5

DOUBLE SIMULATED ANNEALING MODEL FOR MAPPING OF GRAPHS TO SINGLE-ROW NETWORKS

Loh Ser Lee
Shaharuddin Salleh
Nor Haniza Sarmin

Connected graph G is a graph where every pair of distinct vertices in the graph is connected either directly or indirectly. In our real life, many science and engineering applications can be reduced into the connected graph representations, such as telecommunication and wireless sensor problems. A connected graph can be transformed into single row-routing network S as shown by Salleh *et. al.* in 2007. However, the earlier work does not produce optimal results in S as the ordering in G is sequentially done. In this paper we will present a complete optimizing modeling for single row transformation of connected graphs. Thus, the effort to optimize the ordering of vertices in connected graph for transformation into a planar graph in the form of single-row network is carried out through the new interval formation and Double Simulated Annealing (DSA).

5.1 OVERVIEW

Single row routing is a routing technique which has its root in the layout design of printed circuit boards (PCBs). The technique is the crucial backbone in the design on multilayer printed circuit boards in the complex electronic systems. An efficient single-row routing ensures the efficiency in information transferring and it minimizes the objective subject to the cost such as circuit board production. Thus, many researches had been carried out in finding the best modeling to solve the single-row routing problem in minimizing the congestion on the circuit board.

Single-row routing problem has been shown to be NP-complete problem. The optimum solution is not easy to be determined. For this reason, Kuh et al. had developed the necessary and sufficient condition for optimum single-row routing ([1] and [2]). Some heuristic algorithms were proposed to find the solution for single-row routing problem as in [3]. Bhattacharya et al. proposed a new approach based on graph theoretic representation in [4] to solve the single-row routing problem.

In [5], a model which minimizes both the congestion and number of doglegs was developed. The energy function is expressed as a function of one parameter by pivoting the other parameter. In other word, when total energy value is minimized, congestion and number of doglegs are minimized as well. In [6], the relationship between complete graph and its single-row representation was studied for the first time and followed by connected graph using Hopfield neural network to the Maximum Clique problem [7]. A model has been proposed to transform a graph into single-row routing problem and solved it using ESSR in [5].

In this work, we further the study on optimizing the transformation of connected graph into single-row routing before solving it using ESSR. The nodes labeling is permutated in searching the optimal sequence of zones and followed by optimal order of nets searching in its single-row representation. An interval formation model for connected graph is developed and a model called Double Simulated Annealing (DSA) will be presented for the optimal transformation and single-row routing problem solving.

The paper is organized into five sections. Section 5.1 is the introduction while Section 5.2 is the problem statement. Our model in transforming connected graph into single-row representation is presented in Section 5.3. Section 5.4 presents the experimental results, while Section 5.5 is the summary and conclusion.

5.2 PROBLEM STATEMENT

The problem can be defined as follows: Given a connected graph, it needs to be transformed into a planar graph as single-row representation. One example is shown in Figure 5.1. In a connected graph G_n, there are n nodes and each node, $v_i, i = 1, 2, ..., n$, has d_i degree(s). In the single-row transformation, every node v_i in G_n forms a zone, z_i in S_n for $i = 1, 2, ..., n$ and each z_i has a number of terminals equals with $d_i, i = 1, 2, ..., n$. The terminals aligned with evenly spaced on node axis.

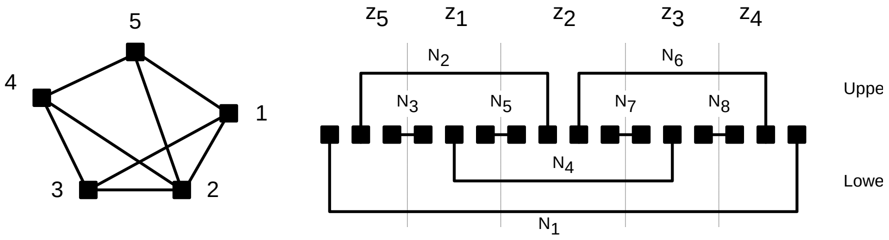


Figure 5.1. A connected graph G_5 and its single-row representation.

Nodes labeling in G_n is corresponded by sequence of zones in S_n. Since the sequence of zones is not sequentially done,

c_j, $j = 1, 2, ..., n$ is used in this paper to represent the order of zones' sequence in S_n. For example, from Figure 5.1 z_{c_1} is referring to z_5 while z_{c_2} is referring to z_1. The communication link between v_i and v_j in G_n are preserved and transformed into nets which connect z_i and z_j in S_n for $i = 1, 2, ..., n$ and $j = 1, 2, ..., n$. Each net is drawn from left to right in one way and they are realized in the form of non-crossing paths. Each path consists of horizontal and vertical line segments such that none of them cross each other.

In a single-row network, the number of horizontal tracks in upper street called the upper street congestion (Q_u) while the number of horizontal tracks in lower street called the lower street congestion (Q_l). The congestion for a network is defined as the maximum between the upper and lower street congestion or $Q = \max\{Q_u, Q_l\}$. Dogleg is the vertical segment of path which intersects the node axis.

In the transformation, the nodes labeling can affect the optimality of the result obtained in single-row routing problem. Thus, our main objective is to find the best nodes labeling which can lead to the optimal result with minimum congestion and number of doglegs in single-row routing problem. The objective function in the form of the Energy function in [5] is then applied to minimize both the congestion and number of doglegs.

5.3 OUR MODEL

5.3.1 FORMATION OF ZONES AND TERMINALS

The formation of zones represents the mapping from *n* vertices, v_i in G_n into *n* zones, z_i in S_n. Every z_i in S_n has a number of terminals equally with its number of degree, d_i, of vertex v_i in G_n.

The initial solution for the sequence of zones is sequentially done. Since nodes labeling can affect the optimality of the result obtained in single-row routing problem, the sequence of zones will be permutated using simulated annealing in order to obtain the order of zones' sequence, $c_j,\ j = 1, 2, ..., n$ which reflects the nodes labeling in G_n. In connected graph, total number of terminals formed from G_n to S_n is $\sum_{i=1}^{n} d_i$. All terminals are aligned on node axis and numbered successively follow the order of zones' sequence, $c_j,\ j = 1, 2, ..., n$. The method for the formation of zones and terminals is shown in Algorithm 5.1.

/**Algorithm 5.1:** *Formation of zones and terminals in graph* S_n *from the connected graph,* G_n

```
Given $c_j$ as the order of zones' sequence;
/* $z_{c_j}$ in $S_n$ corresponds to $v_{c_j}$ in $G_n$ for $j = 1, 2, ..., n$ */
Determine the degree, $d_{c_j}$ of every vertex, $v_{c_j}$ in $G_n$, for $j = 1, 2, ..., n$;
Set t = 0;
for $j = 1$ to $n$
    for $k = 1$ to $d_{c_j}$
            Set the terminal number, $t \leftarrow t + 1$;
            Set the zone, $z_{c_j k} = t$, in $S_n$ which corresponds to node $v_{c_j}$ in $G_n$;
```

5.3.2 FORMATION OF INTERVALS

Sum of degree is the total number of degrees for n vertices. Hence, the number of nets to be formed is $m = \frac{\text{sum of degree}}{2} = \frac{1}{2}\sum_{i=1}^{n} d_i$

nets. To avoid confusion, z_i is connected with z_j in S_n refers to nodes v_i and v_j in G_n is connected. There are two parts in nets construction.

In the first part, a net is formed on node axis between two consecutive terminals from two consecutive zones if the two consecutive zones in $c_j, j = 1, 2, ..., n$ are connected. This step produces the nets with zero energy.

For the second part, a $g = \{g_1, ..., g_n\}$ list is defined as the zones' order for the interval formation. Set the middle zone, z_{c_j} in S_n as the first element, $g_1 = j$ in g list as the middle zone will be the first zone where the nets start to be formed. The two zones which are next to the middle zone at left and right hand side are set to be the second and third elements respectively in g list. After that the two zones which are second next from middle zone at left and right hand side are set to be forth and fifth elements respectively in g list. The process is repeated until the list is completed.

For the zone $z_{c[g_1]}$, a net is formed between the first available terminal in it and the last available terminal in the zone at left hand side which is nearest and connected with $z_{c[g_1]}$. All the zones at left hand side of $z_{c[g_1]}$ will be checked one by one in direction from $z_{c[g_1]}$ to the left. The process is repeated until all the zones at left hand side which are connected with $z_{c[g_1]}$ each has a net formed with $z_{c[g_1]}$. Following by that, a net is formed between the last available terminal in $z_{c[g_1]}$ and the first available terminal in the zone at right hand side which is nearest and connected with $z_{c[g_1]}$. Then, all the zones at right hand side of $z_{c[g_1]}$ will be checked one by one in direction from $z_{c[g_1]}$ to the right. The process is repeated until all the zones at right hand side which are connected with $z_{c[g_1]}$ each has a net formed with $z_{c[g_1]}$.

After the first iteration, the zone $z_{c[g_1]}$ has $d_{c[g_1]}$ nets formed. From the connected graph G_5 shown in Figure 5.1, the single-row representation up to this moment is shown in Figure 5.2.

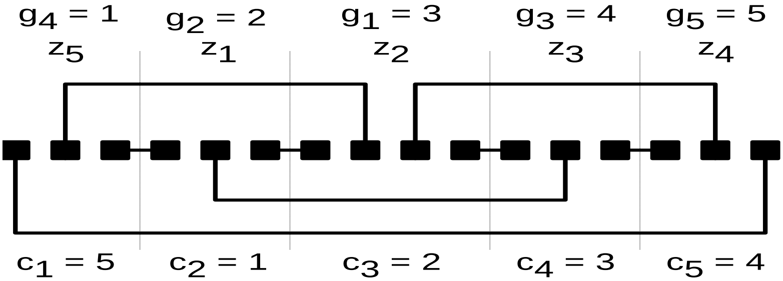


Figure 5.2. Single-row representation for G_5 after the first iteration of net construction for $z_{c[g_1]}$.

In Figure 5.2, it is clear that after first iteration, all terminals in zone $z_{c[g_1]}$ have paired with one terminal from each zone which is connected with $z_{c[g_1]}$ to form the nets. The iteration is repeated for zone $z_{c[g_2]}$ until the last zone which is zone $z_{c[g_n]}$. After n iterations, the single-row representation is said to be completed. The method of interval formation is outlined as in Algorithm 5.2.

Algorithm 5.2: *Formation of Intervals*

```
for $i = 1$ to $n - 1$
    if zone $z_{c_i}$ is connected with zone $z_{c_{i+1}}$
    A net is formed between 2 consecutive terminals from $z_{c_i}$ and $z_{c_{i+1}}$;
Define a $g = \{g_1, ..., g_n\}$ list as the order for interval formation of zones;
Set the middle zone in $S_n$ as the $1^{st}$ element in $g$;
Complete the $g$ list in the order of zones which nearest to the middle zone from
    left and right hand side alternately;

for $k = 1$ to $n$
    $i = c_{g_k}$;
    for $j = c_{g_k - 2}$ to $c_1$
    if $z_i$ is connected with $z_j$
                A net is formed between the last available terminal in $z_j$ and the
                    1st available terminal in $z_i$;
   for $j = c_{g_k + 2}$ to $c_n$
    if $z_i$ is connected with $z_j$
              A net is formed between the last available terminal in $z_i$ and the 1st
                  available terminal in $z_j$;
```

Once the construction of nets is done, the nets will be undergoing resorting and renumbering based on their beginning terminals in ascending order. After that, each net is assigned into a unique level which represents the nets ordering [6]. Finally, ESSR or *Enhanced Simulated Annealing for Single-row Routing* [5] is applied to obtain the optimal sequence for nets ordering which gives the lowest energy value.

5.3.2 DOUBLE SIMULATED ANNEALING (DSA)

Simulated Annealing has been proven for its efficiency in searching the optimal solution for combinatorial optimization problems. Thus it is used in searching for optimal nodes labeling and the optimal order of nets.

Double Simulated Annealing (DSA) is our proposed method of applying the simulated annealing twice: first on *G* to produce the optimal labeling of the nodes, then on *S* to produce the optimal routing. DSA is outlined in Algorithm 5.3.

Algorithm 5.3: *DSA Algorithm for Nodes Labeling and Net Ordering*

```
Set Cooling rate, $\alpha$ = 0.95;
Set initial temperature = 100;
Set an initial list, $c_0$;
Compute $E_0$ and $Q_0$;
for $k = 1$ to maximum iteration
    for $j = 1$ to maximum iteration
        Select one pair of zones and swap their order to form $c_r$;
        Formation of zones and intervals;
        Set initial order of nets;
        SA Algorithm for Net Ordering; //[6]
        Evaluate the new energy $E_r$ and congestion $Q_r$;
        if $Q_r \le Q_k$
            if $(\Delta E = E_r - E_k \le 0)$, or (if $\Delta E > 0$ and
              $\exp(-abs(\Delta E)/T_k) > \varepsilon$ )
                    Update $c_k \leftarrow c_r$ , $E_k \leftarrow E_r$ and $Q_k \leftarrow Q_r$;
                    if $Q_k < \min\_Q$ and $E_k < \min\_E$
                        Store $best\_c = c_k$;
                        Update $\min\_E \leftarrow E_k$ and $\min\_Q \leftarrow Q_k$;
                    endif;
                    break;
                else
                    nonimprovement++;
              else
                    nonimprovement++;
              if nonimprovement > 6
                    break;
        Update $T_{k+1} \leftarrow \alpha T_k$;
        if $T_{k+1} < \varepsilon$
            break;
```

5.4 EXPERIMENTAL RESULTS

We apply DSA on two cases of graphs, namely, a connected graph and its special case of complete graph. We also apply DSA on the case of binary graph for comparison. A complete graph is a graph where every two distinct vertices in the graph are adjacent. Table 5.1 shows the results comparison between different types of interval formation for complete graph.

Table 5.1. Comparison between Different Types of Interval Formation for Complete Graphs

C_n	*Previous work in [6]*			*DSA*		
	E	Q	D	E	Q	D
$n = 5$	11	3	1	11	3	1
$n = 6$	28	4	4	29	4	5
$n = 8$	128	9	21	121	8	21
$n = 10$	403	16	53	395	14	57

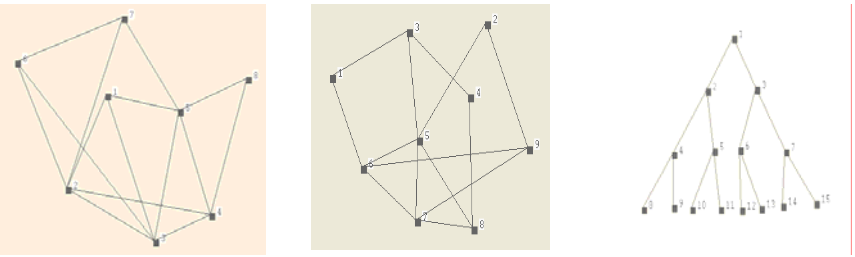
In the complete graph, the permutation of nodes labeling makes no difference to the final result since all the pairs of nodes are adjacent. Thus, by comparing with previous work in [6] which worked on single row transformation for complete graph, our new intervals formation model can also obtain good results.

However, in the connected graph where the complete graphs are excluded, the labeling of nodes in G_n will affect the optimality in the single-row transformation from G_n to S_n. Thus, DSA shows its ability in improving the results for connected graphs excluding complete graphs. Table 5.2 shows the results comparison between SSA and DSA for some connected graphs.

Table 5.2. Comparison of results between SSA and DSA

Graph	*Single SA*			*DSA*		
	E	Q	D	E	Q	D
Graph 1	24	4	4	13	3	1
Graph 2	23	3	4	9	2	1
Graph 3	46	5	8	6	2	0

Single SA in this report refers to the previous transformation work in [6] but with new formation of intervals algorithm (Algorithm 5.2). The transformation from G_n to S_n in Single SA is not optimal as the nodes labeling in G_n is sequentially done. Hence, DSA is developed to permute the nodes labeling. Some random generated connected graphs and a binary graph which are shown in Figure 5.3 are used as the experimental data to show the efficiency of DSA model compare with Single SA.



Graph 1 Graph 2 Graph 3

Figure 5.3. Some random generated connected graphs and a binary graph.

The permutation of nodes labeling tends to produce the set of nets which is shorter and if possible, with zero energy value.

Therefore, DSA gives better results, as shown in Figures 5.4 and 5.5.

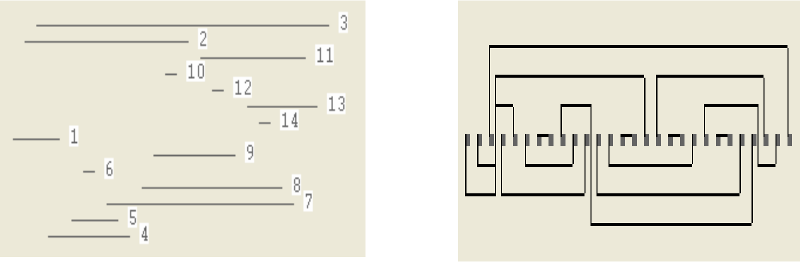


Figure 5.4. Nets ordering of Graph 2 (left) with minimum energy, *E* = 23 and its final realization (right) using Single SA algorithm.

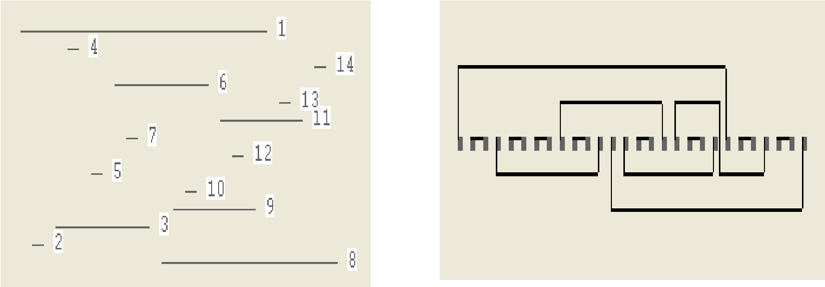


Figure 5.5. Nets ordering of Graph 2 (left) with minimum energy, *E* = 9 and it final realization (right) using DSA.

5.5 CONCLUSION

In this paper, we propose a new modeling for interval formation and DSA. The transformation of connected graph G_n into single-row representation S_n involves Algorithm 5.1 (formation of zones and terminals) and Algorithm 5.2 (formation of intervals). We then apply a new method called DSA (Algorithm 5.3) which optimizes the transformation from G_n to S_n and use the basic of simulated annealing [7] from earlier work called ESSR to obtain the result. Experimental results show that this new modeling is able to improve the results for single-row transformation of connected graphs. However, the connected graphs have various types with different patterns and characters, thus we will focus and study on the characters of different connected graphs to come out with a suitable modeling for that connected graph such as binary graph.

5.6 REFERENCES

[1] B. S. Ting, E. S. Kuh, L. Shirakawa. The multilayer routing problem: algorithms and necessary and sufficient conditions for the single row, single layer case. *IEEE Transactions Circuit and Systems*, 23:768-778, 1976.

[2] E. S. Kuh, T. Kashiwabara, and T. Fujisawa, On optimum single-row routing. *IEEE Transactions Circuit and Systems,* 26(6):361-368, 1979.

[3] T. T. Tarng, M. M. Sadowska, and E. S. Kuh. An efficient single-row algorithm. *IEEE Transactions on Computer-Aided Design*, 3(3):178-183, 1984.

[4] B. B. Bhattacharya, J. S. Deogun, and N. A. Sherwani. A graph theoretic approach to single row routing problems. *Proceedings of the 1988 IEEE International Symposium on Circuit and Systems*. June 7-9, 1988. 2:1437-1440, 1988

[5] S. Salleh, B. Sanugi, H. Jamaludin, S. Olariu, and A. Y. Zomaya. Enhanced simulated annealing technique for the single-row routing problem. *Journal of Supercomputing,* 21(3):285-302, 2002.

[6] S. Salleh, S. Olariu and B. Sanugi. Single-row transformation of complete graphs. *Journal of Supercomputing*, 31, 265-279, 2005.

[7] S. Salleh, S. Olariu, A. Y. Zomaya, L.Y. Kiew and N.A.B. Aziz. Single-row mapping and transformation of connected graphs. *Journal of Supercomputing*, 39:73-89, 2007.

[8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing. Science*, 220(4598):671-678,1983.*