Routing problem in rectangular mesh network using shortest path based Greedy method

Routing problem in rectangular mesh network using shortest path based Greedy method

Noraziah Adzhar[1], Shaharuddin Salleh[2], Yuhani Yusof[1] and Muhammad Azrin Ahmad[1]

[1] Applied & Industrial Mathematics (AIMs) Research Cluster, Faculty of Industrial Sciences & Technology, Universiti Malaysia Pahang, 26300 Gambang, Kuantan, Pahang, Malaysia.
[2] Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia (UTM), 81310 Johor Bahru, Johor, Malaysia

noraziahadzhar@ump.edu.my

Abstract. A supercomputer can have thousands of processor-memory pairs which often referred as processing pins. Each of these pins is connected to each other through networks and passes message using a standard message passing mechanism such as Message Passing Interface. In this research, we consider the routing problem in rectangular mesh network. Each terminal pin in the network needs to be connected with its destination pin for it to function properly. Thus, maximizing the number of connection for each pair of pins and keeping the total energy throughout the network minimum becomes our main objective. In order to achieve this objective, each net need to be routed as shortest as possible. Therefore, developing a shortest path based routing algorithm is in need. In this research, Dijkstra's algorithm is used to establish the shortest connection for each net. While this method guarantees to provide the shortest connection for each single net (if exists), however each routed net will become the obstacles and block later connections. This will add complexities to route later nets and make its routing longer than optimal or sometimes impossible to complete. Therefore, the routing sequence need to be rip-up and all nets need to be re-routed. This paper presents a complete routing algorithm which can further refine the solution by using Dijkstra's based greedy method. The outcomes from this research is expected to benefit engineers from electric & electronic industry.

1. Introduction

Very Large Scale Integration is a process of integrating hundreds of thousands on a single microchip. One of the most important step in building this integrated circuit is determining the optimal route for each pair of pins on the circuit (sometimes called terminal or modules) by satisfying the design rules. An optimal route (in terms of minimum cost, shortest path, lowest running time) for each connection increase the chip performance and reducing energy level.

Researchers have shown much interest on this problem due to its pervasive applications and is being studied extensively [1]. Few improvements have been reported in the literature regarding providing deadlock-free routing scheme [2], routing platform associated with and without obstacles [3-4], and finding longest path with the presence of obstacles in a rectangular routing grids [5].

The rest of this paper is organize as follows: Section II describes the routing problem in rectangular mesh layout and present the routing algorithm Dijkstra's based Greedy method routing algorithm. Section III presents simulation results and comparison with our previous proposed routing algorithm [6] while section IV concludes this paper.

2. Routing in rectangular mesh network

Network can be defined as a group of two or more interconnecting components. A network is often converted into its equivalent graph to provide a better explanation graphically and to gain actionable insights. Through graph the properties of the network can be studied effectively.

In this study, we consider a rectangular mesh network where the vertices of the graph represent pins and the link between vertices represent the communication links. The graphical illustration of this network is as in Figure 1.

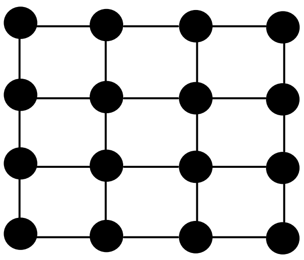

Figure 1. 4x4 rectangular mesh network model

Each source component, S (henceforth regards as pin) in the network need to be connected to its destination or target pin, T. For a 4x4 rectangular mesh network model, there are a maximum of 8 pair of source-target pin (henceforth regard as net, N). The main objective in this routing problem is to provide the optimal connection for all nets by satisfying the design rules and necessary conditions as follows:

(i) A pin should belong to only one net and either be a source pin or target pin. Mathematically, $N_i \cap N_j = \varnothing, i \neq j$.
(ii) Each net should have exactly one route only from its source pin to target pin.
(iii) Each route should not overlap with another route but can cross.
(iv) Each route will be based on grid routing and follow the specified communication links without specific direction.

In order to do this, each net must be connected in the shortest way as possible. However, previous route will become a new obstacle to upcoming path. This cause later path will be more complex and make its routing longer than optimal or sometimes impossible to complete. We develop a parameter q_{ij} to identify whether a path for a specific net is blocked or not blocked. A path will have $q_{ij} = 1$ if and only if the path taken to route S_j and T_j of the respective N_j is not blocked by other paths. While $q_{ij} = 0$ if the path of that respective N_j is blocked or not exist.

The objective function for this problem was defined as:

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

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

Another objective in this study besides maximizing the number of routed nets is to minimize the total energy taken to route all nets. From Eq. 1, the term a_{ij} is simply replaced with d_{ij}, where d_{ij} is the total distance for routing N_j. Therefore, the energy function for this problem is:

$$E = \sum_{j=1}^{m} q_{ij}d_{ij}, \qquad (2)$$

for $i = 1, 2, 3, ..., m!$

2.1. Single source shortest path problem

Applications that are dealing with shortest path problem are geographical map, telecommunication network, transportations and many more. Single source shortest path problem indicates a problem to determine the shortest path from the source point to its target point.

Lee’s algorithm is one of commonly used routing algorithm. This algorithm works in the ‘wave propagation manner’. In the process of determining shortest path, first of all, the source pin will send message to its four neighbors. The message then propagates in the form of wave and move towards another pin. The first wave front reaching the target pin completes the connection. This algorithm guarantees to provide a shortest connection (if exists). However, this algorithm has a worst case running time and consume a lot of memory. Due to these weaknesses, a lot of improvements on this method has been reported in the literature in terms of coding scheme [7-8], search space [9] and search algorithm [10].

Few other common shortest path methods are Bellman-Ford algorithm, Floyd-Washall Algorithm and Dijkstra’s algorithm. Floyd-Washall algorithm is not suitable to be used for a single source shortest path problem. The weight for each link in our weighted graph that represent distance does not consist a negative value. Thus, Bellman-Ford algorithm also will not be suitable. Therefore, only Dijkstra’s algorithm [11] is suitable to be use in our study and we would like to propose a Dijkstra’s shortest path based routing algorithm. The classical Dijkstra’s algorithm is powerful in its class since it guarantees to determine the shortest path for a source-target point if exists and also use a smaller the memory usage. The summary of these methods is simplified in the following Figure 2.

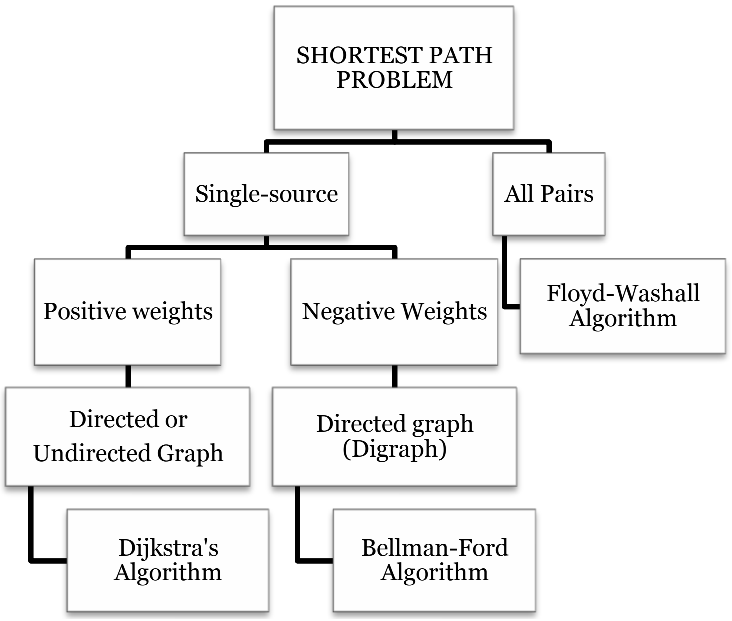

Figure 2. Several popular approaches to single-source and all pairs shortest path problem.

2.2. Greedy method
While Dijkstra's algorithm guarantees the shortest path for each net, the previous route will become a new obstacle to upcoming path. This cause later path to be more complex and make its routing longer than optimal or sometimes impossible to complete. Therefore, we need a method to further refine the sequence for routing.

A greedy algorithm is one of the methods that is often used to solve optimization problems. This method is easy to implement and require less computing resources. The execution time is also shorter since the algorithm is not really complicated. Greedy algorithms make choices that look the best at that every moment without regard for future consequences. Several classical optimization problems such as minimum spanning tree yield global optimum solutions using greedy algorithm.

With regard to routing problem, greedy algorithm never gives a second thought on worse cases and always accept a good move. That is why it is called 'greedy' and faster to execute. This differs with simulated annealing technique which put worse cases under a consideration through a probability function. In this research, we would like to compare the efficiency of these two methods on our routing problem. The Dijkstra's based greedy algorithm for routing in rectangular mesh network is presented as follows:

Dijkstra's Based Greedy Method Routing Algorithm

1. Determine an initial sequence for all nets to be routed, called P_0. Set $P = P_0$.
2. Apply Dijkstra's algorithm to compute the shortest path for each nets. For each source point, S_i a tentative distance value is assigned to every node; set it to zero for our initial node and infinity for all other pins.
3. Mark all pins 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. After finish 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 P. Repeat the process for every nets in P.
6. Compute the number of successful routed nets, R_0 using Eq 1. Compute initial energy, E_0 using Eq 2. Mark that sequence as 'accept'.
7. From P_0, generate new sequence by swapping any two different elements randomly.
8. Set $P = P_{i+1}$ and repeat Steps 2-5. Evaluate the new number of routed nets, $R(P_{i+1})$ and the new energy $E(P_{i+1})$ of the new sequence.
9. If $R(P_{i+1}) > R(P_i)$, proceed to Step 12.
10. If $R(P_{i+1}) < R(P_i)$, reject the sequence and repeat Step 7.
11. If $R(P_{i+1}) = R(P_i)$, compute the energy change, $E(P_{i+1})$. If $E(P_{i+1}) - E(P_i) \le 0$ proceed to Step 12. Otherwise, reject the sequence and repeat Step 7.
12. Accept the candidate sequence as a current solution, set $P = P_{i+1}$, $R = R(P_{i+1})$ and $E = E(P_{i+1})$. Update $i = i+1$.. Repeat Step 7.
13. If no further improvement occurs, then, stop.

3. Results and discussion

A program has been built by using Microsoft Visual C++ 2010 and run on Intel Core2 Duo CPU 2.00GHz machine with 3GB Memory to simulate the algorithm. The algorithm has been tested for various data set and network sizes. The results are tabulated in the following Table 1.

Table 1. Simulation results for various data set and network sizes.

Network Size	**#Nets**	**Greedy Method**	
		$\# R$	E
7x7	5	4	28
9x9	7	6	61
11x11	9	8	94
12x12	4	4	44
	6	6	65
	8	8	90
	10	9	107

From the results, it can be noticed that our proposed routing algorithm able to maximize the number of routed nets and thus minimize the number of blocked nets. The total number of blocked nets is as summarized in Table 2.

Table 2. Number of blocked nets by using proposed routing algorithm.

Network Size	#Nets	Number of blocked nets
7x7	5	1
9x9	7	1
11x11	9	1
12x12	4	0
	6	0
	8	0
	10	1

We also compare the results given by Dijkstra's shortest path based Greedy Method (GM) with Dijkstra's shortest path based simulated Annealing Method [6]. The results are tabulated in the following Figure 3-4.

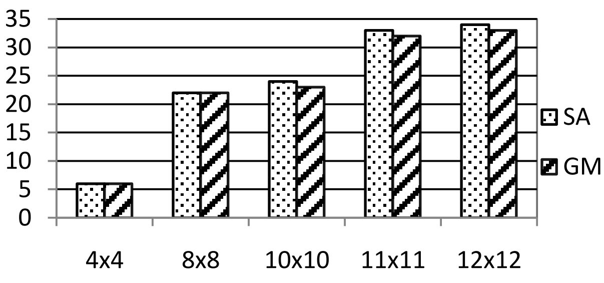

Figure 3. Maximization of *R* using greedymethod and simulated annealing.

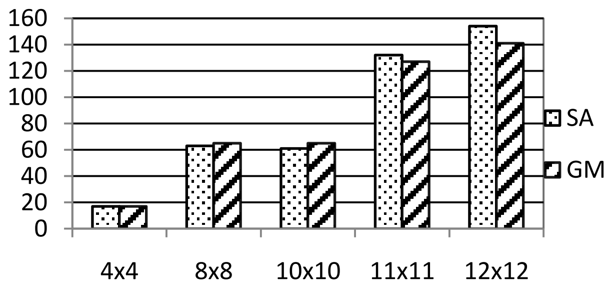

Figure 4. Minimization of *E* using greedy method and simulated annealing.

Graphs in Figure 3-4 show that for small size of networks greedy method able to produce the same near-optimal result with simulated annealing in shorter running time. This is due to the process of accepting some uphill moves in simulated annealing procedures to avoid getting trap in local minima. However, as the network size increases, greedy method was no longer able to produce a good result. Its performance starts to degrade neither in terms of number of *R* nor its energy level.

Even though simulated annealing needs a longer running time to avoid getting trap at local minima, the result it produces is far better than greedy method.

4. Summary

This research explains the basic routing model in the form of rectangular array that is commonly used in VLSI' routing which is rectangular mesh. From the simulation results, as the network size increases, greedy method cannot perform better in maximizing number of R and reduces the energy level compared to our previous proposed method [6]. The limitation in this optimization problem is we only consider a routing platform without any placement of blocks or inactive pins and this can be the suggestion for further research.

Acknowledgments
The authors would like to thank Universiti Malaysia Pahang for supporting part of this research under research grant RDU1703188.

References

[1] Daeho, S. Akif, A. Won-Taek L et al. 2005 Near-optimal worst-case throughput routing for two-dimensional mesh networks. *Technical Report Purdue University.*
[2] Duan, X. and Wu, J. 2013 Deadlock-free routing scheme for irregular mesh topology NoCs with oversized regions. *Journal of Computers*, 8(1), pp 27-32.
[3] Yan, J. T. and Chen, Z. W. 2011 Obstacle-aware length-matching bus-routing. *Proc. International Symposium on Physical Design (ISPD'11),*pp 61-67.
[4] Adzhar, N. and Salleh, S. 2015Obstacle-aware routing problem in a rectangular mesh network. *Applied Mathematical Sciences*, 9, p 14.
[5] Yan, J. T. Jhing, M. C. and Chen, Z. W. 2010 Obstacle-aware longest path using rectangular pattern detouring in routing grids. *Proc. Design Automation Conference (ASP DAC),* 287-292.
[6] Adzhar, N. and Salleh, S. 2014 Simulated annealing technique for routing in a rectangular mesh network. *Modelling and Simulation in Engineering.*
[7] Akers, S. B. 1967 A modification of Lee's path connection algorithm. *IEEE Trans Electronic Computers*, EC-16, pp 97-87.
[8] Hadlock, F. O. 1977 A shortest path algorithm for grid graphs. *Networks*, 7, pp 323-334.
[9] Sait, S. M. and Youssef, H. (1999). Iterative computer algorithms: and their applications in engineering. *IEEE Computer Society Press.*
[10] Soukup, J. 1978 Fast Maze Router. *Proc. DAC,* 100-102.
[11] Dijkstra, E. W. 1959 A note on two problems in connexion with graphs. *Numerishce Mathematik,* 1, pp 269-271.