 1. It consist of a rectangle with fixed terminals on its top and bottom boundaries and flexible terminals on its left and right boundaries. The signals that originate from the left and end on the right are called *pass signals.* Signals that originate from the top or bottom and end on the top or bottom are termed as *center signals.* Signals originate from the left and end on the top or bottom

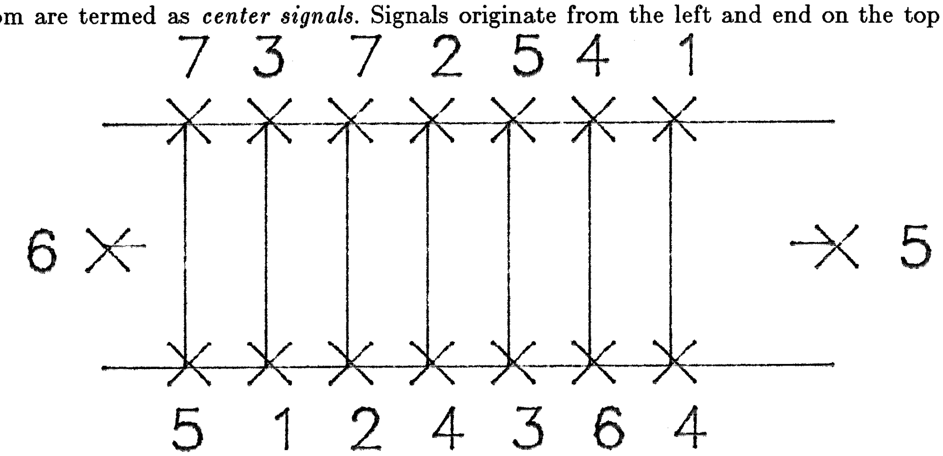

Figure 1. Example of a Routing Channel.

are termed as ***left signals***. ***Right signals*** are similarly defined. It is clear that pass signals do not contribute to the uncertainty in channel track demands, and they will be ignored in the following discussion.

The channel track demand in general depends not only on the terminal locations and net specifications, but also on the actual routing algorithm employed. Even for fixed terminal locations, a theoretical exact track demand prediction is still not available although there are results concerning bounds [11]. To avoid results which depend on specific routing algorithms involved, we will describe situations where most existing "good" channel routers all produce the same number of tracks needed.

The general channel routing problem has two graphs associated with it, namely a vertical constraint graph (VG) and an interval graph (IG). Both graphs have as their set of vertices the set of nets. On each column having different signal pins on both top and bottom boundaries, a directed arc is drawn from the lower signal to the upper signal (see Fig. 2.). The VG displays the vertical routing constraints. It must be free of circuits if the channel routing problem is free of constraint loops. Doglegging is necessary to break constraint loops in order to produce a feasible routing pattern.

With each signal, we associate an interval from its leftmost pin of the signal to its rightmost pin. An edge is added to the interval graph IG between two vertices, when the corresponding intervals intersect each other (see Fig. 3.). It is easy to see that the maximum clique number of IG is simply the channel density, and represents an absolute lower bound on channel track demand. As is shown in [11], actual lower bound on track demand can be much higher than that of the channel density.

In the following discussion, we will concentrate on the cases when the channel utilization factor[1] is low ($\leq$0.2). Under this assumption, the channel density becomes the dominant factor in determining channel track demand. (See Table 1.) We will concern ourselves with estimations of specific channels. Then, we will discuss routing demand for channels having a fixed set of signal connections, but otherwise totally random permutations among the signals. This model could be useful during top down wiring organizations to estimate channel congestion when signals are being assigned to specific edges of circuit modules and channels. Next, we

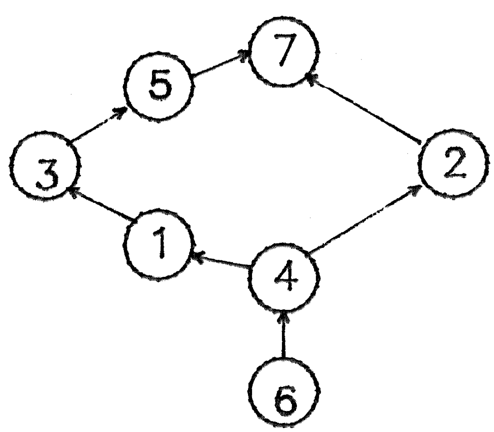

Figure 2. Vertical Constraint Graph.

[1] ratio of number of pins to number of grid slots in channel

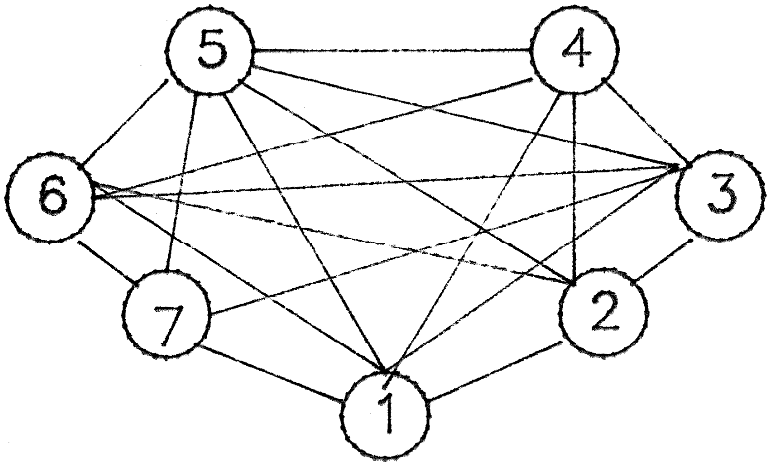

Figure 3. Interval Graph.

will discuss routing demand for channels having a predefined signal ordering among its pins, this being the most natural information to be obtained from a circuit designer based on his knowledge of the specific circuits under consideration.

3. Random Permutations of Signals

In the following discussion, under the assumption of low channel utilization factors, we make three more simplifications :

(1) The channel is collapsed into a singled sided channel making no distinction between the top and bottom pins.

(2) All nets involved are two terminal nets. The embedded stochastic process with state transitions defined at terminal locations is Markovian.

(3) The length of the channel is equal to the exact number of pin slots needed to accommodate all the pins involved.

The single sided representation is also used in [7]. (See Fig. 4.) The main difference between the single sided representation and the double sided one occurs when we have exact alignment of terminals from the same net in the same column. In the former representation, an extra demand of 1 is always generated locally, while the latter case does not. Another such local discrepancy occurs when the top and bottom pins are the opposite extreme pins of two different nets. Our assumption is that the number of cases with exact alignment of terminals is much smaller in comparison to those with no alignments.

A multiple-pin net can be abstracted by representing it using only its leftmost (head) pin and its rightmost (tail) pin. All intermediate pins will not modify the channel density and hence the track demand. Probabilistically, in the limit of channel utilization factor going to zero, the stochastic behavior of a multiple-terminal net approaches that of the ideal two-terminal net.

Having made the first two simplifications, the third follows naturally. Extra pin slots no longer affect the stochastic behavior of channel density. They simply multiply all pin permutations by the same amount.

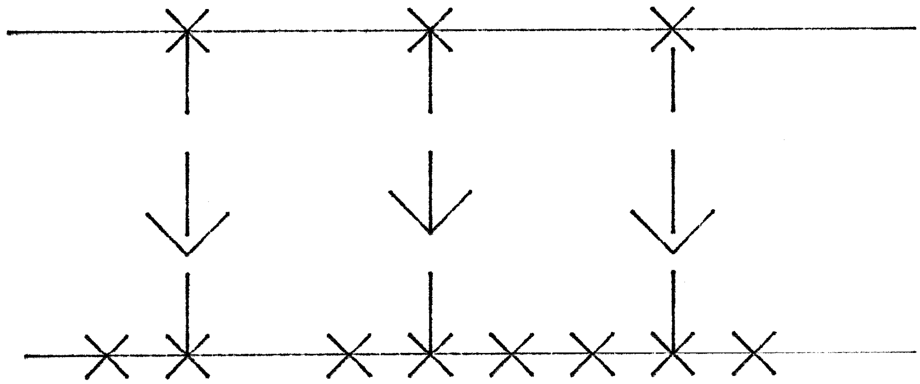

Figure 4. Single Sided Channel Representation.

Or alternatively, the length of the channel is equal to the number of state transitions for the embedded Markov Chain mentioned.

We will justify these simplifications by comparing to results obtained by actual simulations.

To proceed with the discussion of the model, we define the following constants :

L	= Number of left signals.
R	= Number of right signals.
S	= Number of center signals.
N	= Total number of pins lying inside the channel.
B_d	= Total number of routing tracks available.

Similarly, we define the following random variables :

I(n)	= Number of started center signals up until column n.
J(n)	= Number of finished center signals up until column n.
P(n)	= Number of left pins up until column n.
Q(n)	= Number of right pins up until column n.
D(n)	= Local density of the channel at column n.

It is obvious that :

$$N = L + R + 2S \tag{3.1}$$

$$n = I(n) + J(n) + P(n) + Q(n) \tag{3.2}$$

and

$$D(n) = L + I(n) - J(n) + Q(n) - P(n) \tag{3.3}$$

We define a state of the channel at column n by the 5-tuple :

$$(I(n), J(n), P(n), Q(n), n)$$

The last entry is in fact equal to the sum of the first four entries, but is put in for clarity.

The conditional transition probabilities for the embedded Markov Chain are listed below:

$$P\,[(i\,,j\,,p\,,q\,,n\,)\rightarrow(i+1,j\,,p\,,q\,,n+1)] = \frac{2(S-i)}{N-n} \tag{3.4}$$

$$P\,[(i\,,j\,,p\,,q\,,n\,)\rightarrow(i\,,j+1,p\,,q\,,n+1)] = \frac{i-j}{N-n} \tag{3.5}$$

$$P\,[(i\,,j\,,p\,,q\,,n\,)\rightarrow(i\,,j\,,p+1,q\,,n+1)] = \frac{P-p}{N-n} \tag{3.6}$$

$$P\,[(i\,,j\,,p\,,q\,,n\,)\rightarrow(i\,,j\,,p\,,q+1,n+1)] = \frac{Q-q}{N-n} \tag{3.7}$$

Intuitively, we can understand these transition probabilities as follows: *(N-n)* is the total number of pins yet to be inserted into the channel; *(i-j)* is the number of center nets with head pins inserted but not their tail pins; *(P-p)* is the number of left pins yet to be inserted; *(Q-q)* is the number of right pins yet to be inserted; and *(S-i)* the number of center nets yet to be started, and *2(S-i)* the corresponding number of pins to be inserted. It is not hard to see that they add up to unity as we would expect. From these transition probabilities, we can write down the Forward Kolmogorov Equation for the Markov Chain :

$$O\,(i\,,j\,,p\,,q\,,n+1) = O\,(i-1,j\,,p\,,q\,,n\,)\left(\frac{2(S-i+1)}{N-n}\right) + O\,(i\,,j-1,p\,,q\,,n\,)\left(\frac{i-j+1}{N-n}\right) +$$

$$O\,(i\,,j\,,p-1,q-1,n\,)\left(\frac{P-p+1}{N-n}\right) + O\,(i\,,j\,,p.q-1,n\,)\left(\frac{Q-q+1}{N-n}\right) \tag{3.8}$$

Where O(i,j,p,q,n) is the occupancy probability for the state (i,j,p,q,n).

In general these transition probabilities govern the behavior of the random variable D. The state of the channel starts at (0,0,0,0,0) and gradually ***diffuses*** out to encompass a whole set of states of varying likelihood. These states will also have varying D values associated with them. Our problem is to determine the distribution of the variable D corresponding to these states.

Qualitatively, when we ask for the probability of a successful channel track capacity assignment under our assumptions, we are putting a boundary on the possible range of D. Alternatively, we can view the stochastic process as a ***diffusion*** of probabilities from the source at state (0,0,0,0,0) to the drain at state (S,S,P,Q,N). The boundary on D now becomes an ***absorbing*** boundary for the probabilities. It is with this view that we develop the algorithm as in ALGORITHM A.

For the special case of having only left and right signals and no center signals, the distribution can be expressed analytically in closed form by applying the classical Ballot Theorem in combinatorics [12]. Without loss of generality, we assume R $\geq$ L, then we have

ALGORITHM A

```
ALGORITHM A
INPUT:
        left    = number of left signals
        center  = number of center signals
        right   = number of right signals
        bound   = track capacity limit
OUTPUT:
        probability of channel density not exceeding bound
BEGIN
        len := 2*center + 2*left + right
        set all state[k,p] to 0
        center := center + left
        state[left, left] := 1
        l := 0
        FOR n:=left+1 to len DO
        BEGIN
                FOR k:= 0 to bound DO
                BEGIN
                        i := (n+k) div 2
                        FOR 0≤p≤center and 0≤(i-p)≤right DO
                        BEGIN
                                update state[k,p] by transitions from
                                state[k-1,p], state[k-1,p-1] and state[k+1,p]
                                according to the Kolmogorov Equation
                        END
                END
        END
        probability := state[right,center]
END
```

$$\textit{Probability of success} = 1 - \frac{\binom{N}{\frac{N+2B_d-R-L+2}{2}}}{\binom{N}{\frac{N+R-L}{2}}} \tag{3.9}$$

where we follow the usual conventions for the binomial coefficients. The second term in Equation (3.9) is simply the number of lattice paths which cross the boundary B_d divided by the total number of lattice paths possible.

Another special case occurs when we have only center signals but no left or right signals. Here one can write down the generating function F(X) for the number of permutations not exceeding a specified bound in terms of a continued fraction expansion [13] whose denominators are reciprocal Hermite polynomials :

$$F(X) = \cfrac{1}{1-\cfrac{X^2}{1-\cfrac{2X^2}{1-\cfrac{3X^2}{1-\cfrac{\cdots}{1-\cfrac{\cdots}{1-B_d X^2}}}}}} \tag{3.10}$$

By finding the dominant roots of these polynomials, one can obtain asymptotic results for this special case.

4. Predefined Channel Pin Order

In the previous section, we were concerned with distribution of track demand for pins located randomly across the channel. In the top down floor-planning process, the next refinement made is to determine the exact order of net terminals along the top and bottom edges of the channel. Probabilistically, the uncertainty in channel density arises from the uncertainty in relative locations of the top and bottom pins. Intuitively, however, we would expect the uncertainty to be much smaller in comparison to those described, as a result of our marching closer to the deterministic realization of the channel. This order information is the most naturally obtained from the circuit designer once major placement of circuit modules is completed.

Again, under the assumption of a low channel utilization factor, we make the following simplifications :

(1) In the limit of channel utilization factor going to zero, the channel can be collapsed to a single sided representation as before.

(2) The embedded process with state transitions defined at pin locations is Markovian.

(3) The length of the channel is equal to the exact number of pin slots needed to accommodate all the pins involved.

Single sided representation is again used to restrict state transitions to one pin at a time. It suffers the same difficulty concerning exact signal pins alignment as the previous model does. Again the rationale is the assumption that the number of premutations with alignments is much smaller in comparison to that of no alignments; further that it is very unlikely to affect the channel density when they do occur. This assumption, of course, relies heavily on the original assumption of a low channel utilization factor.

Notice that we do not restrict the nets involved be two-terminal nets, although in the limit of channel utilization factor going to zero, two-terminal nets are valid abstractions.

The length of the channel is simply equal to the number of state transitions for the embedded Markov Chain with state transitions defined at pin locations.

Hence, here we have instead of random distribution of net pins, two vectors of predefined net orders : one for the top edge and one for the bottom edge. We also have two set of nets : one denotes the set of nets coming in from left edge of channel, and one denoted the set of nets going out at right edge of channel. Our objective here again is to determine the distribution of the channel density. We define the following constants :

X = Total number of pins on top edge of channel.

Y = Total number of pins on bottom edge of channel.

N = Total number of pins from both top and bottom edges of channel.

B_d = Total number of routing tracks available.

Similarly, we define the following random variables :

T(n) = Number of top pins already inserted up until column n.

B(n) = Number of bottom pins already inserted up until column n.

D(n) = Local density of channel at column n.

It is obvious that :

$$n = T(n)+B(n) \tag{4.1}$$

We define a state of the channel at column n by the 3-tuple :

$$(T(n),B(n),n)$$

Again the last entry is inserted only for clarity.

The conditional transition probabilities for the embedded Markov Chain are listed below:

$$P[(t,b,n)\rightarrow(t+1,b,n+1)] = \frac{X-t}{N-n} \tag{4.2}$$

$$P[(t,b,n)\rightarrow(t,b+1,n+1)] = \frac{Y-b}{N-n} \tag{4.3}$$

These transition probabilities can be understood intuitively as the ratio of the number of top (bottom) pins yet to be inserted to the total number of pins yet to be inserted. We can use these transition probabilities to write down the Forward Kolmogorov Equation :

$$O(t,b,n+1) = O(t-1,b,n)\left(\frac{X-t+1}{N-n}\right) + O(t,b-1,n)\left(\frac{Y-b+1}{N-n}\right) \tag{4.4}$$

Where O(t,b,n) is the occupancy probability of state (t,b,n).

From the predefined pin orders, and the set of left and right signals, it is possible to determine the local density at column n given T(n) and B(n), which we denote by D^*.

In the following discussion, we assume we have a table of D^*'s either given or computed. We present below an algorithm to compute the probability of channel density not exceeding a specified bound. It proceeds like our previous algorithm A. by computing iteratively state occupancy probabilities using the Forward Kolmogorov Equation. The specified bound on channel density again plays the role of an absorbing boundary.

We present the algorithm in ALGORITHM B.

ALGORITHM B

```
ALGORITHM B
INPUT:
        leftSize = number of left signal nets
        topSize = number of top signal pins
        bottomSize      = number of bottom signal pins
        bound           = specified track limit
        table           = table of D*'s
OUTPUT:
        probability of channel density not exceeding bound
BEGIN
        len := topSize + bottomSize
        set all state[i,j] to 0
        state[0,0] := 1
        FOR n :=1 to len DO
        BEGIN
                FOR k:=0 to bound DO
                BEGIN
                        FOR i+j=len and 0≤i≤topSize and 0≤j≤bottomSize DO
                        BEGIN
                                IF table[i+1,j]=k
                                        THEN  update state[i+1,j] by
                                              transition from state[i,j]
                                              according to Kolmogorov Equation
                                IF table[i,j+1]=k
                                        THEN  update state[i,j+1] by
                                              transition from state[i,j]
                                              according to Kolmogorov Equation
                        END
                END
        END
        probability := state[topSize, bottomSize]
END
```

5. Comparison of Results

In table 1. we list the statistical success ratio of randomly generated examples routed by a modified version of the Greedy Router by Rivest and Fiduccia[3] implemented by the author.[14] The ratio are obtained by allowing track capacities only upto the channel densities. The channel utilization factor ranges from 0.05 to 0.3 . From the results, it is clear that for low utilization factors, there is plenty of room for constraint resolving.

We listed at the end of this report figures obtained by simulations of random channel pin locations and/or signal assignments and figures obtained by direct computations. Tables 2. to 7. are five examples of specific channels. Notice for cases where left and right signals are absent, the 50th percentile occurs just a few tracks above half the number of center signals involved. This has also been observed for many other runs of simulations and computations. The nets generated in simulations are Poisson with λ equal to 3 in the number of pins per net. The figures show close agreement with results computed by using the two-pin net assumption.

Tables 8. to 12. are five examples of specific channels with predefined signal orders on opposite edges. It is clear from the figures we obtained that the variance of the distributions are much less than those obtained with total random permutations. In almost all cases, we obtain a less than 4 % error between the simulated results and the computed results. More extensive simulations and comparisons however, are definitely needed.

References

[1] R.L Rivest, "The *PI* (Placement and Interconnect) System." *Proceedings of the 19th Design Automation Conference (1982).*

[2] J.E. Hassett, "Automatic Layout in ASHLAR: An Approach to the Problems of "General Cell" Layout for VLSI." *Proceedings of the 19th Design Automation Conference (1982).*

[3] R.L. Rivest and C.M. Fiduccia, "A *Greedy* Channel Router." *Proceedings of the 19th Design Automation Conference (1982).*

[4] T. Yoshimura and E.S. Kuh, "Efficient Algorithms for Channel Routing." *IEEE Trans. CAD of ICs and Systems CAD-1 (1) (1982).*

[5] M. Burstein. and R. Pelavin, "Hierarchical Channel Router." *Proceedings of the 20th Design Automation Conference (1983).*

[6] Ralph H.J.M. Otten, "Automatic Floorplan Design." *Proceedings of the 19th Design Automation Conference (1982).*

[7] D.W.Hightower, "The Interconnection Problem: A Tutorial." *Computer 7,4 (April 1974).*

[8] W.R. Heller, W.F. Mikhail and W. Donath, "Prediction of Wiring Space Requirements For LSI." *Journal of Design Automation and Fault-Tolerant Computing. (1978) (pp. 117-144)*

[9] Abbas A. El Gamal, "Two-Dimensional Stochastic Model for Interconnections in Master Slice Integrated Circuits." *IEEE Trans. on Circuits and Systems. Feb. 1981, pp. 127-138.*

[10] Abbas A. El Gamal and Zahir A. Syed, "A New Statistical Model For Gate Array Routing." *Proceedings of the 20th Design Automation Conference (1983).*

[11] D.J. Brown and R.L. Rivest, "New Lower Bounds for Channel Width." *CMU Conference on VLSI Systems and Computations. (1981).*

[12] Lajos Takacs, *Combinatorial Methods in the Theory of Stochastic Processes. Ch. 1.* John Wiley & Sons, Inc. 1967.

[13] P. Flajolet, "Combinatorial Aspects of Continued Fractions." *Annals of Discrete Mathematics 32(1980) pp. 125 - 161.*

[14] John Y. Ngai, *The General Interconnect Problem of Integrated Circuits.* Master Thesis, Department of Computer Science, California Institute of Technology, 1984.

Table 1.	
number of each test cases = 100	
Utilization Factor	Success Percentage
0.05	99
0.10	99
0.15	97
0.20	94
0.25	94
0.30	92

Table 2.		
runs = 1000, channel length = 200 left = 0, center = 20, right = 0		
Tracks Available	Simulated Probability	Predicted Probability
5	0.000 %	0.001 %
6	0.000 %	0.032 %
7	0.200 %	0.352 %
8	1.900 %	2.104 %
9	7.900 %	7.845 %
10	18.400 %	20.297 %
11	37.000 %	39.370 %
12	57.600 %	60.882 %
13	77.600 %	79.200 %
14	90.600 %	91.117 %
15	96.000 %	97.040 %
16	98.900 %	99.260 %
17	99.600 %	99.869 %
18	99.900 %	99.986 %
19	100.000 %	99.999 %
20	100.000 %	100.000 %

Table 3.		
runs = 1000, channel length = 200 left = 0, center = 30, right = 0		
Tracks Available	Simulated Probability	Predicted Probability
8	0.000 %	0.000 %
9	0.100 %	0.005 %
10	0.100 %	0.044 %
11	0.300 %	0.274 %
12	1.000 %	1.194 %
13	2.600 %	3.884 %
14	9.100 %	9.873 %
15	19.400 %	20.365 %
16	31.800 %	35.174 %
17	47.200 %	52.316 %
18	64.000 %	68.795 %
19	78.300 %	82.059 %
20	89.000 %	91.036 %
21	95.000 %	96.149 %
22	97.700 %	98.595 %
23	99.300 %	99.571 %
24	99.900 %	99.892 %
25	100.000 %	99.978 %
26	100.000 %	99.997 %
27	100.000 %	100.000 %

Table 4.

runs = 1000, channel length = 200
left = 12, center = 0, right = 15

Tracks Available	Simulated Probability	Predicted Probability
15	25.300 %	25.000 %
16	51.800 %	51.471 %
17	73.300 %	73.039 %
18	87.200 %	87.229 %
19	94.800 %	94.892 %
20	98.000 %	98.297 %
21	99.500 %	99.536 %
22	99.900 %	99.899 %
23	100.000 %	99.983 %
24	100.000 %	99.998 %
25	100.000 %	100.000 %

Table 5.

runs = 1000, channel length = 200
left = 10, center = 25, right = 10

Tracks Available	Simulated Probability	Predicted Probability
14	0.000 %	0.001 %
15	0.000 %	0.005 %
16	0.000 %	0.029 %
17	0.200 %	0.136 %
18	0.500 %	0.507 %
19	1.500 %	1.534 %
20	4.300 %	3.877 %
21	7.900 %	8.370 %
22	14.900 %	15.722 %
23	25.600 %	26.122 %
24	37.300 %	38.970 %
25	51.300 %	52.937 %
26	64.200 %	66.382 %
27	77.200 %	77.893 %
28	86.100 %	86.686 %
29	91.200 %	92.689 %
30	95.400 %	96.355 %
31	98.200 %	98.357 %
32	99.400 %	99.334 %
33	99.700 %	99.758 %
34	100.000 %	99.922 %
35	100.000 %	99.978 %
36	100.000 %	99.994 %
37	100.000 %	99.999 %
38	100.000 %	100.000 %

Table 6.

runs = 1000, channel length = 200
left = 5, center = 20, right = 20

Tracks Available	Simulated Probability	Predicted Probability
19	0.000 %	0.000 %
20	0.500 %	0.355 %
21	2.000 %	1.635 %
22	5.600 %	4.804 %
23	11.400 %	10.905 %
24	19.200 %	20.531 %
25	32.200 %	33.351 %
26	45.900 %	48.029 %
27	61.900 %	62.653 %
28	74.900 %	75.438 %
29	84.900 %	85.298 %
30	91.700 %	92.031 %
31	96.100 %	96.106 %
32	97.600 %	98.293 %
33	99.500 %	99.332 %
34	99.700 %	99.768 %
35	99.900 %	99.929 %
36	100.000 %	99.981 %
37	100.000 %	99.996 %
38	100.000 %	99.999 %
39	100.000 %	100.000 %
40	100.000 %	100.000 %

Table 7.

runs = 1000, channel length = 200
left = 10, center = 25, right = 35

Tracks Available	Simulated Probability	Predicted Probability
34	0.000 %	0.000 %
35	2.000 %	1.289 %
36	5.800 %	4.192 %
37	11.300 %	9.336 %
38	18.200 %	17.052 %
39	29.900 %	27.187 %
40	39.900 %	39.061 %
41	50.300 %	51.622 %
42	62.500 %	63.711 %
43	73.800 %	74.359 %
44	82.500 %	82.975 %
45	89.000 %	89.399 %
46	93.200 %	93.820 %
47	96.500 %	96.633 %
48	98.300 %	98.288 %
49	99.100 %	99.189 %
50	99.800 %	99.643 %
51	100.000 %	99.854 %
52	100.000 %	99.945 %
53	100.000 %	99.981 %
54	100.000 %	99.994 %
55	100.000 %	99.998 %
56	100.000 %	100.000 %

Total number of signals : 10

Left Signals:

2 6 8

Top Pins in Order:

7 3 1 9 10
6 2 4 5

Bottom Pins in Order:

1 3 4 5 7
8 9 10

Table 8.		
runs = 1000, channel length = 200		
Tracks Available	Simulated Probability	Predicted Probability
5	0.000 %	0.000 %
6	5.400 %	6.047 %
7	49.900 %	52.324 %
8	100.000 %	99.996 %
9	100.000 %	100.000 %
10	100.000 %	100.000 %

Total number of signals : 20

Left Signals:

1 3 6 8 10
17 19

Top Pins in Order:

14 19 7 5 6
2 4 3 18 9
12 20 11 16 15

Bottom Pins in Order:

1 2 4 5 7
8 9 10 11 12
13 14 16 17 20

Right Signals:

13 15 18

Table 9.		
runs = 1000, channel length = 200		
Tracks Available	Simulated Probability	Predicted Probability
7	0.000 %	0.000 %
8	1.900 %	2.430 %
9	17.700 %	18.905 %
10	91.700 %	91.620 %
11	99.900 %	99.813 %
12	100.000 %	99.973 %
13	100.000 %	100.000 %

Total number of signals : 15

Left Signals:

1 11 13

Top Pins in Order:

5 4 3 10 7
8 13 6 10 14
9

Bottom Pins in Order:

7 9 15 2 1
6 8 11 12 14
5 12

Right Signals:

2 3 4 15

Table 10.		
runs = 1000, channel length = 200		
Tracks Available	Simulated Probability	Predicted Probability
9	0.000 %	0.000 %
10	4.200 %	6.275 %
11	73.300 %	74.956 %
12	96.800 %	97.335 %
13	100.000 %	100.000 %

Total number of signals : 10

Top Pins in Order:

1 2 3 4 5
6 7 8 9 10

Bottom Pins in Order:

1 2 3 4 5
6 7 8 9 10

Table 11.		
runs = 1000, channel length = 200		
Tracks Available	Simulated Probability	Predicted Probability
1	0.900 %	0.554 %
2	24.300 %	21.307 %
3	60.900 %	58.248 %
4	85.500 %	83.218 %
5	94.700 %	94.755 %
6	98.800 %	98.766 %
7	99.800 %	99.794 %
8	100.000 %	99.978 %
9	100.000 %	99.999 %
10	100.000 %	100.000 %

Total number of signals : 10

Top Pins in Order:

1	2	3	4	5
6	7	8	9	10

Bottom Pins in Order:

10	9	8	7	6
5	4	3	2	1

Table 12.

runs = 1000, channel length = 200

Tracks Available	Simulated Probability	Predicted Probability
8	0.000 %	0.000 %
9	3.700 %	0.000 %
10	100.000 %	100.000 %