Jurnal Teknologi, 55 (Sains & Kej.) Keluaran Khas (1), Mei 2011: 241-253

PARTITIONING TECHNIQUE FOR TRANSFORMATION OF CONNECTED GRAPHS INTO SINGLE-ROW NETWORKS

SERLEE LOH [1], SHAHARUDDIN SALLEH [2*] & NOR HANIZA SARMIN[3]

Abstract. In this paper, we present our work called Connected Graph Sequence (CGS) which transforms a partially dense graph into the single-row network. Partially dense graph is a graph where a number of connected components, namely subgraphs, are connected by some links and each subgraph has a higher density value compare to the graph. The transformation is necessary in applications such as in the assignment of telephone channels to caller-receiver pairs roaming in cells in a cellular network on real-time basis. In this channel assignment application, each caller and receiver in a call is treated as a node, while their pair connection is treated as the edge. A specific case of the graph in the form of a partially dense graph is then transformed into its corresponding single-row network for assigning the channels to the caller-receiver pairs.

Keywords: Partially dense graph; single-row network; transformation; subgraph; graph partitioning

Abstrak. Dalam kertas kerja ini, kita membentangkan kerja kita bertajuk Susunan Graf Berkait (CGS) di mana ia menukarkan graf sebahagian pahat kepada rangkaian baris tunggal. Graf sebahagian padat merupakan graf di mana sejumlah komponen berkait, bernama subgraf-subgraf, dikaitkan oleh beberapa sambungan dan setiap subgraf mempunyai nilai ketumpatan yang lebih tinggi berbanding dengan graf itu. Transformasi diperlukan dalam applikasi seperti penetapan saluran telefon kepada pasangan pemanggil-penerima berayau dalam sel dalam rangkaian radio atas dasar masa-nyata. Dalam applikasi penepatan saluran, setiap pemanggil dan penerima dalam satu panggilan dekenali sebagai nod sementara pasangan kaitan mereka dikenali sebagai sambungan. Satu kes spesifik graf dalam bentuk graf sebahagian padat ditransformasi kepada rangkaian baris tunggal tersebut untuk menetap saluran kepada pasangan pemanggil-penerima.

Kata kunci : Graf sebahagian padat; rangkaian baris tunggal; transformasi; subgraf; graf partisi

[1,2,3] Department of Mathematics, Faculty of Science, Universiti Teknologi Malaysia, 81310, UTM Johor Bahru, Johor Darul Ta'azim, Malaysia
* Corresponding author: ss@utm.my

1.0 INTRODUCTION

Single-row routing problem has been shown to be NP-complete problem [1]. 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 problem ([1] and [2]). In [3], a partitioning strategy was proposed to group the nets into zones which produces some reasonably good solutions for some restricted models. Bhattacharya et al. proposed a new approach based on graph theoretic representation in [4] which relates the intervals of the single-row network with the overlap and interval graphs to solve the single-row routing problem.

A model called *Enhanced Simulated Annealing Technique for Single Row Routing* (ESSR) was proposed in 2002 to optimize the network by minimizing both the congestion and number of doglegs [5]. When the total energy value is being minimized, congestion and number of doglegs are minimized as well. Based on the simulated annealing technique [8], the energy function in ESSR is a function of the height of the segments of the nets in the single-row network. The technique has successfully been applied to produce optimal solutions to all net sizes.

Many engineering and science problems can be represented as a problem in graph theory. The graph represents the scenario of the real-life applications where the nodes in the graph can be treated as nodes in a network, and the edges are representing the communication links between the nodes. The relationship between a complete graph and its single-row representation was first formulated in [6] and [7]. Both models discuss the technique of transforming a graph into a single-row network where ESSR [5] is applied to produce optimal results. The transformation finds its application, for example, in assigning telephone channels to caller-receiver pairs roaming in cellular regions in a cellular network on real-time basis.

The relationship between a graph and its single-row network is demonstrated through the transformation in [6]. The nodes in graph G are mapped to zones in single-row network S while the intervals are created for every edge in G. Each zone has a number of terminals which equal to the degree of the corresponding node. The set of intervals formed are drawn as non-crossing paths joining two terminals in the single-row network S.

Our present work focuses on a special type of graph called partially dense graph. Partially dense graph is a graph where a number of connected components, namely subgraphs, are connected by some links and each subgraph has a higher density value compare to the graph. In this work, we focus on the transformation of partially dense graph into its single-row representation before applying ESSR for optimal results. We propose a model called *Connected Graph Sequence* (CGS) which performs single-row transformation through three steps: The graph is

partitioned into a number of subgraphs with higher density of graph, the subgraphs are then arranged on planar axis in such a way every pair of subgraphs with more interlinks between them are closer to each other, then the zones are formed based on the subgraphs arrangement and the connection among the nodes to obtain the optimal nodes labeling. CGS is followed by ESSR for producing an optimal single-row network.

2.0 PROBLEM STATEMENT

The transformation problem can be stated as follows:

> *Given a connected graph in the form of a partially dense graph, how can it be transformed optimally into a single row network S so as to minimize the congestion and number of doglegs in the network?*

The problem is illustrated in Figure 1. The figure shows a partially dense graph, ψ_{10} where $n = 10$ is the size, and its single-row expansion where each node in ψ_{10} occupies a zone z_i in the single-row network, S. In the single- row transformation, a node in ψ_{10} is mapped to d terminals in S where d is the degree of the node.

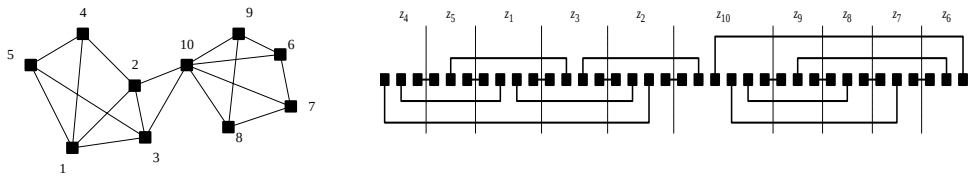


Figure 1 Partially dense graph ψ_{10} and its mapping to the single-row network S

1.1 Partially Dense Graph and Single-Row Network

Our earlier work in [6] involves the transformation of an arbitrary connected graph G into the single-row network S where each edge in G forms a net in S. This stems the idea to transform a partially dense graph, ψ, into single-row network through partitioning of graph. For example, in cellular telephone network, a subgraph may represent a city while the link between a pair of nodes in a subgraph is a matching between the two nodes which may represent a real-time telephone line between two users from different locations in a city. Hence, the transformation

from ψ to S, has potential in producing an optimal path between the two users from either a same or different cities for fast and uninterrupted communication.

1.2 Single-Row Network

A single row network is a network in planar form which can be found on printed circuit board. Terminals are arranged horizontally on a single horizontal line known as single-row axis. They are joined by nets where the nets are made up of horizontal and vertical line segments drawn from left to right to form the non-crossing paths, and a reverse direction is not allowed.

The number of horizontal tracks in upper street is called an *upper congestion*, Q_u, while the number of horizontal tracks in lower street is called a *lower congestion*, Q_l. The overall street congestion Q of a realization is defined as the maximum of its upper and lower street congestions, namely $Q = \max\{Q_l, Q_u\}$. A vertical crossing in single-row axis is called a *dogleg*. The main objective in single-row routing problem is to minimize the congestion of a single-row representation.

1.3 Partially Dense Graph

Partially dense graph has the property where the set of vertices can be partitioned into a number of vertices sets called subgraphs in which each subgraph has a higher density value than the graph itself. The links in a subgraph are called *intralinks* while the links connect the subgraphs are known as *interlinks*. In the subgraphs, every node has a higher degree of intralinks compared with interlinks (if any) and thus, the density of graph for each set of nodes is higher than the graph ψ. The density of graph is computed as

$$Density = \frac{2|E|}{|V|(|V|-1)} \tag{1}$$

where E and V are set of edges and set of vertices respectively.

In a partially dense graph, ψ, there are n nodes where each node v_i has d_i degree. The nodes v_i in ψ will be transformed into zones z_i in single-row network S where each zone has d_i terminals for $i = 1,2,...,n$. The terminals of zones are aligned evenly on a single node axis where the sequence of zones in S matches node labeling in ψ. The labeling order of the terminals is critical as this

will lead to the degree of congestion in the single-row network. The communication links or edges between v_i and v_j in ψ are preserved as the nets that connect the z_i and z_j in S.

In order to achieve optimality in S, the zones are arranged based on the graphical properties of ψ. For example, from Figure 1, graph ψ can be partitioned into two subgraphs; first with nodes number one to five and second with nodes number six to ten. Both subgraphs have the graph density of 0.8 compared with 0.4 given by graph ψ. The zones correspond to the nodes from a subgraph are then arranged as a group of zones in S to reduce the congestion and doglegs caused by nets from different group of zones. This idea contributes to the optimal realization from the single-row transformation.

High value of congestion leads to system overhead and doglegs reduce the transmission efficiency on printed circuit board. Hence, our main concern in this work is to optimize the single-row transformation in such a way it minimizes the energy value of single-row representation by ESSR as the lower energy represents the lower congestion and number of doglegs.

2.0 CGS THE TRANSFORMATION MODEL

In single-row transformation of connected graph, DSA is an efficient model for connected graph with smaller size while STCGM is an efficient model for connected graph with low density of graph. As the size and the density of graph increased, both models are incapable to produce a good result for the single-row transformation problem. Hence, the connected graph with bigger size and higher density needs to be partitioned into a number of subgraphs. There are three main steps comprised in CGS, namely graph partitioning to obtain a number of higher density subgraphs, partitions arrangement to sort the subgraphs based on the links between the subgraphs, and formation of zones followed the sequence of sorted subgraphs.

2.1 Graph Partitioning

The edges between the pair of nodes in a subgroup are called *intralinks* while the edges between the nodes from any two subgraphs are called *interlinks*. Each of the subgraphs formed has a higher density of graph compared to the graph ψ. Thus, the total of intralinks in a subgraph is greater than its total of interlinks. In the single-row transformation, the nodes in the subgraph will be transformed into zones which allocated near to each other as a group of zones on the node axis. The

construction of intervals set for each group of zones is shorter and more systematic compared to the set of intervals corresponds to the interlinks. Obviously, the arrangement of the zones and intervals for the single-row representation is well organized with the partitioning of graph ψ since it reduces the congestion and doglegs caused by interlinks. To partitioning a graph into a number of subgraph, we need to reduce all the nodes into stable or conditional stable form.

A node is said to be stable in a subgraph if it has more intralinks compared to interlinks and vice versa for an unstable node. For example, stable node v_1 in subgraph A has one intralink without any interlink while node v_3 has three interlinks and one intralink showing its instability in subgraph A. A node, for example, node v_5 is conditional stable if it has the equal number of interlinks and intralinks. If all nodes are stable in their own subgraphs, the partitioning of graph is said to be in a stable form. However, in some cases, there are some graphs which can never be reduced into a stable form unless the related subgraphs are combined into one bigger size of subgraph as shown in Figure 2.

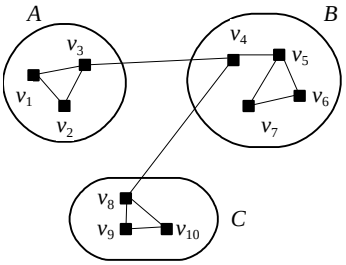
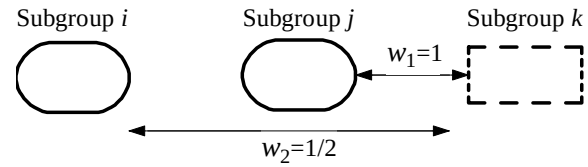


Figure 2 Illustration of unstable form in subgraph B

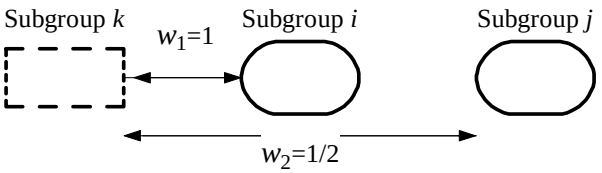
During the partitioning process, the total number of interlinks for all nodes and the total number of subgraphs are decreased as the nodes are removed from one subgraph and combined with another to achieve the higher stability form. Each subgraph has a higher density of graph compared to the graph ψ after the completion of graph partitioning. The zones from a denser subgraph need to be arranged in such a way they are closer to each other on node axis in S. This implies the sequence of subgraphs needs to be arranged by taking into consideration of the connection between the subgraphs.

2.2 Partition Arrangement

In partition arrangement, the sequence of subgraphs is arranged to form the zones in next stage. The number of interlinks between two subgraphs reveals the strength of the bond between them. The partitions of zones correspond to the subgraphs which have the higher bond strength between them should have the closer placement on the node axis in *S*. This arrangement produces shorter set of nets leads towards lower congestion and doglegs. The number of interlinks between any pair of subgraphs i and j is determined and stored as g_link_{ij}. Partition arrangement starts from the pair of subgraphs i and j with highest degree of g_link_{ij}. The subgraph with bigger size is chose when a tie is exist. The fitness values for the rest of subgraphs are then computed as shown in Figure 3.



$$weight_{Rk} = w_1 \times g\_link_{jk} + w_2 \times g\_link_{ik}$$



$$weight_{Lk} = w_1 \times g\_link_{ki} + w_2 \times g\_link_{kj}$$

Figure 3 Computational of $weight_{Rk}$ and $weight_{Lk}$

The subgraph k with the highest degree of fitness value, $\max\_w$, among $weight_{Rk}$ and $weight_{Lk}$ for $k = 1,...,Ttl\_grp$ is chosen. For example, if the $\max\_w$ is given by $weight_{Rk}$, the sequence of subgraph will be updated as i, j, k and vice versa, k, i, j. Somehow, when there exists a tie on the $\max\_w$, the Look Ahead Tie-Breaking Strategy may used to break the tie. The look ahead fitness value, $LA\max\_w$, for each tie case is computed as the previously fitness value computation by including the subgraph $k+1$ to obtain the most promising sequence of subgraphs. Choose the case with the highest degree of $LA\max\_w$ as the tie breaking case. The iteration is repeated until the sequence of subgraphs is formed.

The optimal sequence of subgraphs is stored as P_k with n_k nodes for $k = 1,...,Ttl\_grp$. The n_k nodes from subgraph P_k is then transformed into n_k zones in partition of zone, $Part_k$, on the node axis in S.

2.3 Formation of Zones and Terminals

The sequence of the n_k zones within each partition is then sorted out in such a way it is optimal in terms of result. The connection of the interlinks is also took into consideration for the zones formation. In order to optimize the nodes labeling which contribute towards optimal arrangement of nets correspond to the interlinks, the degree favors to the left and right for all zones i, $i = 1,..., n$ in each partition are computed and denoted by v_Lfav_i and v_Rfav_i respectively. The node with more interlinks with the subgraph at the left ought to be allocated at the left and vice versa for lower energy value.

Degree favors to the left , v_Lfav_i, is computed by the sum of interlinks of node i with subgraphs on the left where the interlinks from the nearer subgraph to the node is given a higher value of weight. In order to obtain the optimal sequence of zones within the subgraph, the arrangement of zones needs to lead towards shorter set of nets in formation of zones. This includes nodes with higher degree are allocated near to the middle of their own partitions of node axis and the arrangement of zones gives as much as possible of zero energy nets. Besides, the interlinks between the subgraphs need to be considered as well. Hence, sequence of zones for every subgraph is done by computing the first, last and finally middle part of the zones in alternating way.

In computing the first zone in every partition, except the first partition, the weight for node i allocated to the left, v_L_i, is computed as $v\_L_i = \frac{\alpha + \beta * v\_Lfav_i}{d_i}$, parameters α and β are both set to be 0.5 based on empirical test. Parameter α adds the favourite to the nodes in P_k which connects to the node corresponded to the last zone in P_{k-1}. For the last zone in each partition, except the last partition, the weight for node i allocated to the right, v_R_i, is computed for every available node i as $v\_R_i = \frac{\beta * v\_Rfav_i}{d_i}$.

Hence, in each subgraph, the node with highest value of v_L_i (v_R_i) will be transformed and allocated as the first (last) zone in the corresponding partition. The zones in the middle part are then filled up alternately by using the same way as first and last zones are formed. The formation of zones is done after the nodes from all subgraphs are mapped to zones accordingly.

Terminals for the zones are formed after the completion of formation of zones. Terminals are joined by the intervals which formed in the next stage. Each zone has d_i terminals where d_i is the degree of each corresponding nodes. All terminals are aligned on node axis and numbered successively follow the order of zones' sequence.

2.4 Formation of Intervals

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 connected zones. This part produces the nets with zero energy, which contribute towards minimum congestion in the network.

In the second part, every partition of zones, $Part_k$ for $k = 1,..., Ttl\_grp$, has a list $g = \{g_1,..., g_{n_k}\}$ which defined as the zone ordering of n_k zones for interval formation. Set the middle zone z_{c_j} in $Part_k$ as the first element ($g_1 = j$) in the 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 the left and the right hand side are set to be the second and third elements respectively in the g list. It follows the two zones which are second next from the middle zone at the left and the right hand side are set as the fourth and fifth elements, respectively. The process is repeated until the whole ordering list is completed.

In 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 the left hand side which is nearest and connected with $z_{c[g_1]}$. Each zone at the left hand side of $z_{c[g_1]}$ is checked one by one in the direction leftward from $z_{c[g_1]}$. The process is repeated until all the zones on the left hand side which are connected with $z_{c[g_1]}$ each produces a net formed from $z_{c[g_1]}$. It follows that a net is formed between the last available terminal in $z_{c[g_1]}$ and the first available terminal in the zone at the right hand side which is nearest and connected with $z_{c[g_1]}$. Then, each zone at the right hand side of $z_{c[g_1]}$ will be checked one by one in the direction from $z_{c[g_1]}$ rightward. The process is repeated until all the zones at the right hand side which are connected with $z_{c[g_1]}$ each produces a net formed from $z_{c[g_1]}$.

When the net construction is completed for $z_{c[g_1]}$, the iteration is repeated for $z_{c[g_2]}$ one by one until the last zone, $z_{c[g_{nk}]}$ to produce a complete set of intervals

for $Part_1$. The procedure is repeated for next $Part_k$ until all partitions of zones form their sets of intervals. The intervals correspond to the interlinks are then formed using the same procedure by treating each $Part_k$ as a zone to form the initial single-row network. The step where intervals within each partition of zones are constructed before the intervals for interlinks is to reduce the energy value.

3.0 SIMULATIONS AND RESULTS

A program has been developed based on the CGS model's algorithms using Microsoft Visual C++ 6.0. The CGS model has been applied for partially dense graph with different sizes and graph densities. This specific type of connected graph has the property where it can be partitioned into a number of vertices sets and each set has a higher density value than the graph. The simulations are run and compared by two models from our previous works, STCGM and DSA. The results are shown in Table 1.

The simulations shows CGS model gives the best result compared with STCGM and DSA models for graph with sizes ranging from $n = 10$ to 30 and densities ranging from 0.1 to 0.6 except the case where the graph is a tree. The results for each size and density of graph from three different methods are being compared and presented in Figure 4(a), (b) and (c) accordingly. It is clear that the energy values are proportional related to the size and density of graph for all three methods.

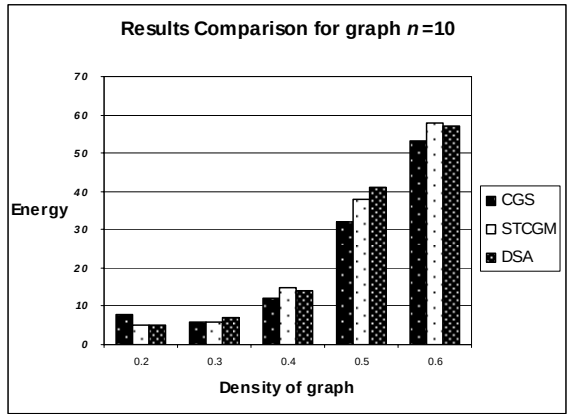


Figure 4(a) Results comparison between CGS, STCGM and DSA for with densities 0.2 to 0.6

A tree with n nodes has $n-1$ edges. From the Equation (1), a graph with n nodes which has the minimum density of $2/n$ is a tree. In the cases of graph $n=10$ with density of graph 0.2 and graph $n=20$ with density of graph 0.1, in which they are trees, CGS gives the higher energy value of eight units compared with five units given by STCGM and DSA as shown in Figure 4(a) and (b).

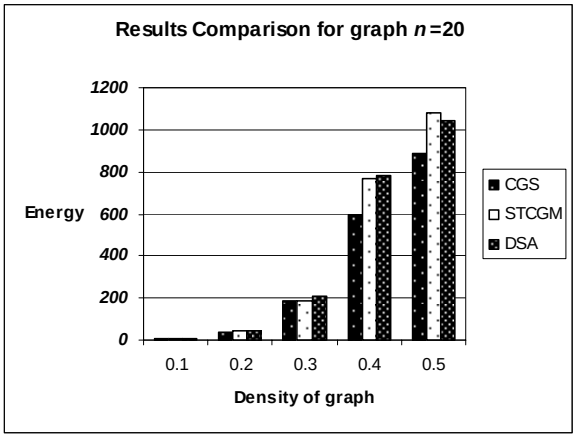


Figure 4(b) Results comparison between CGS, STCGM and DSA for g $n=20$ with densities 0.1 to 0.5

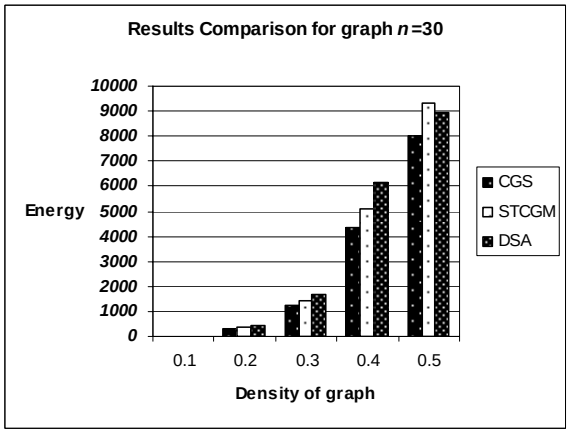


Figure 4(c) Results comparison between CGS, STCGM and DSA for gra with densities 0.1 to 0.5

Figure 4(a), (b) and (c) show STCGM gives the good results for graphs with lower density which is less than 0.4 while DSA works efficiently to the graphs with smaller size. STCGM produces a spanning tree from the given partially dense

graph before solving it using TSM [11], a model from our previous work which specially developed for single-row transformation of tree; the sequence of zone formed is then used to solve the single row transformation of the given partially dense graph. Hence, STCGM can produce a good result for graph with lower density as tree has the minimum density of graph.

As the size of graph increases, the different between energy values of CGS and DSA is getting larger. DSA applied simulated annealing twice in forming the zones and intervals. The sequence of zones is determined by changing a pair of nodes in every iteration using simulated annealing. Thus, it is efficient in transforming smaller size of connected graphs into single-row network.

For all cases, CGS produces the better results with lower energy value compared to STCGM and DSA except in the cases where the partially dense graph is a tree. Besides having the low congestion, CGS produces the least doglegs in all cases. CGS partitions partially dense graph into a number of connected components called subgraphs and these subgraphs are connected by some links called interlinks. The subgraphs are arranged in such a way the more interlinks between a pair of subgraphs, the closer of the pair of subgraphs are allocated. This lead towards systematic arrangement of zones and intervals for each subgraph to produce an efficient routing with lower congestion and doglegs.

4.0 SUMMARY AND CONCLUSION

Partially dense graph is a graph which can be partitioned into a number of vertices sets where each set has a higher density degree compare to the graph. Based on the properties of partially dense graph, our work proposes a new technique for transforming a partially dense graph into a single-row network. We proposed a new model called Connected Graph Sequence (CGS) which includes three steps, namely, the graph partitioning, and its arrangement. The nodes are then mapped to zones and terminals of the single-row network. The terminals produced from CGS are arranged optimally based on the properties of the partially dense graph, and this leads into the formation of intervals or nets in the single-row network. We then apply our earlier simulation model to the intervals called ESSR to produce an optimal single-row network by minimizing the energy.

CGS has been tested using 15 different models of partially dense graph with sizes ranging from 20 to 30 and densities ranging from 0.1 to 0.6. Both of our earlier work, STCGM and DSA models have been applied for the 15 cases to compare the simulation results from CGS. The simulations show CGS is outstanding in terms of results since it produces optimal results in each case for minimum energy in the final realizations.

Partially dense graph is one of our works in generalizing the transformation technique to an arbitrary graph. All types of graph have the inherent properties that make them have the difficulty in optimally transformation using a same model. Hence, the analysis in single-row transformation of graph needs to be done case by case to achieve the objective of optimal single-row transformation.

ACKNOWLEDGMENT

The authors would like to thank the Malaysian Ministry of Higher Education and Universiti Teknologi Malaysia for their financial funding through FRGS grant 78330.

REFERENCES

[1] B. S. Ting, E. S. Kuh, L. Shirakawa. 1976. 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.

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

[3] T. T. Tarng, M. M. Sadowska, and E. S. Kuh. 1984. An Efficient Single-Row Algorithm. *IEEE Transactions on Computer Aided Design*. 3(3): 178-183.

[4] B. B. Bhattacharya, J. S. Deogun, and N. A. Sherwani. 1988. A Graph Theoretic Approach to Single Row Routing Problems. *Proceedings of the 1988 IEEE International Symposium on Circuit and Systems*. 2: 1437-1440.

[5] S. Salleh, B. Sanugi, H. Jamaludin, S. Olariu, and A. Y. Zomaya. 2002. Enhanced Simulated Annealing Technique for the Single-Row Routing Problem. *Journal of Supercomputing*. 21(3): 285-302.

[6] S. Salleh, S. Olariu and B. Sanugi. 2005. Single-row Transformation of Complete Graphs. *Journal of Supercomputing*. 31: 265-279.

[7] S. Salleh, S. Olariu, A. Y. Zomaya, L.Y. Kiew and N.A.B. Aziz. 2007. Single-Row Mapping and Transformation of Connected Graphs. *Journal of Supercomputing*. 39:73-89.

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

[9] S. L. Loh, S. Salleh and N. H. Sarmin. Double Simulated Annealing Model for Mapping of Graphs to Single-Row Networks. Unpublished.

[10] S. L. Loh, S. Salleh and N. H. Sarmin. 2009. Spanning Tree Transformation of Connected Graphs into Single-Row Networks. *World Academy of Science, Engineering and Technology*. 62: 597-601.