The Journal of Supercomputing, 31, 265–279, 2005

Single-Row Transformation of Complete Graphs

SHAHARUDDIN SALLEH
Department of Mathematics, Faculty of Science, Universiti Teknologi Malaysia, 81310 Johor Bahru, Malaysia

STEPHAN OLARIU
Department of Computer Science, Old Dominion University, Norfolk, VA 23529, USA

BAHROM SANUGI
MOHD ISMAIL ABD AZIZ
Department of Mathematics, Faculty of Science, Universiti Teknologi Malaysia, 81310 Johor Bahru, Malaysia

Abstract. A complete graph is a fully-connected graph where every node is adjacent to all other nodes in the graph. Very often, many applications in science and engineering are reducible to this type of graph. Hence, a simplified form of a complete graph contributes in providing the solutions to these problems. In this paper, we present a technique for transforming a complete graph into a single-row routing problem. Single-row routing is a classical technique in the VLSI design that is known to be NP-complete. We solved this problem earlier using a method called ESSR, and, the same technique is applied to the present work to transform a complete graph into its single-row routing representation. A parallel computing model is proposed which contributes in making the problem modular and scalable. We also discuss the application of this work on the channel assignment problem in the wireless cellular telephone networks.

Keywords: complete graph, single-row routing, simulated annealing and channel assignments

1. Introduction

In many cases, problems in engineering and other technical problems can be represented as problems in graph theory. A problem of this nature is said to be reducible to the form of vertices and links of a graph, and the solution to the problem can be obtained by solving the graph problem. Furthermore, several solutions to the problems in graph theory have found their roots in some well-known prototype problems, such as the traveling salesman problem, the shortest path problem and the minimum spanning tree problem. Solutions to these problems are provided in the form of dynamic programming techniques, mathematical programming and heuristics. Most of these prototype problems have been proven to be NP-complete and, therefore, no absolute solutions to the problems are established. However, their reduction to the form of graphs have, in some ways, simplified their complexity and pave way for further improvement to their solutions.

In this paper, we study the relationship between a complete graph and its single-row representation. A complete graph is a graph where every vertex in the graph is adjacent to all other vertices. Single-row routing is a classical problem about finding an optimum routing from a set of *terminals*, or nodes, arranged in a single-row in the printed circuit boards (PCB) design. In the *Very Large Scale Integration* (VLSI) technology, the terminals are the pins and vias, and the routes consist of non-intersecting horizontal and vertical tracks called *nets*. The main goal in single-row routing is to find a realization that reduces the congestion in the network.

In this paper, we propose a model for transforming a complete graph as nets in a single-row axis. The motivation for this proposal comes from the fact that some problems in engineering are reducible to the form of a complete graph. A complete graph shows the working relationship between all pairs of nodes in the graph. The relationship, in this case, may represent parameters such as the precedence in a directed flow, the communication cost for transferring data and the matching between the nodes.

This idea may also apply to the subgraphs of the graph in the form of one or more cliques. A clique is a complete subgraph of a graph. Our suggestion in this case is to solve the original problem using the divide-and-conquer approach. The problem may be broken into several components where each component is represented as a clique in the graph. This suggests the formation of a parallel computing model with the cliques forming separate and independent modules. A clique may form its own computing base in a distributed computing system. A group of cliques may form a cluster in a parallel computing system.

In this work, we study the mapping properties of a complete graph into its single-axis representation, in the form of the single-row routing problem. We devise a strategy for mapping this graph, and then apply the method for solving a graph-reducible problem, namely, the channel assignment problem in the wireless cellular telephone networks. Channel assignment problem is a NP-hard problem which has its root in the graph coloring problem. The application of the complete graph transformation in the channel assignment problem suggests our method is applicable to the real world applications.

Our paper is organized into eight sections. Section 1 is the introduction. Section 2 describes the problem in the paper, while Section 3 presents the elementary symbols and terminologies used in the paper. In Section 4, we describe the single-row routing problem, while its solution using the simulated annealing method is discussed in Section 5. We also discuss our earlier model called the *Enhanced Simulated annealing for Single-row Routing* (ESSR) technique in this Section. In Section 6, we outline the details of the mapping strategy for converting the complete graph into its single-row axis representation. For the applications, we propose two parallel computing model for this problem in Section 7 involving a single-row multiprocessor network and a cellular network model for the channel assignment problem. We conclude the paper with the summary and conclusion in Section 8.

2. Problem formulation

The problem can be stated as follows: given a complete graph C_m, how can the edges in this graph be drawn so that they don't cross each other? The problem here translates into finding a planar graph. Our approach for solving this problem is to transform the complete graph into a single-row routing problem [9], as the single-row representation is a form of planar graph. It is easy to verify that a complete graph, C_m, with m vertices has $m(m-1)$ links (or edges). This is because each vertex in the graph has a degree of $(m-1)$.

Figure 1 shows a complete graph C_4 (left) and its single-row representation (right). In this transformation, each node in the graph forms a zone having the number of terminals

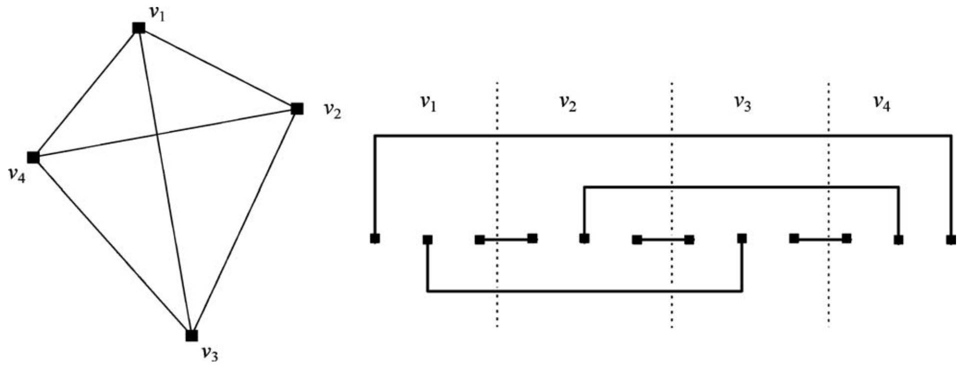

Figure 1. A complete graph C_4 (left) and its single-row representation (right)..

equals its degree, which is three in this example. The communication links between the nodes are still preserved in this transformation as each zone has exactly one link with all other zones in the network. Therefore, communication between the nodes in the original graph is maintained in this new representation.

The problem begins with the mapping of the links in this graph as terminals in a single-row axis. Single-row routing problem is an important component in finding an optimum routing in VLSI design [6, 9]. The single-row representation, S_m, of the graph, C_m, consists of m zones and $m(m-1)$ terminals, all aligned in a single-row axis. The terminals are to be formed in equally-spaced intervals along the single-row axis. In VLSI, each terminal represents a pin or via. In the single-row routing problem, nets joining pairs of terminals are to be formed to allow communication between the terminals. A net is made up of non-intersecting horizontal and vertical lines that is drawn in the order from left to right.

In order to produce a practical area-compact design, the nets have to be drawn according to the routes that will minimize the wiring requirements of the network. The main objective in the single-row routing problem is to determine the most optimum routing between pairs of the terminals so as to reduce the congestion in the whole network. It is also important that the routing is made in such a way that the interstreet crossings (doglegs) between the upper and lower sections of the single-row axis be minimized. This is necessary as the terminals in the single-row axis are very close to each other, and a high number of interstreet crossings will generate an intolerable level of heat that may cause problems to the network. In optimization, the problem of minimizing the congestion in the network reduces to a search for the right orderings of the nets, based on a suitable energy function.

3. Notations and symbols

Symbols are used based on two categories, namely, a graph and its single-row representation. A graph G consists of vertices, v_j, for $j = 1, 2, \ldots, m$, and a set of edges, or links, joining these vertices. To avoid confusion, the nodes in the single-row axis are referred to as *terminals* in this paper. The following notations are being used:

G	A graph
C_m	A complete graph with m vertices
S_m	Single-row representation of C_m
Q	Congestion of the nets in S_m
D	Number of interstreet crossings (doglegs) in S_m
E	Total energy in S_m
L	Partial order of nets arranged from top to bottom in S_m
v_j	Vertex j in the graph
d_j	Degree of vertex j in the graph
m	Number of vertices in the graph
t_i	Terminal i
b_k	Left terminal of net k
e_k	Right terminal of net k
n_k	Net k, given as $n_k = (b_k, e_k)$
$n_{y,i,m}$	The ith net in level y in S_m
$b_{y,i,m}$	Beginning (left) terminal of the ith net in level y in S_m
$e_{y,i,m}$	End (right) terminal of the ith net in level y in S_m
$w_{y,m}$	Width of every net in level y in S_m
$r_{y,m}$	Number of nets in level y in S_m
z_j	zone j in S_m

4. Single-row routing problem

Single-row routing is a combinatorial optimization problem that has been proven to be NP-complete [6, 9]. Traditionally, single-row routing is one of the techniques employed for designing the routes between the electronic components of a printed-circuit board. Each path joining the terminals is called a *net*. In the single-row routing problem, we are given a set of $2m$ evenly-spaced terminals (pins or vias), t_i, for $i = 1, 2, \ldots, 2m$, arranged horizontally from left to right in a single horizontal row called the *single-row axis.* The problem is to construct m nets from the list $L = \{n_k\}$, for $k = 1, 2, \ldots, m$, formed from horizontal intervals, (b_k, e_k), in the node axis, where b_k and e_k are the beginning and end terminals of the intervals, respectively. Each horizontal interval is formed from a pair of two (or more) terminals through non-intersecting vertical and horizontal lines. The nets are to be drawn from left to right, while the reverse direction is not allowed.

Figure 2 shows a realization in a single-row routing. Physically, each net in the single row represents a conductor path for its terminals to communicate. The area above the single-row axis is called the *upper street*, while that below is the *lower street*. The number of horizontal tracks in the upper and lower streets are called the *upper street congestion*, C_u, and *lower street congestion*, C_l, respectively. The overall street congestion Q of a realization is defined as the maximum of its upper and lower street congestions, that is, $Q = \max(C_u, C_l)$.

Various methods based on graph theory, mathematical programming and heuristics have been proposed to solve the single-row routing problem. In [4], Kuh et al. proposed some necessary and sufficient conditions for determining the most minimum congestion. In [1], a graph decomposition technique was proposed to obtain an optimum routing based on the

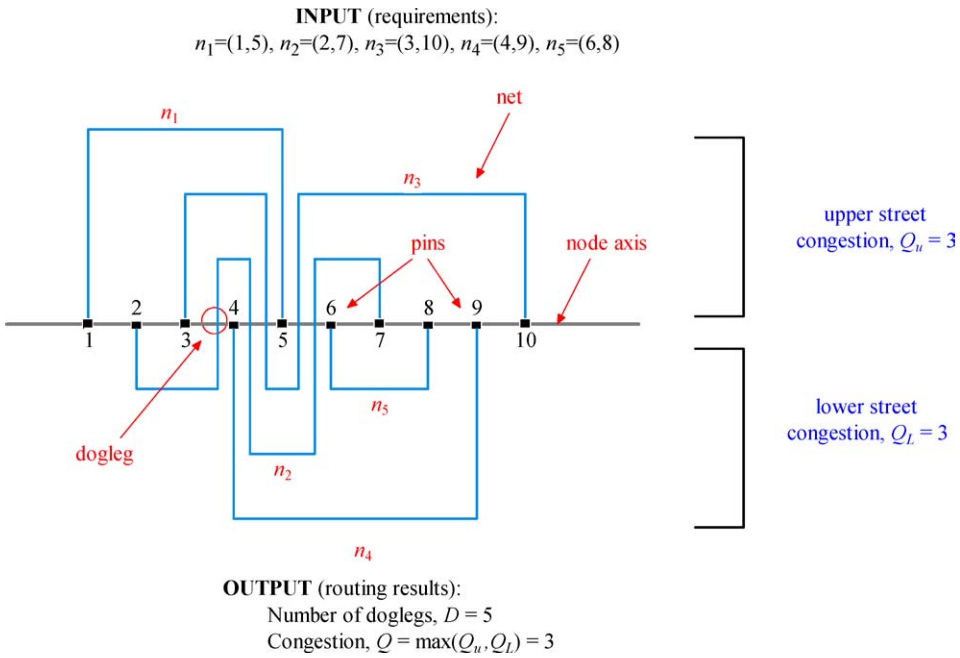

Figure 2. Terminologies in the single-row routing problem.

overlapping intervals of a graph. In [2], Du and Liu proposed a heuristic for finding an optimum routing based on a method that sorts the nets according to their classes, internal cut numbers and residual cut numbers.

5. Simulated annealing technique

In this section, we describe our previous method in [8] for solving the single-row routing problem using simulated annealing. Simulated annealing [3] is a heuristic method for solving combinatorial optimization problems using the atomic properties of particles undergoing thermal activities. As demonstrated by Rutenbar [7], it is possible to apply simulated annealing in VLSI design. In his work, simulated annealing has been applied to design an optimum layout for the chip layout and floor-planning.

Simulated annealing is a stochastic and hill-climbing heuristical method based on the gradient-descent optimization technique. Simulated annealing often produces good solutions that are comparable to other techniques. The simulated annealing method based on the Metropolis Algorithm [3] implements the Boltzmann distribution function as its energy minimization network. This probability function is known to have the mechanism to escape from getting trapped in a local minimum. Therefore, convergence is guaranteed although at the expense of a long computation time. In addition, simulated annealing is easier to implement as the objective function does not have to be in an explicit functional representation.

Simulated annealing involves a massive iterative improvement process of local search for the global minimum of a given energy function. This method is based on the simple and natural technique of trial and error. Initially, the process in simulated annealing requires the definition of a solution space, a cost function and a set of moves that can be used to modify the solution. Through the iterative improvement method, one starts with an initial solution, x_0, and compares this solution with its neighbors. A solution, x', is said to be a *neighbor* of a solution, x, if x' can be obtained from x via one of the moves. The neighbors of x_0 are examined until a neighborhood solution, x, with a lower cost is discovered. In this case, x becomes the new solution and the process is continued to examine the neighbors of the new solution. The algorithm terminates when it arrives at a solution which has no neighboring solution with a lower cost.

In our problem, a pertubation is performed to examine the neighbors by moving a net at random to a new position. The resulting change in energy ΔE is then evaluated. If the energy is reduced, that is $\Delta E < 0$, the new configuration is accepted as the starting point for the next move. However, if the energy is increased, $\Delta E > 0$, the move generates the probability of acceptance, given by Pr[acceptance] $= e^{-\Delta E/T}$. The move is accepted if this probability is greater than a threshold probability of acceptance, ε, and rejected otherwise. The value of ε is proportional to the rate of acceptance or rejection. With a higher value, the number of moves accepted for $\Delta E > 0$ are reduced and the same rule applies vice versa.

Our objective in this problem is to obtain a realization that minimizes both the street congestion, Q, and the number of doglegs, D. However, this objective is very difficult to achieve as the two components are separate but dependent entities. While having one component minimized, the other tends to show some resistance to its minimization. In [8], we proposed the Enhanced Simulated annealing for Single-row Routing (ESSR) method based on simulated annealing that produces a routing that minimizes both the congestion and the number of interstreet crossings. Figure 3 illustrates how ESSR is implemented to solve the single-row routing problem.

To express the above requirement, the energy in a given net ordering is expressed as the total length of all the tracks, according to the energy function, E, derived from our earlier work, as follows:

$$E = \sum_{k=1}^{M} \sum_{j=1}^{M_k} |h_{k,j}|. \tag{1}$$

In the above equation, $h_{k,j}$ is the energy of segment j in net k, while M is the number of nets in the problem and M_k is the number of segments in net k.

6. Complete graph partitioning strategy

A graph G consists of a set of vertices and edges, connecting some of these vertices. A graph where a path exists between any pair of vertices in the graph is called a *connected graph*, otherwise it is a *disconnected graph*. Node j in the graph having d links with its neighbors is said to have a degree of d_j. A graph with m nodes where every node is a neighbor of every other nodes in the graph is a complete graph, C_m. In C_m, every vertex has the same degree of $m - 1$.

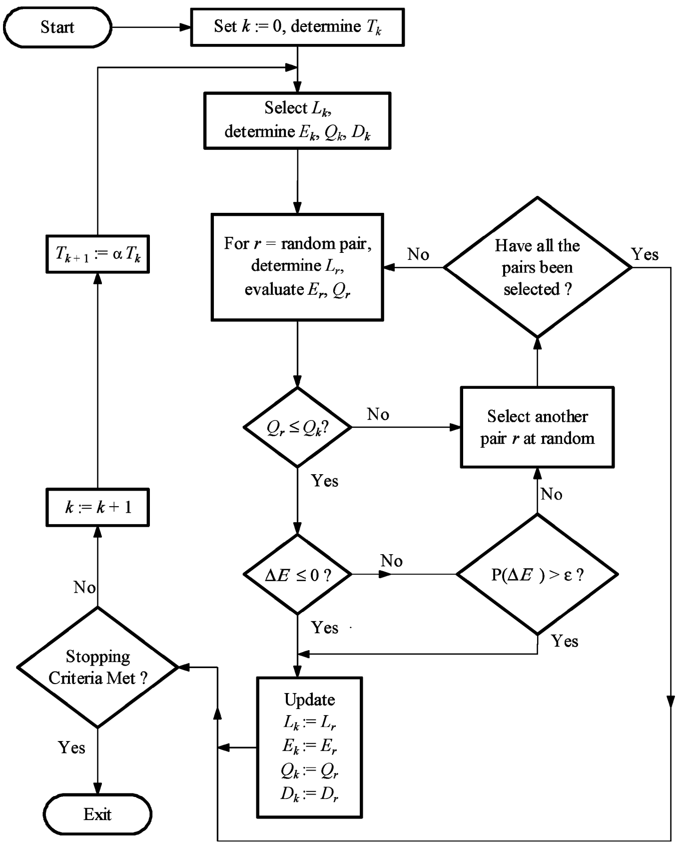

Figure 3. ESSR method for the single-row routing problem.

6.1. Formation of zones and terminals from a complete graph

In C_m, every link between a pair of vertices in the graph is mapped as a terminal in S_m. Therefore, a C_m graph having m vertices and $m(m-1)$ links is mapped into m zones with a total of $m(m-1)$ terminals in S_m. A vertex with degree j in the graph occupies a zone in S_m with d_j terminals.

We outline the overall strategy for mapping a complete graph. In general, the transformation of a complete graph, C_m, into its single row representation, S_m, consists of two main steps. First, the vertices, v_j, are mapped into the zones, z_j, that are numbered according to their vertex number, j, for $j = 1, 2, \ldots, m$. The next task is to determine the number of terminals in each zone, z_j, in S_m, which is simply the degree, d_j, of its corresponding vertex, v_j, in C_m. Finally, we obtain the complete layout of S_m by combining all the terminals from each zone and number them successively beginning from the first zone to the last.

Our method for creating the zones and their terminals in S_m from a complete graph, C_m, is outlined in `Algorithm 1`, as follows:

/* **Algorithm 1:** Formation of zones and terminals in S_m from C_m */
Given a complete graph C_m;
Draw the zones, z_j, in S_m, which corresponds to v_j in C_m, for $j = 1, 2, \ldots, m$;
for $j = 1$ **to** m
    Determine the degree, d_j, of every vertex, v_j, in C_m;
    Set $i = 1$;
    for $k = 1$ **to** d_j
        Set the terminal number, $t_i = i$;
        Update $i \leftarrow i + 1$;

6.2. *Construction of nets from a complete graph*

In the previous section, we described a plan to form the zones and nets in S_m from C_m using `Algorithm 1`. We illustrate the idea on the problem of forming a single-row representation of C_5, a complete graph with $m = 5$ vertices. In this problem, each vertex in the graph has a degree of 4. There are $m = 5$ zones, z_i, for $i = 1, 2, \ldots, 5$ and the number of terminals on the single-row axis is $m(m-1) = 20$. Hence, the number of nets formed is $r_m = \frac{m(m-1)}{2} = 10$. Figure 4 shows the zones and terminals in S_5 formed from C_5 when `Algorithm 1` is applied.

We now present a technique for forming the nets in the network that contributes in minimizing the total energy in S_m. The technique calls for the formation of the nets by grouping them first into several levels based on their width. The *width* of net k, denoted as w_k, is defined as the difference between its beginning and end terminals, given as $w_k = e_k - b_k$.

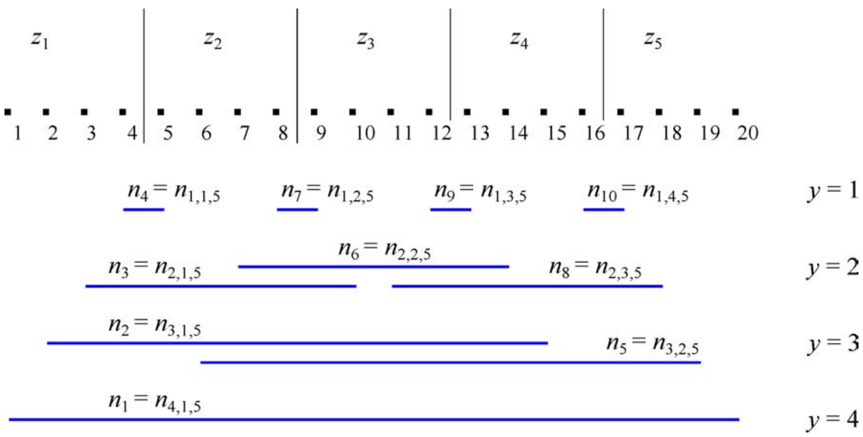

Figure 4. Formation of nets based on the zones and levels in S_5 from C_5.

A *level*, y, in S_m consists of a set of equal-width nets grouped in ascending order from the lowest width to the highest. Our strategy begins with Proposal 1 which consists of first forming levels where the nets with equal width are grouped. In this proposal, the nets in S_m are created by defining their end-points. Once the nets have been formed, the next step consists of sorting and renumbering the nets based on their beginning terminals, in ascending order from the lowest to highest. These two steps are summarized in `Algorithm 2`.

Proposal: The ith net in level y in S_m, denoted as $n_{y,i,m} = (b_{y,i,m}, e_{y,i,m})$, formed from the complete graph, C_m, is grouped into levels based on its width, $w_{y,m}$, according to the following relationships:

$$b_{y,i,m} = (m - y) + (m - 1)(i - 1), \tag{2a}$$

$$e_{y,i,m} = b_{y,i} + w_{y,m} \tag{2b}$$

for $y = 1, 2, \ldots, m - 1$, and $i = 1, 2, \ldots, m - 1$.

From Proposal, we obtain the width of the nets in level y of S_m, given as follows:

$$w_{y,m} = 1 + (m + 1)(y - 1), \tag{3}$$

and the number of nets in each level as follows:

$$r_{y,m} = (m - y). \tag{4}$$

The strategy for grouping the nets into levels based on their width is to minimize the total network energy, given in Equation (1). This goal can be achieved by forming nets starting from the shortest width, continue with the next shortest and so on. Starting with level 1, that is, $y = 1$, the nets are formed from two consecutive terminals from two different zones. This level has the most minimum width possible, given by $w_{1,m} = 1$. This minimum width has the advantage of producing the net energy equals 0, as the net can be drawn directly on the node axis. The ith net is formed from the last terminal in z_i and the first terminal in zone $(i + 1)$th, to make sure that the width remains the same. Using Equations (2a) and (2b) from Proposal, we then obtain the ith net in this level, $n_{1,i,m} = (b_{1,i,m}, e_{1,i,m})$, given as $b_{1,i,m} = (m - 1) + (m - 1)(i - 1)$ and $e_{1,i,m} = b_{1,i} + 1$.

In level 2, the first net is obtained by having the second last terminal in z_1 as its beginning terminal, and the second terminal of z_2 as the ending terminal. This gives the width as $w_{2,m} = 1+(m+1) = m+2$. In general, the ith terminal in this level, $n_{2,i,m} = (b_{2,i,m}, e_{2,i,m})$, is given by $b_{2,i,m} = (m - 2) + (m - 1)(i - 1)$ and $e_{2,i,m} = b_{2,i} + w_{2,m}$.

`Algorithm 2` summarizes our method for constructing the nets based on the levels of the nets with equal width. In this algorithm, the nets are formed based on Equations (2a) and (2b). The number of nets and their width in each level are determined from Equations (3) and (4), respectively. Once the nets have been formed, the algorithm then sorts and renumbers the nets based on their beginning terminals, in ascending order from the lowest to highest. `Algorithm 2` prepares the nets before the next important step, which is their execution in ESSR to determine their optimum routing.

```
/*Algorithm 2:  Construction of nets in S_m*/
Given a complete graph C_m with m vertices;
Let the number of nets in level 1, r_1 = r;
The initial width of nets in level 1 is 1, that is, w_{1,m} = 1;
for y = 1 to r
   if y > 1
      Update the width of the nets in level y, w_{y,m} ← 1 + (m + 1)(y − 1);
      Update the number of nets in level y, r_{y,m} ← (m − y);
   for i = 1 to r_y
      Form the ith net in level y as follows:
          Update the left terminal of net y, b_{y,i,m} ← (m − y) + (m − 1)
            (i − 1);
          Update the right terminal of net y, e_{y,i,m} ← b_{y,i} + w_{y,m};
for y = 1, 2, ..., r
   for i = 1, 2, ..., r_y
      Sort (b_{s,i,m}, e_{s,i,m}) in ascending order with b_{s,i,m} as the primary key;
for k = 1 to m(m−1)/2
   Assign n_k = (b_k, e_k) from the sorted (b_{s,i,m}, e_{s,i,m});
```

We illustrate the idea of constructing the nets using Proposal through the example in Figure 4. In this figure, a complete graph with 5 vertices, C_5, maps as S_5. The zones and terminals are obtained by applying `Algorithm 1`. By applying Equations (2a) and (2b) from Proposal 1, we obtain the nets grouped into 4 levels, as shown in Figure 4. `Algorithm 2` transforms the C_5 into S_5. Table 1 shows the nets obtained from this construction. We then apply ESSR to the nets to obtain the results in the form of an ordering with minimum energy, $E = 11$, as shown in Figure 5. The final realization of the network with $Q = 3$ and $D = 1$ is shown in Figure 6.

We also apply the method to several other models of complete graphs. Table 2 summarizes the results of these graphs with m vertices, C_m, in their single-row representations. Figure 6 shows the final realization of the routing obtained using ESSR from C_{10}.

7. Parallel computing model

Single-row routing technique is not restricted to the design of the printed-circuit boards only. We explore other potential benefits and suggest two of them in this section.

Table 1. Formation of nets in S_5 from C_5

Level, y	Width, $w_{y,5}$	#nets, r_y	Nets
1	1	4	(4,5), (8,9), (12,13), (16,17)
2	7	3	(3,10), (7,14), (11,18)
3	13	2	(2,15), (6,19)
4	19	1	(1,20)

Table 2. Summary of results for some complete graphs, C_m

C_m	#nets	E	Q	D
$m = 5$	10	11	3	1
$m = 6$	15	28	4	40
$m = 8$	28	128	9	21
$m = 10$	45	403	16	53

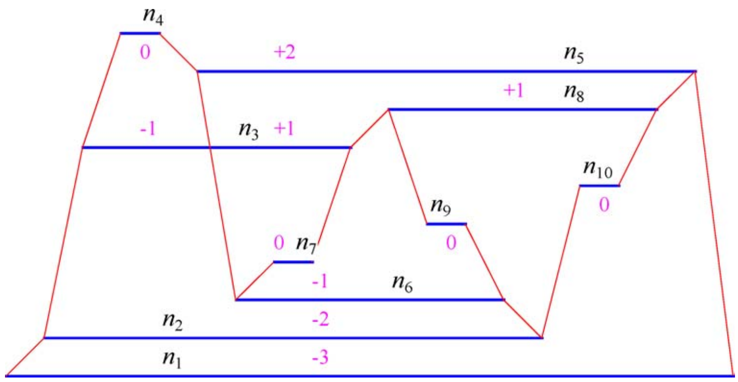

Figure 5. Nets ordering with minimum energy, $E = 11$, of C_5 using Algorithm ESSR.

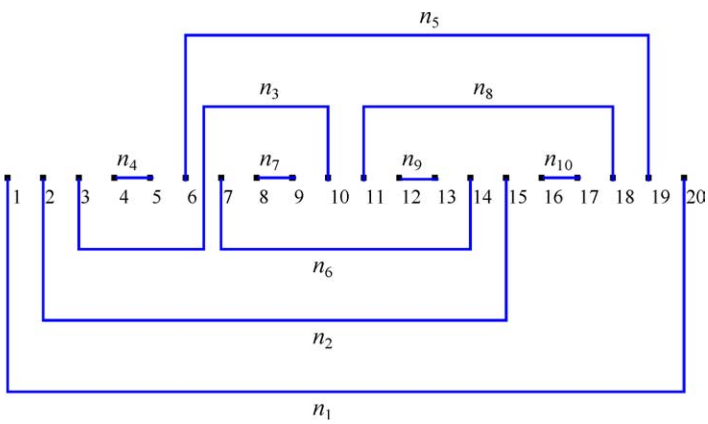

Figure 6. Final realization of C_5 with $E = 11$, $Q = 3$ and $D = 1$.

The models suggested are parallel computing networks involving the designs of the single-row multiprocessor network and the channel assignment problem in a wireless cellular telephone network. We describe the two models in a brief note according to their relationship to the single-row transformation problem.

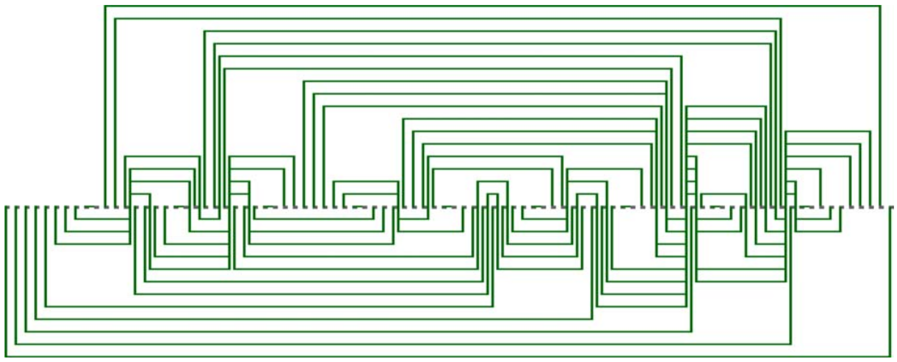

Figure 7. Realization of an optimum assignment of 45 nets from C_{10} in Table 2.

7.1. Single-row multiprocessor system

The transformation results in the form of non-crossing nets in the single-row model suggest the nets are independent of each other. In a single-row routing without doglegs, communication between the pairs of terminals in the single row can be established without passing through any intermediate node. A model called a single-row multiprocessor system is proposed, as shown in Figure 8. In this diagram, the processors are the shaded rectangles arranged in the single-row axis. The circles in the upper and lower streets of the network are the switches which can route the communicating lines into the north, south, east and west directions. Each communicating line in this network model is called a bus.

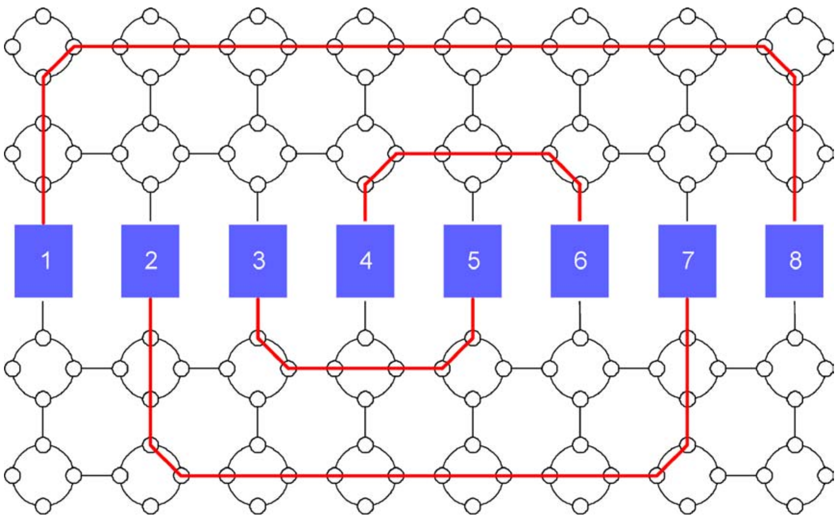

Figure 8. Single-row multiprocessor model having no doglegs.

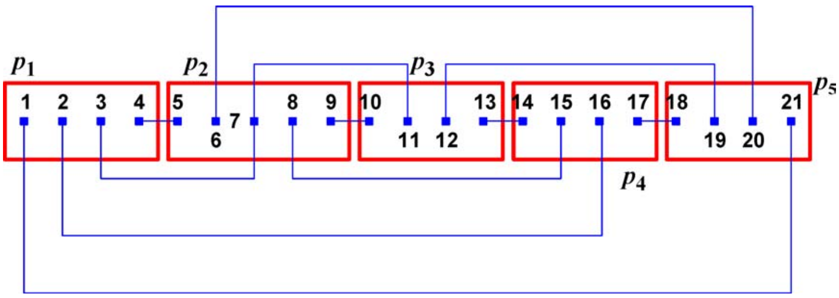

Figure 9. Single-row multiprocessor network with a dogleg.

Figure 9 shows another single-row multiprocessor model which involves doglegs. This model is derived from the transformation results of the complete graph C_5 into the single-row model S_5, which produces five zones. The routing results are obtained from ESSR with an optimum congestion of three and one dogleg. In this model, each zone formed in the single row axis is represented as a processor while the terminals form the ports in the processors. The processors in this diagram are labeled as p_i, for $i = 1, 2, \ldots, 5$. A dogleg in the network means the net between terminals 3 and 11 crosses the single row axis through p_2. In this case, an extra port is created in this processor to enable the crossing to take place. It follows that doglegs can be represented as extra terminals in the host processors at the crossings.

7.2. *Application to the channel assignment problem*

The single-row mapping strategy can also be applied to the problem of assigning radio channels in a wireless cellular telephone network. In the wireless cellular telephone network [5], the assignment of radio frequencies to the mobile users within the network can be modeled as the problem of mapping a complete graph into non-intersecting single-row nets. This network consists of a geographical region partitioned into several cells where each cell is allocated with a *base station* (BS). A base station has a transmitter and a receiver for receiving and transmitting messages from/to mobile users within its cell. The base stations in the network are linked with high-speed cables to the *mobile switching center* (MSC), which functions as a controller to allow communication between any two mobile users in the network by assigning a channel each to each of them.

Figure 10 shows two cells in a cellular network where communication is established using the single row routing technique. The model is produced from the transformation of C_6 into S_6. There are 42 channels and there are represented as terminals in the node axis. When a call is made from a cell, a request is received by the base station in the cell. The base station relays this request to the mobile switching center (MSC). Assuming the call is made and received within the network, a channel each needs to be assigned to the caller and the receiver. In this network, MSC plays an important role in assigning a

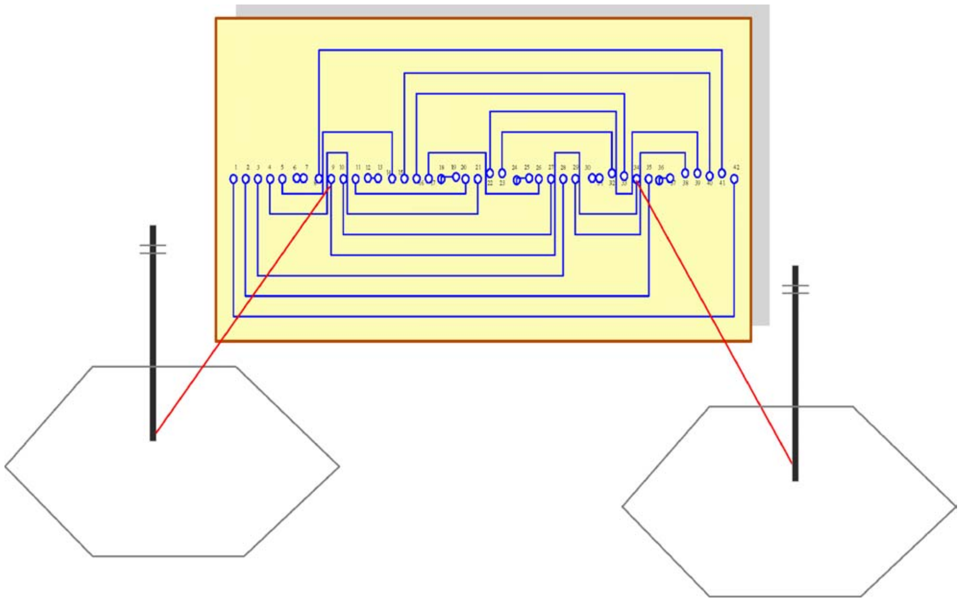

Figure 10. Single-row routing model for the cellular network.

pair of different channels to both the caller and the receiver, to allow immediate circuit switching.

In the channel assignment problem, we model the channels as the edges of a complete graph. The cells in the network are then represented as nodes in the graph. In the single-row axis, each of these cells is a zone and the channels allocated to a cell are terminals in the zone. Communication between two mobile users from two different cells is established through a net linking their two terminals. We model the single-row communication to be handled by the mobile switching center. This is because MSC has a control on all channel assignments in the network, and this important task must be done immediately without delay when requests for calls are received. In addition, MSC must also be able to provide services associated with problems in channel assignments, such as location finding of mobile users, and channel handovers as a mobile user moves from one cell to another.

We illustrate our model using an example with a network of 5 cells. The problem reduces to the complete graph, C_5, which is represented as the zones, z_j, for $j = 1, 2, \ldots, 5$ in S_5, as shown in Figure 5. Hence, 20 channels are available for assignments and each of these channels is represented as a terminal in the single-row axis. The channels are formed using the same technique discussed in Section 6.2, to produce the results as shown in Table 1 (the channels are numbered by assuming there are no electromagnetic interferences on the channels) . Figure 6 is then the final realization of the optimum routing of the nets obtained using ESSR.

8. Summary and conclusion

In this paper, we propose a method for transforming a complete graph, C_m, into its single-row representation, S_m. We first describe the single-row routing problem, which is a classical technique in VLSI design. We relate the problem to our earlier work which solved the problem using a method based on simulated annealing, called ESSR. The transformation from C_m to S_m involves the formation of nets based on Proposal. The proposal groups the nets with equal width which contributes in reducing the overall energy of the network. The whole process is implemented using `Algorithms 1` and `2`. We then apply ESSR to the network to obtain some reasonably good results for optimum routing. Finally, we also describe briefly the application of this transformation technique in solving the channel assignment problem in the wireless cellular telephone networks.

References

1. J. S. Deogun and N. A. Sherwani. A decomposition scheme for single-row routing problems. Technical Report Series #68, Dept of Computer Science, Univ. of Nebraska, 1988.
2. D. H. Du and L. H. Liu. Heuristic algorithms for single row routing. *IEEE Trans. Computers*, 36(3):312–319, 1987.
3. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing. *Science*, 220(4598):671–678, 1983.
4. E. S. Kuh, T. Kashiwabara, and T. Fujisawa, On optimum single-row routing. *IEEE Trans. Circuits and Systems*, 26(6):361–368, 1979.
5. R. Mathar and J. Mattfeldt, Channel assignment in cellular radio networks. *IEEE Trans. Vehicular Technology*, 42(4), 1993.
6. R. Raghavan and S. Sahni, Single row routing. *IEEE Trans. Computers*, 32(3):209–220, 1983.
7. R. A. Rutenbar, Simulated annealing algorithms: An overview. *IEEE Circuits and Devices Magazine*, 19–26, 1989.
8. S. Salleh, B. Sanugi, H. Jamaluddin, S. Olariu, and A. Y. Zomaya. Enhanced simulated annealing technique for the single-row routing problem. *Journal of Supercomputing*, 21(3):285–302, 2002.
9. B. S. Ting, E. S. Kuh, and L. Shirakawa, The multilayer routing problem: Algorithms and necessary and sufficient conditions for the single row, single layer case. *IEEE Trans. Circuits Systems*, 23:768–778, 1976.