Applied Mathematical Sciences, Vol. 9, 2015, no. 14, 653 - 663
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ams.2015.411911

Obstacle-Aware Routing Problem in a Rectangular Mesh Network

Noraziah Adzhar

Department of Mathematical Sciences, Faculty of Science
Universiti Teknologi Malaysia
81310 UTM Johor Bahru, Johor, Malaysia

Shaharuddin Salleh

Centre for Industrial and Applied Mathematics
Universiti Teknologi Malaysia
81310 UTM Johor Bahru, Johor, Malaysia

Abstract

In an automatic design process for very large scale integration/printed circuit board (VLSI/PCB), routing problem is one of the most important and crucial part after components placement phase. This process is notoriously difficult, and a problem consists of a set of two-pin nets, is known to be NP-complete. In this problem, we considered a rectangular mesh network with all the location for blocks with pins on the boundaries and obstacles is predefined. Given a set of two-pin nets, our goal is to establish connections in the network by avoiding all the obstacles while satisfying the design rules. In this paper, we are using simulated annealing routing method with Dijkstra's shortest path algorithm to provide the best path for each connection if it exists. The simulation results showed that by accepting some uphill movements based on Boltzmann probability function led to better results and yield to lower energy level. This suggests the technique is suitable for adoption into real problem.

Keywords: PCB's routing, Rectangular mesh network, Shortest path algorithm, Simulated annealing method

1 Introduction

Routing in a PCB or VLSI design is the process of determining and prescribing paths between various electronic components in order to establish the connection between a given source node and its target. Routing in a modern chip is a notoriously difficult problem, and even the simplest routing problem, which consists of a set of two-terminal nets, is known to be NP-complete [1]. Since routing problem is very complex, this problem is usually split into two parts which is known as global routing and detailed routing in order to make it manageable. Global routing will tessellated the routing layout into an array of rectangular mesh and each mesh have variously referred to as grid cells, global routing cells, global routing tiles or bin. This process is as illustrated in **Figure 1**.

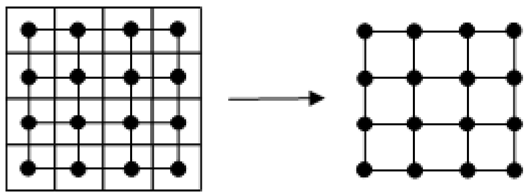


Figure 1: Partitioned layout into $N \times N$ rectangular array

Several methods has been proposed in the literature for two-terminal routing problem such as wave propagation method by Lee's[2], line-search routing[3-4], A* search routing[5] and shortest path approach. Several improvements have been made to Lees's algorithm in terms of coding scheme [6-7], search space [8] and search algorithm [9]. Past research often used heuristics method such as simulated annealing to produce a set of sequence-pair in the global routing problem [10]. However, the routing area is not utilized with the placement of obstacles. Obstacles in the layout may represent blocks that are placed on the surface, some nets that are already laid out, active element and others. The presence of obstacles will complicate the problem. It limits number of communication links in the region; therefore the best path for a respective net might not be short. Moreover, each successful connection will block later paths. The problem will become more complex since our objective is to have the maximum possible number of paths in a routing region.

In this paper, we are solving routing problem in VLSI/PCB design with the presence of blocks with active pins on the boundaries which represent the obstacles in our problem. Shortest path method is used to generate the best path in order to meet our objectives requirement. Simulated annealing technique which is based on cooling theory in solid is used in solution refining procedure to provide better results.

The rest of the paper is organized as follows. Section 2 is the problem background. In Section 3, the routing model is discussed and in Section 4, simulated annealing routing algorithm was presented. Section 5 presents the simulation part under random data, and the performance is evaluated. Finally, Section 6 concludes the paper.

2 Problem Statement

In this problem, our routing region was assumed to be tessellated into $p \times p$ square cells, and each cell was assumed contained exactly one pin. With regard to wiring, we assumed that the locations of blocks with pins on the boundaries were specified. Supposed we were given a set of routing requirements, which consisted of two-terminal nets, $N = \{N_1, N_2, ..., N_n\}$ where $N_i = (S_i, T_i)$ with S and T as the source and target nodes, respectively, the ultimate goal of this routing problem was to maximize the number of nets routed and minimize total wire length. In order to do this, we first sought the maximum number of interconnecting structure for a set of nets, N for each layer(s) while minimizing the level of congestion throughout the region. The objective function for this problem was defined as

$$\text{Max} \quad R = \sum_{i=1}^{m!} \sum_{j=1}^{m} q_{ij} a_{ij},$$

where $q_{ij} = \begin{cases} 1 & \text{non blocking} \\ 0 & \text{blocking} \end{cases}$, $a_{ij} = m \times m$ representing matrix of order by pair,
$m =$ total number of nets,
$i =$ order, and $j =$ nets. (1)

A complete wiring on this SD-Torus network should obey the design rules and satisfy the necessary conditions as follows:

(i) $N_i \cap N_j = \varnothing, i \neq j$

(ii) For each N_i, there must be exactly one connection only.

(iii) Each path can cross but should not overlap each other.

(iv) Each connection will be made using the communication links with no specific direction. This allows for a simpler representation of the routing configuration, even though it reduces freedom during routing.

3 Routing Model

As a result of its importance to the industry, we were motivated to produce a significant method to perform optimal routing. **Figure 2** illustrates the conceptual structure of a 9×9 rectangular mesh model with the locations of obstacles is predefined. For clarity, the illustration for all processing nodes is omitted.

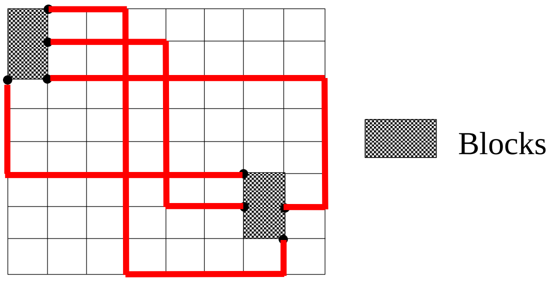


Figure 2: Routing in layout of size 9×9 with the presence of obstacles

The major features of this grid model are as follows:

- We define $p \times p = p^2$ as the grid size. For a 4×4 rectangular grid, we designated an ID for each node in the first row as 1, 2, 3 and 4 respectively with the upper left corner as 1 and the upper right corner as 4. Similarly, the ID for each node in the last row was 13, 14, 15 and 16 respectively with the lower left corner as 13 and the lower right corner as 16.
- The position of blocks with active pins on the boundaries is predefined. Since, the blocks itself will become an obstacle, therefore the communication links in the entire blocks is become passive and each links is assigned as 99 in our program.
- Other active edges held a value of 1 as its weight.
- Each connection is independent of other connections.
- Each pin will belong to only one net.
- Each pin could communicate with other pins in the left, right, upward or downward direction as long as the path did not overlap.
- Only one pin could log data in a given sub-bus at a time. This operation was performed sequentially and depended highly on the net ordering. Therefore, earlier connection would block later paths.

3.1 Simulated Annealing Routing Algorithm

Routing algorithm will manage network traffic and determine the best route for to connect source node and destination node. Once a path is established, the path it takes will be blocked and cannot be used to route other pairs of nodes. This will increase the congestion inside the routing region and makes later route longer than optimal and sometimes impossible to complete, thus, increase the energy level throughout the network as well as the cost (mainly refers to link numbers).

A shortest path algorithm is crucial for this problem. It was important to have all nodes connected in the shortest way to reduce energy level and network cost. In addition, we could provide larger routing space to route the remaining nets, thus maximizing the number of successive nets in the layer. In this paper, we are using Dijkstra's algorithm as a tool to provide best path since all the weights in the graph is positive values.

Current sequential routing often applies heuristic methods to further refine the solution. Through these processes, the connections for some nets will be swapped and re-routed in a different order to improve the routing quality. In this research, we applied a probabilistic method simulated by Metropolis et. al. and Kirkpatrick et al. [11-12] called simulated annealing to produce better sequence. This algorithm has become a very useful tool in solving various combinatorial optimization problems.

This method was based on the theory of annealing in solids. The term simulated annealing was derived from the roughly analogous physical process of heating and then slowly cooled a substance to obtain a strong crystalline structure [11]. During the simulation, temperature was lowered gradually until the system 'froze' and no further change occurred. At each temperature, the simulation must proceed long enough for the system to reach a steady state or thermal equilibrium. This method avoids being trapped in local minima by accepting uphill movement sometimes. The acceptance was determined using Boltzmann probability function:

$$P(\Delta E) = e^{\frac{-\Delta E}{T_i}}$$

where ΔE is the cost difference between the current and previous solutions, while T_i is the current temperature. For a given annealing schedule of temperature, $T = \{t_1, t_2, ...\}$, our implemented simulated annealing algorithm are as follows:

1. Determine an initial sequence for all to be routed nets, called L_0. Set $L = L_0$.
2. Then Dijkstra's algorithm is applied to compute shortest path for each nets. For each S_i, assign to every node a tentative distance value: set it to zero for our initial node and to infinity for all other nodes.
3. Mark all nodes unvisited. Set the initial node as current. For the current node, consider all of its unvisited neighbors and calculate their tentative distances. Compare the newly calculated tentative distance to the current assigned value and assign the smaller one.
4. When we are done considering all of the neighbors of the current node, mark the current node as visited. A visited node will never be checked again.
5. If T_i has been marked visited, then compute $d(S_i, T_i)$. If T_i is not reached, therefore a blockage has occur. Abandon the net and continue with the next net in L. Repeat the process for every nets in L.
6. Compute initial energy, E_0 using equation:

$$E = \sum_{j=1}^{m} q_{ij} d_{ij}, \text{ for } i = 1, 2, 3, ..., m!$$

Where i is the sequence (i can be referred as number of iterations too). Compute the number of successful routed nets, R_i. Mark that sequence as 'accept'.

7. From L, generate new sequence by swapping any two different elements randomly.
8. Set $L = L'$ and repeat Steps 2-5. Evaluate the new energy for the new sequence, $E(L')$ and the new $R(L')$.

9. If $R(L') > R(L_0)$, proceed to Step 13.
10. If $R(L') < R(L_0)$, reject the sequence and repeat Step 7.
11. If $R(L') = R(L_0)$, compute the energy change, $E(L')$. If $E(L) - E(L') < 0$, proceed to Step 13. Otherwise, go to Step 12.
12. Apply Boltzmann distribution formula. If $P(\Delta E) = e^{\frac{-\Delta E}{T_i}} > \varepsilon$ where $\varepsilon \sim U(0,1)$, go to Step 13. Otherwise, reject the move and repeat Step 7.
13. Accept the candidate sequence as a current solution, and set $L = L', E(L) = E(L')$ and $T = T_k$. Using reducing parameter, $\alpha = 0.95$, update the temperature counters and parameters. Set $k = k+1, T_{k+1} = \alpha T_k$ and repeat Step 7. Stop until no further changes occur.

4 Simulation Results

For obstacles-aware routing problem in rectangular pattern, our proposed algorithm has been implemented by using Microsoft Visual C++ 2010 [13] and run on a Intel Core2 Duo CPU 2.00GHz machine with 3GB Memory. Instead of maximizing number of routed nets in a layer, decrease the congestion level in the layer is also crucial. Thus, the overhead cost will be lower and the performance is improved. The energy level that has been using in this paper is derived from our objective function in (1) by simply replaced a_{ij} with d_{ij} becomes:

$$E = \sum_{j=1}^{m} q_{ij} d_{ij}, \text{ for } i = 1, 2, 3, ..., m!$$

where d_{ij} is number of communication links taken to get T_i from S_i. d_{ij} can be formulated as:

$$d(i, j) = d(i,k) + d(k, j).$$

Suppose the given location of blocks with pins for $p = 7$ is as illustrated in **Figure 3** and the net requirement is as follows:

$$N_1 = (23, 27),\ N_2 = (10, 34),\ N_3 = (17, 41),\ N_4 = (16, 42),\ N_5 = (20, 24).$$

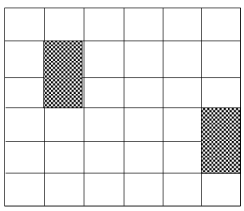


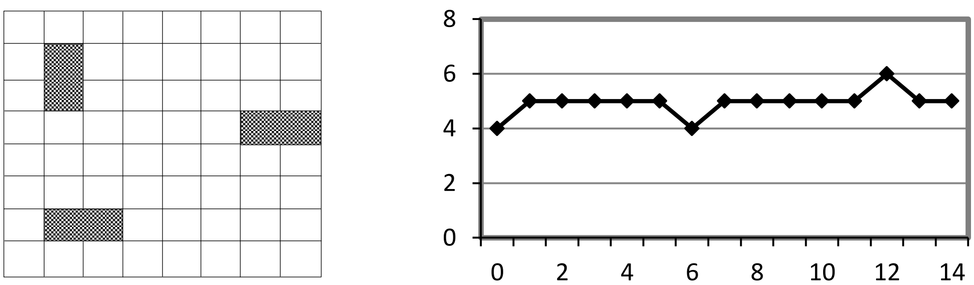
Figure 3: Placement of obstacle in 7×7 mesh model

The process of calculating maximum number of routed nets, R and energy level, E in this OAP model can be summarized as in **Table 1**. At the second iteration, N_3 is swapped with N_5 and producing the sequence N_1, N_2, N_5, N_4, N_3. By using this sequence, the number of blocked nets is reduced, thus R is increased even though E is increasing too. However, as the process continues, the value of E will gradually lowered into an optimal and acceptable result.

Table 1: Calculation of R and E in $p = 4$ routing region.

i	j	N_i	q_{ij}	d_{ij}	R	E
1	1	N_1	1	$d(23,27) = 6$	3	26
	2	N_2	1	$d(10,34) = 8$		
	3	N_3	1	$d(17,41) = 12$		
	4	N_4	0	$d(16,42) = 0$ *Path is blocked		
	5	N_5	0	$d(20,24) = 0$ *Path is blocked		
2	1	N_1	1	$d(23,27) = 6$	4	30
	2	N_2	1	$d(10,34) = 8$		
	3	N_5	1	$d(20,24) = 4$		
	4	N_4	1	$d(16,42) = 12$		
	5	N_3	0	$d(17,41) = 0$ *Path is blocked		

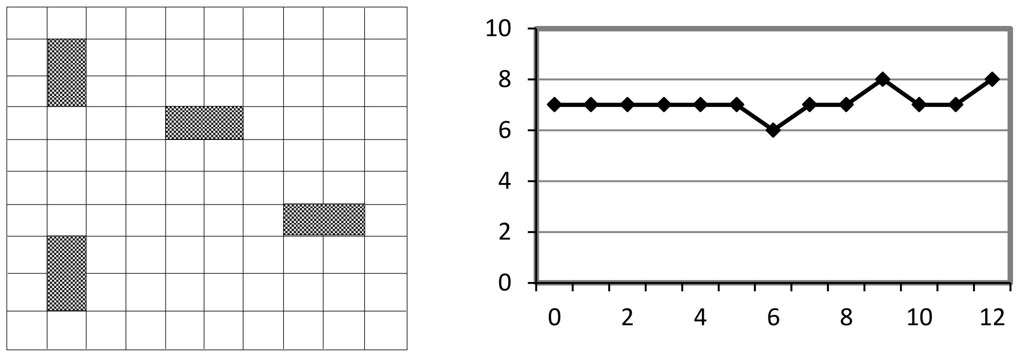
Notice that E reflects to the total length of wire needed to perform wiring and the complexity/congestion level in that routing region. Then, a simulation is performed on 9×9 rectangular array using our program called OAP. Three blocks of obstacles is placed inside the region which takes about 15% of the layout with $N = 7$ as shown in **Figure 4(a)** and the result is summarized in **Figure 4(b)**.



(a) **(b)**

Figure 4: (a) Placement of obstacles **(b)** Number of connections versus iterations no.

The graph in **Figure 4(b)** showed that our proposed algorithm able to route up to 6 nets out of 7 with $E = 61$ compare to previous best result of $R = 5$ and $E = 51$. This increment in energy level is due to the extra number of nets to be routed. However, as the program continues, the energy level will gradually decrease to an optimal and acceptable result. The program is then further tested on larger size of MESH with $p = 11$, 18 pins and $N = 9$ as illustrated in **Figure 5(a)**.



(a) **(b)**

Figure 5: (a) Placement of obstacles **(b)** Number of connections versus iterations no.

From the results in **Figure 5(b)**, the algorithm again succeeds in producing lower number of blocked nets. In this case, only one net has appeared to be blocked for connection. This problem can be easily overcome by introducing new layer to the region but is not discussed in this paper.

No swapping preference scheme is apply specifically in our proposed algorithm. However, for original MESH problem without obstacles, we notice that route shorter nets first often yield to better result. Since shorter nets will be routed in lowest cost too, therefore, more routing space is left for later nets yield to high number of R. However, it is not necessary to be true when applied into MESH with obstacles problem. This is due to the fact that the shortest way to route shorter nets might be long due to the presence of obstacles. For this problem, it has been noticed that routing longer nets first often yield to better results in comparison to shorter nets preference routing method. This is due There is no specific scheme for sequence-based routing method, and the task to find such of net ordering has proven to be NP-hard [14].

Finally, another simulation was performed with varied obstacles distributions to emphasize the performance of this method. In this simulation, the placement of obstacles is increased gradually by 10% over $p = 12$ network size containing 144 processing nodes and value of R and E is recorded. In Case 1, 2 blocks of obstacles was first introduced with N=4 and the result is recorded. Then, an extra block is placed in the layout with another two extra nets added in each cases. The result is as shown in **Figure 6(a)**. As the number of obstacles and number of nets to be routed increases, our program still able to perform complete routing until Case 3, but in Case 4 with 5 blocks and 10 nets, the maximum number of R achieved was 8 with $E = 101$. In this case, the routing layout become more congested but yet still produced a high number of R. Meanwhile, the congestion level for all cases is also kept minimum as possible. This is as illustrated in **Figure 6(b)**. Therefore, our proposed algorithm is proved to produce an optimal and acceptable results for most cases.

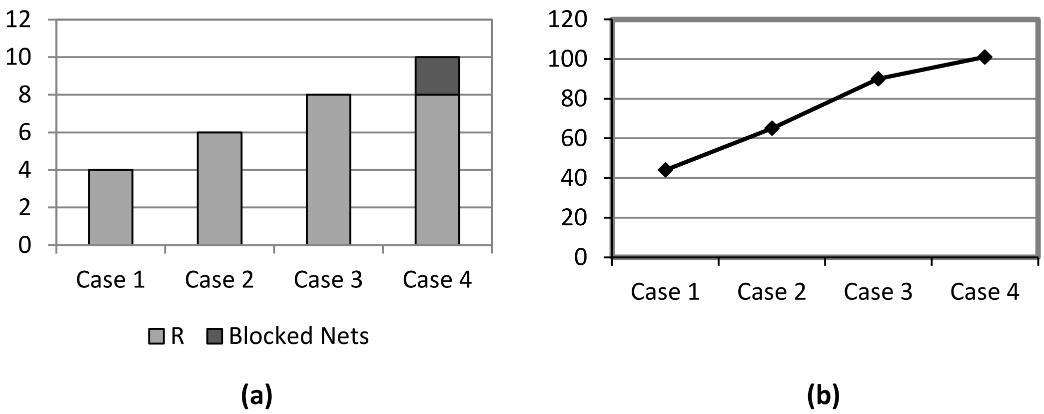


Figure 6: (a) Maximization number of R among four cases with increasing obstacles placement and net requirement **(b)** Minimization of E for all cases.

The whole result is tabulated in Table 2. Our proposed algorithm manages to produce high number of connections in various network sizes with less congestion level throughout the region. Therefore, it suggests that this method is suitable for adoption into real problem.

Table 2: The whole result.

Net. Size	#Nets	Obs.	$\#R$	$\%R$	E	% Con.
7x7	5	14/84	4	80	28	33.33
9x9	7	21/144	6	85.71	61	42.36
11x11	9	28/220	8	88.89	96	43.63
12x12	4	14/264	4	100	44	16.67
	6	21/264	6	100	65	24.62
	8	28/264	8	100	90	34.09
	10	35/264	8	80	101	38.26

*__Net.__=Network, **Obs.**=Obstacles, **Con.**=Congestion.

5 Summary

An industrial PCB can have more than 7000 nets with 12 layers. This routing problem was previously routed manually and with the advances of technology, it can be solved in less than a second for small size network. This paper described an obstacles-aware routing problem. This routing graph can be modeled as $G(N,E)$, where graph N represents a set of processing nodes, and the element of graph E is the link of communication between nodes. Supposed we were given a set of routing requirements, which consists of pairs of processing nodes (henceforth regard as nets), $N=\{N_1,N_2,...,N_n\}$ where $N_1=(S_1,T_1),N_2=(S_2,T_2),...,N_n=(S_{n,}T_n)$ with S and T as the source and target nodes, respectively. In this paper, we proposed a method to maximize the number of connections in grid consisting of mesh-connected processing nodes with number of nets and the location of blocks with active pins on the boundaries is predefined. Each routed nets will become obstacles itself to other nets and this increase the complexity to route later paths. Here, we used annealing technique to further refine the sequence and produce better result. It avoids being trap at local minimum by accepting some uphill movements based on the Boltzmann distribution formula. The simulation results proved that, this method is able to produce considerably high number of routed nets even the placement of obstacles is introduced. Nevertheless, the energy level is kept minimized while maximizing number of R. Therefore, the congestion level inside the routing region is lower. This suggests that out proposed method is suitable for adoption into real problem.

Acknowledgments. The author would like to thank the Ministry of Higher Education of Malaysia for the scholarship given.

References

[1] L. T. Wang, Y. W. Chang, and K. T. Cheng, *Electronic Design Automation: Synthesis, Verification, and Test*, Boston: Elsevier, 2009.

[2] C. Y. Lee, An algorithm for path connections and its applications, *IRE Trans Electronic Computers*, **EC-10** (1961), 346-365. http://dx.doi.org/10.1109/tec.1961.5219222

[3] K. Mikami and K. Tabuchi, A computer program for optimal routing of printed circuit connectors, in Proc. Int. Federation for Information Processing, November 1968, 1475–1478.

[4] D. Hightower, A solution to line routing problems on the continuous plane, in Proc. ACM/IEEE Design Automation Conf., June 1969, 1–24. http://dx.doi.org/10.1145/800260.809014

[5] P. E. Hart, N. J. Nilsson, and B. Raphael, A formal basis for the heuristic determination of minimum cost paths, *IEEE Trans. on Systems Science and Cybernetics*, **4** (1968), 100-107. http://dx.doi.org/10.1109/tssc.1968.300136

[6] S. B. Akers, A modification of Lee's path connection algorithm," *IEEE Trans Electronic Computers*, **EC-16** (1967), 97-87.
http://dx.doi.org/10.1109/pgec.1967.264620

[7] F. O. Hadlocks, A shortest path algorithm for grid graphs", *Networks*, **7** (1977), 323-334. http://dx.doi.org/10.1002/net.3230070404

[8] S. M. Sait, and H. Youssef, *Iterative computer algorithms: and their applications in engineering*, IEEE Computer Society Press, 1999.

[9] J. Soukup, "Fast Maze Router", Proc. DAC, (1978), 100-102.
http://dx.doi.org/10.1109/dac.1978.1585154

[10] M. P. Vecchi, and S. Kirkpatick, Global Wiring by Simulated Annealing", IEEE Trans. On Computer-Aided Design, **CAD-2** (1983).
http://dx.doi.org/10.1109/tcad.1983.1270039

[11] N. Metropolis, A.W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys*. **21** (1953), 498. http://dx.doi.org/10.1063/1.1699114

[12] S. Kirkpatrick, C.D. Gellat, M. P. Vecchi, Optimization by Simulated Annealing. *Science*, **220** (1983), 671-680.
http://dx.doi.org/10.1126/science.220.4598.671

[13] S. Salleh, A. Y. Zomaya and A. B. Sakhinah, *Computing for Numerical Methods Using Visual C++*, Wiley, Hoboken, NJ, 2008.
http://dx.doi.org/10.1002/9780470192634

[14] L.C. Abel, On the ordering of connections for automatic wire routing. *IEEE Trans. On Computers*, **21** (1972), 1227–1233.
http://dx.doi.org/10.1109/t-c.1972.223482

Received: November 21, 2014; Published: January 19, 2015