SIMULTANEOUS ESCAPE ROUTING USING NETWORK FLOW OPTIMIZATION

***Kashif Sattar*[1]*, Aleksandar Ignjatovic*[2]**

[1] School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad, Pakistan

[2] School of CSE, University of New South Wales (UNSW), Sydney, Australia

E-mail: kashif.sattar@seecs.edu.pk[1], ignjat@cse.unsw.edu.au[2]

ABSTRACT

With the advancement in technology, the size of electronic components and printed circuit boards (PCB) is becoming small while the pin count of each component is increasing. This has necessitated the use of ball grid array (BGA) type of components where pins are attached under the body of component as a grid. The problem of routing pins from under the body of component to the boundary of the component is known as escape routing. It is often desirable to perform ordered simultaneous escape routing (SER) to facilitate area routing and produce elegant PCB design. The task of SER is non-trivial, given the small size of components and hundreds of pins arranged in random order in each component that needs ordered connectivity. In this paper, first we propose flow models for different inter pin capacities. We then propose linear network flow optimization model that simultaneously solves the net ordering and net escape problem. The model routes maximum possible nets between two components of the PCB, by considering the design rules. Comparative analysis shows that the proposed optimization model performs better than the existing routing algorithms in terms of number of nets routed.

Keywords: PCB Routing, Planer Graph, Inter-pin capacity, Network Flow Model, SER Optimization Model.

1.0 INTRODUCTION

There is a constant demand for reducing the weight and size of electronic devices. Small light weight devices require sophisticated design and significantly advanced technologies for producing small sized electronic circuit boards. Smaller size of electronic circuits with more pin count [1,2] is achieved using Integrated Circuits (ICs), which are based on ultra small-scale integration and often contain entire system on chip [3]. Majority of these ICs are BGA type with hundreds of connectivity pins [4–6]. Unlike conventional ICs where the pins exist at the boundary of the IC, BGA components have the connectivity pins arranged as a grid of small soldering balls under the entire body of IC. A single printed circuit board (PCB) contains multiple ICs of BGA type. Presence of BGA components on PCBs and the smaller size of PCBs significantly increase the complexity of routing.

Routing in PCBs is the problem of connecting the pins of the PCB components that are required to be on same signal/voltage level for the circuit to be operational. Each pin pair that needs to be connected is called a net. A single PCB can have multiple of tens to a few thousand nets, depending upon the complexity of the electronic circuit. Ideally, the routing algorithm shall connect all nets in such a way that the nets do not overlap (graph emerging from nets shall be planar) and the nets shall be entirely contained within the area of the PCB. In addition, the constraints like synchronous signaling and signal-power separation are also considered. Given the smaller sizes of PCBs, high density of components and thousands of nets to be routed, a single layer is not enough to route all nets in a planar fashion. Therefore, multiple routing layers are used as shown in Fig. 1. In this case, all nets being routed in a single layer must result in a planar graph.

The problem of PCB routing is a specific instance of the generic problem of constructing planar graph for a given set of nodes, which is known to be NP-Hard [7]. The complexity of the problem further increases because of BGA components. For BGA components, routing the desired pins from under the body to the boundary of the component is challenging. Therefore, PCB routing has been divided into two parts: (i) Escape routing where pin to component boundary route is established and (ii) Area routing where the net between the component boundaries is established. The two parts are shown in Fig. 1. Given the high number of pins of BGA components, it is generally desirable that pin escaping for multiple components is done simultaneously in a specific order to reduce the complexity of area routing. Ordered escaping of pins for multiple components is known as simultaneous escape routing (SER) [8–14]. Fig. 2 shows an example instance of SER. If a pin uses top most boundary point in one component for escape, then the same pin number of other component also uses the top most boundary point for escape to maintain the ordered escaping. In this way the area routing becomes very simple and it is just a matter of connectivity of straight wires without any crossing to maintain planarity.

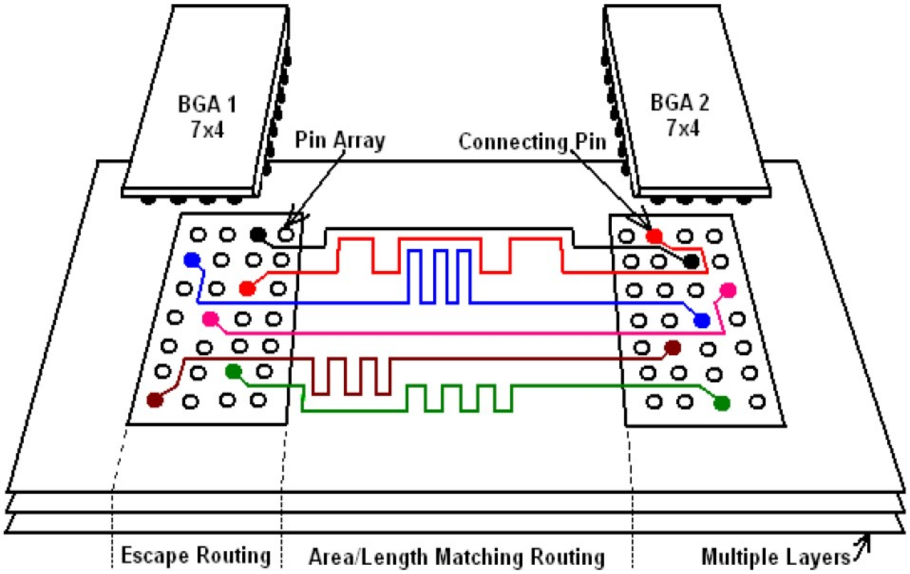


Fig.1. PCB Routing

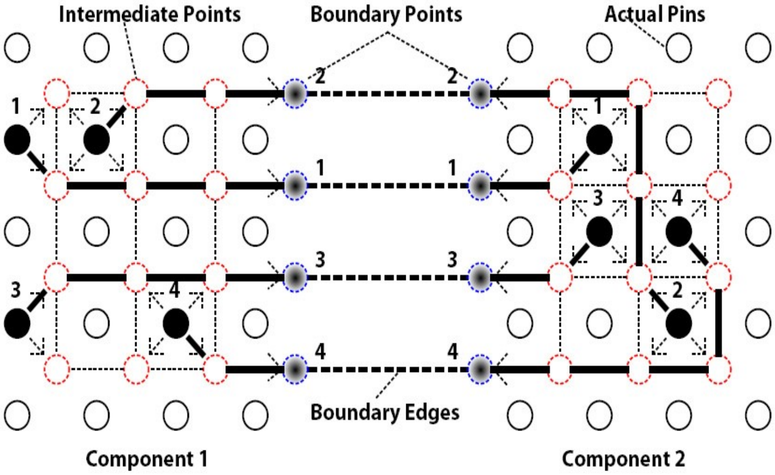


Fig.2. Simultaneous Escape Routing

Focus of this research is simultaneous escape routing. We propose an optimization model, which can be used to identify the net escape order for a given pair of components as well as establish the escape routes for the nets within two components. With the nets of two components escaped in order, the area routing becomes a trivial task if single side escaping is used. The objective of the model is to maximize the number of nets that can be connected for a given PCB and component placement. The objective is constrained by planar graph of connected nets, path length, power signal integrity and ordered escaping constraints. To the best of our knowledge, this is the first optimization model that solves the constraints, net escape ordering and routing simultaneously for SER problem. Comparative evaluation shows that the proposed optimization model can connect more number of nets in all tested examples when compared with the well-known commercial tools. The proposed optimization model is a contribution towards enhancing the capabilities of PCB designers and can be integrated into commercial software for use.

The rest of the document is organized as follows: In Section 2.0, we highlight some basic techniques used for planar routing and also discuss advancements in this area with latest literature review and the research gap. Section 3.0 presents mathematical model for maximal net routing problem for SER. In Section 4.0, we discuss the experimental setup and perform comparative evaluation of the proposed model. Section 5.0 concludes the paper.

2.0 RELATED WORK

PCB routing can be categorized into two main categories: escape routing and area routing [9]. The objective of escape routing is to establish the routes from the pins to the boundary of the component in an ordered or unordered way, based on constraints on available capacity between adjacent pins. With the escape routes established, the problem of PCB routing is reduced to conventional PCB routing problem of area routing [15, 16]. The objective of area routing is to establish routes between the boundaries of components based on constraints like length matching [17], timing and signal integrity.

Escape routing can be divided into three types [18]: (i) Unordered escape routing where pins of a single component are routed to the component boundary without considering any specific escape order of the pins [19–23]; (ii) Ordered escape routing where pins of one component are escaped in a pre-defined order. Bus escape, also referred as maximum disjoint subset problem [24] is another example of this type of routing; and (iii) Simultaneous escape routing where pins of two components are escaped simultaneously, ensuring that the nets of both components are escaped in same order. Focus of this paper is on SER and we present related literature in the next subsection in more detail.

2.1 Simultaneous Escape Routing

The initial work on SER considered unordered escaping of one component followed by ordered escape routing of second component to match the order of first component pins. However, this approach leaves a number of unconnected nets. The first well known research on SER was carried out by Ozdal et al. [9]. The authors used graph based approach to find maximal escape nets. This approach performs well if the pins of both components are closely aligned to each other. However, fixed escape patterns used in this approach cannot solve complex SER problems. This work has been extended by the same authors Ozdal et al. [10] and randomized algorithm for large scale problems has been proposed. However, the assumption that each net must be escaped monotonically in one direction, limits the optimized results and its applications. Ozdal et al. [11] used congestion driven router for routing pattern generation; however, the graph of escape patterns becomes complicated for large scale SER problems.

B-Escape is a routing algorithm proposed by Luo et al. [12]. The algorithm is based on greedy approach of boundary routing. In this algorithm, one net is routed at a time and the routing area is reduced by excluding the area having already used by the recently routed net. This algorithm performs well when compared with Allegro PCB router [12]; however, it uses heuristic approach based on dynamic net ordering and consumes more time to find a suitable net order. In addition, it also takes longer paths, which reduces the remaining space for escaping of nets and consumes more wire length. Simultaneous Pin Escape [14] is a flow model based approach. Kong et al. show that their approach guarantees planar routing if planar graph construction for the specific instance is possible. The existence of solution can be judged by the relation between maximum flow and number of escape pins of the BGA component [14,19]. If maximum flow is greater or equal to the escape pins, then this approach must provide planar solution; however, exact value of maximum flow is not known and we cannot ensure 'no net crossing', for higher inter pin capacities.

Another way of solving such problem is by making decision trees for all possible paths between each source destination pair of pins. However, decision trees can sometimes be quite difficult to comprehend when the tree size is too big [25]. Rout-ability Driven Net Order proposed by Yan et al [26] is a multi-step approach. An important step is identification of net ordering, which is derived based on planar bipartite graph theory. A subsequent step uses global and detailed routing technique under the constraints of planar routing and escape capacity. This approach improves the computation time by 54.1% as compared to Kong et al [14]. The drawback is that the second step is totally dependent on the output of first step i.e. net ordering. It does not explore all possible paths and may lead to sub-optimal results in complex problem instances. Furthermore, it is not possible to integrate important constraints like signal and power integrity and length matching. Recently researchers are using constraint based optimization models in many applications, from high crowd density facilities [27] to mobile cloud computing [28], to solve such kind of problems. These modelers find optimal result by exploring all possible areas where solution can exist by considering all the given constraints.

Most of the research on SER is based on heuristic approaches with basic constraints on capacity and planar graph construction. In this research, we consider optimization theoretic approach. Instead of solving the problem in parts to get sub-optimal solution, this optimization approach is expected to produce optimal results. This approach not only considers capacity and planarity but also considers constraint of power-signal integrity, net route length and net escape order altogether, to solve the SER problem at the cost of higher computational time. However, with high speed computing units and clustering of computational resources to perform complex computational tasks, the drawback can easily be overcome.

3.0 PROBLEM FORMULATION AND OPTIMIZATION MODEL

In this section, we first define the related terminology and flow models for different capacities and then formulate the SER as planar bipartite graph construction problem. Subsequently, we propose an optimization model that can construct maximum routes for the nets under given set of constraints. We first define the related terminology.

Escape Boundary: A virtual line is assumed to exist outside the actual boundary of component on the side (sides) that are to be used for escape routing of nets, for the pair of components. This virtual line is known as escape boundary. Where, escape routing is the problem of finding the routes from connectivity pins of the component to the points on escape boundary.

Inter-pin Escape Capacity: Inter-pin escape capacity is the number of wires (nets) that can pass between adjacent pins of the BGA component. There are three types of capacities i.e. horizontal, vertical and diagonal between adjacent pins as shown in Fig. 3. Collectively we can notate it as $IP_C(h_{cap},v_{cap},d_{cap})$. This capacity is constrained by inter-pin gap, wire thickness necessary for error free signal propagation and gap between adjacent wires that is necessary to avoid cross talk or coupling effect.

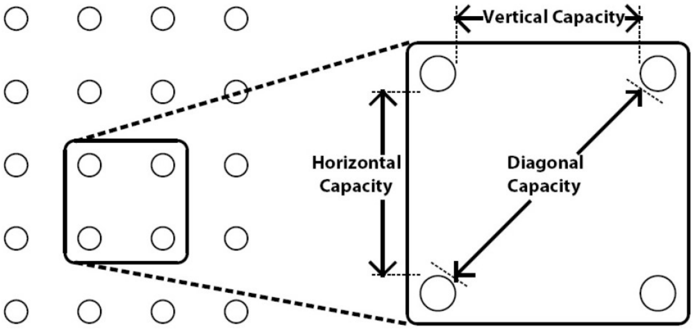


Fig.3. BGA Tile showing capacities

Intermediate Points: Intermediate points are the points between the adjacent pins of the BGA component. For a set of four adjacent pins, 1, 5, 13 intermediate points can exist, depending upon inter-pin capacity of IP_C = 1, 2, 3 and is shown in Fig. 4(a), 5(a) and 6(a) respectively. Adjacent intermediate points can be connected at later stage to form links and wires that can be used for routing the nets.

Escape Boundary Points: These points exist on the escape boundary. There are two types of escape boundary points. (i) Adjacent to every connectivity pin of the component that is on the component boundary (towards escape boundary) and needs to be connected. (ii) Escape boundary points equal to the inter-pin capacity may be created at equal distance on the escape boundary between every pair of component boundary pins. For IP_C=1, it is shown in Fig. 7(b).

Potential Routing Edges: There are five types of potential edges: (i) Edges between Intermediate points and adjacent connectivity pins that need routing, (ii) Edges between adjacent intermediate points, (iii) Edges between intermediate points and adjacent escape boundary points, (iv) Edges between component boundary pins and escape boundary points and (v) Edges between the boundary points of both components. The first four edges are highlighted with (i), (ii), (iii) and (iv) in the left component as shown in Fig. 7(b), where as the last type of edges are shown in Fig. 7(c). All edges can be diagonal, horizontal or vertical except the intermediate to escape boundary point edges and boundary pin to escape boundary point edges that can either be horizontal or vertical, depending upon the escape side. These edges become part of the wires that are used in routing. The edges of type (ii) and (v) are bidirectional whereas all other are unidirectional.

3.1 Problem Formulation

The initial component has a grid like structure of pins. We can formulate the problem by considering its chunk (called tile) consisting of four neighboring pins as shown in Fig. 3. We formulate the flow model (graph) for routing in a tile, and this can be easily extendable for the entire grid to find out complete escape routing solution. As we have already defined that capacities are of three types i.e. horizontal, vertical and diagonal between the adjacent pins of a tile, and we use the notation $IP_C\,(h_{cap},v_{cap},d_{cap})$ for inter-pin escape capacity. We simplify this capacity notation to IP_C, by assuming that h_{cap} and v_{cap} are same. Therefore, IP_C=1 implies that IP_C = (1,1,*x*). Where '*x*' is the d_{cap} and it varies according to the geometrical shape of the flow model and can be calculated by the equation, $x=2.(\,h_{cap}\,-1)+1$.

In order to formulate flow model for planar routing, we need intermediate points *'I'* (according to the capacity IP_C) and edges between them in a tile, such that the net (flow) using these edges must be disjoint and must not cross each other. Single flow model for higher capacities sometimes cannot ensure 'no net crossings' (planarity) or sometimes overburden with many intermediate points and edges. We propose separate flow models for different IP_C values, with exact required number of intermediate points. We show how 1, 5 and 13 intermediate points are exactly required for planar routing for IP_C =1, 2 and 3 respectively.

3.1.1 Flow model for IP_C =1

In this scenario, as we have discussed already, we are assuming that only one edge (net) can pass through between the two orthogonal and diagonal pins of a tile. The flow model for IP_C =1 is shown in Fig. 4(a). There is an intermediate point in the center of the tile, highlighted with dotted circle. For IP_C =1, this can be used only once either by any other neighboring intermediate point or by connecting pin of the tile as shown in Fig. 4(b) and Fig. 4(c) respectively. Dotted lines (edges) show the possible path, which can be used by any net to complete the escape routing towards the boundary of the BGA component. All edges from connecting pin to intermediate point are directional whereas all other edges are bi-directional.

Similarly if we want to increase the diagonal capacity to make IP_C = (1,1,2), then the central intermediate point of a tile is replaced by four intermediate points as shown in Fig. 4(d). The connecting pin is now can be connected to any one out of two possible intermediate points. Fig. 4(e) shows that how two nets can pass through now, with the increase in diagonal capacity whereas in Fig. 4(f), we shows that how one 'connecting pin net' and other 'passing through net' use the flow model.

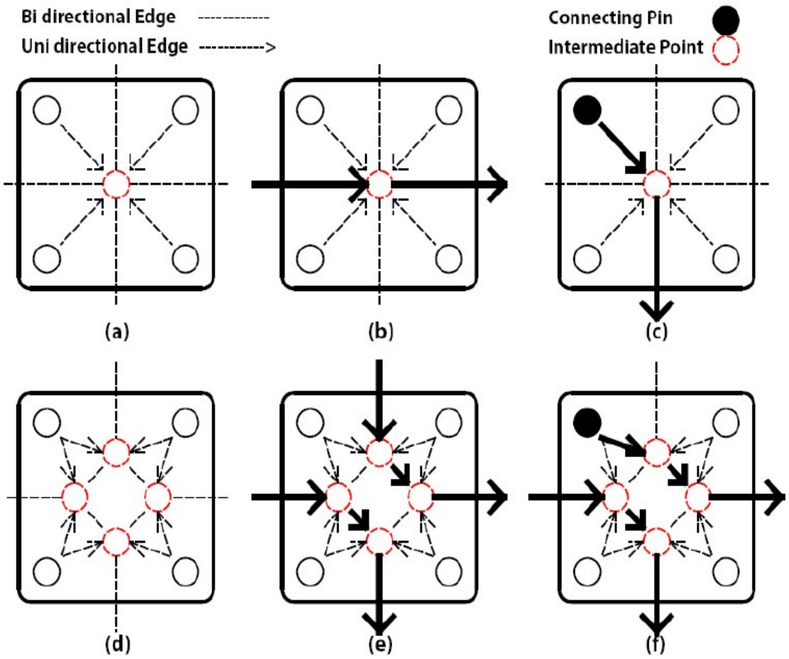

Bi directional Edge ------------
Uni directional Edge ----------->
Connecting Pin ●
Intermediate Point ○
(a)
(b)
(c)
(d)
(e)
(f)

Fig.4. Flow model for $IP_C = 1$

3.1.2 Flow model for IP_C =2

In this scenario, we show that two edges (nets) can pass through between the two horizontal and vertical pins of a tile. The flow model for $IP_C = 2$, is shown in Fig. 5(a). There are five intermediate points in the center of the tile, highlighted with dotted circle that ensures 'no net crossings'. By geometrical layout, possibly three nets can pass through diagonal space (i.e. d_{cap} or $x = 3$) and this turns the inter-pin capacity to $IP_C = (2,2,3)$, as shown in Fig. 5(b).

3.1.3 Flow model for IP_C =3

The last scenario, shows that three nets can pass through now between the two horizontal and vertical pins of a tile. The flow model for $IP_C = 3$, is shown in Fig. 6(a). There are now thirteen intermediate points in the tile to ensure 'no net crossings'. By geometrical layout, now it is possible by five nets to pass through at maximum, through diagonal space (i.e. d_{cap} or $x = 5$) and the inter-pin capacity notation now turns to $IP_C = (3,3,5)$ as shown in Fig. 6(b).

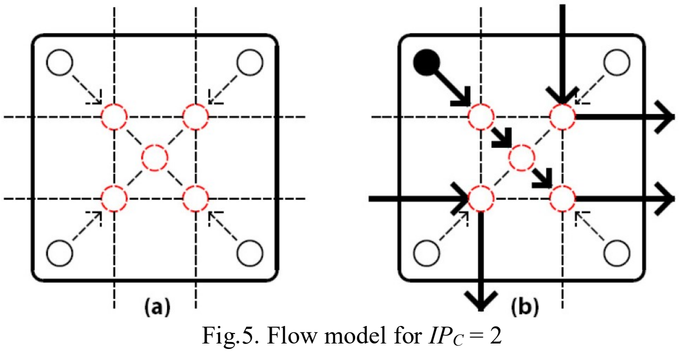


Fig.5. Flow model for $IP_C = 2$

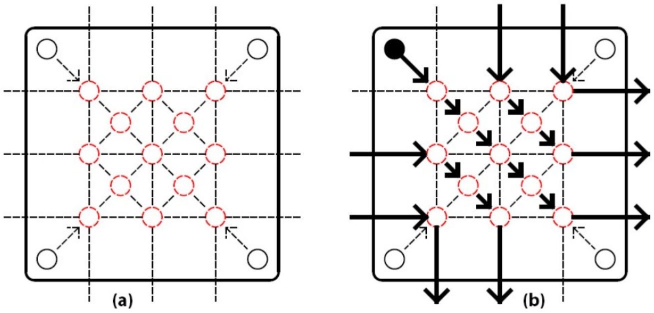


Fig.6. Flow model for $IP_C = 3$

3.2 SER as Planar Bipartite Graph Construction Problem

SER problem can be considered as a special case of planar bipartite graph construction problem. We demonstrate it with a simple example having two BGA components of grid size 5x4 as shown in Fig. 7(a). We also assume here that its single side escape highlighted with a dashed line boundary. Out of 20 pins, only 5 pins are connectivity pins in each component. We draw, intermediate points, escape boundary points for boundary connectivity pin and escape boundary points according to the flow model with IP_C =1, as shown in Fig. 7(b). Also, we draw all unidirectional and bidirectional potential edges and removes unnecessary actual pins, as shown in Fig. 7(c). This clearly shows that pins requiring planar connectivity and the possible connections represent the graph. Hence the problem is planer graph connectivity problem for which we are going to present optimization model.

We define escape routing problem as: Given set V of connectivity pins that need routing, set U of escape boundary points such that $|U| \geq |V|$, set I of intermediate points and set E' of potential routing edges, construct the bipartite graph $G(V, U, E)$ where:

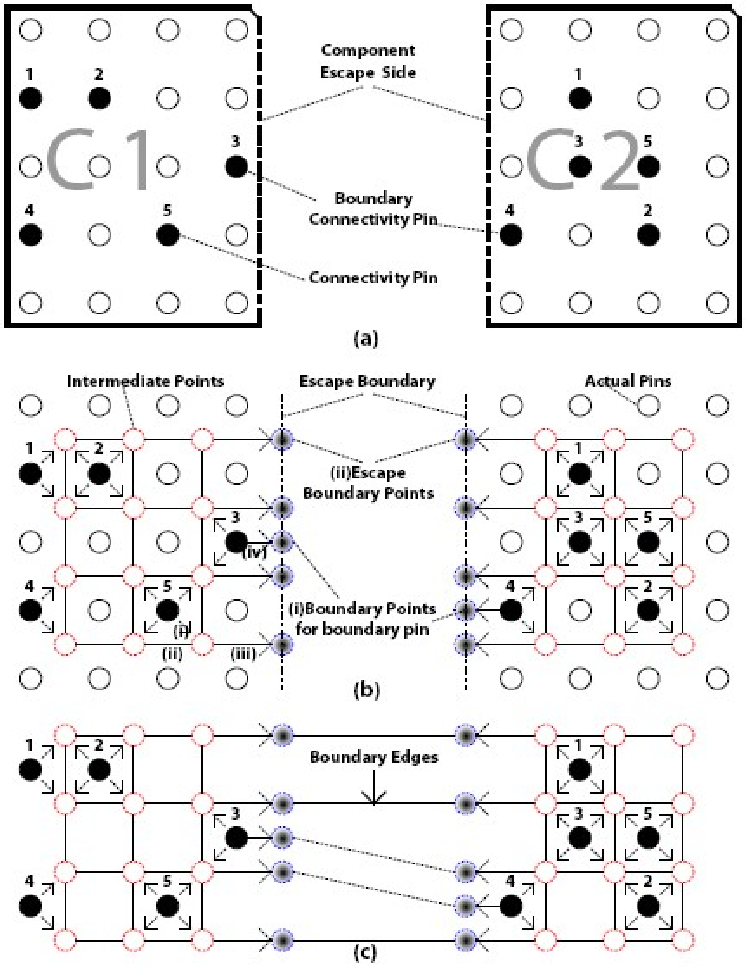
(i) E is set of paths comprising of $i \in I$ and $e \in E'$.

(ii) $\forall i \in I$ and $e \in E'$, i and e belong to only one path $p \in E$ at the most.

(iii) $\forall v \in V, Deg(v) = 1$ and $\forall u \in U, Deg(u) \leq 1$.

We map all the pins, points and potential edges of Fig. 7(c) into vertices and edges of planar bipartite graph as shown in Fig. 8(a). Set $V1$ & $V2$ consists of connectivity pins of component 1 and 2 respectively. Similarly, set $U1$ & $U2$ consists of escape boundary points of each component. The edges between the vertices of set $U1$ & $U2$ are single bidirectional edges, whereas the edges between the vertices of $V1$ & $U1$ and $V2$ & $U2$ consist of paths from a connectivity pin vertices of set $V1$ ($V2$) to boundary pin vertices of set $U1$ ($U2$). These paths must be disjoint with each other (i.e no intermediate vertex or potential edge is common when compared to any other path), to ensure planarity. Planar and maximal net routing requires proper net ordering (sequence of escaping) as shown in Fig. 8(b).

SER is escape routing of two components simultaneously such that the respective pins of both components are escaped in same order. Effectively, the SER comprises of two sub-problems that need to be solved simultaneously. First problem is to decide upon the order in which the nets are to be escaped. The second problem is escape routing of the two components in the desired order. It is possible to define the escape order before escape routing is performed. This significantly reduces the complexity; however, the solution space is also significantly reduced, resulting in fewer connected nets. We present a simple example that shows the significance of net ordering.



(a)

(b)

(c)

Fig.7. Graph construction using flow model with $IP_C = 1$

Consider Fig. 9(a), which shows simple 5x4 grid BGA components that requires 4-pin connectivity using single side escaping. Suppose that the component-1 necessitates the order of escaping 1,2,3,4, as shown in the Fig. 9(a). It is obvious to see that even for this trivial example, the escaping in desired order is not possible for component-2 for all its pins. However, if the order of escaping is changed to 1,2,4,3, the nets can be escaped easily as shown in Fig. 9(b). Hence the problem of identifying a feasible order is non-trivial and requires to be simultaneously solved with the escape routing problem.

We now propose the optimization model that takes as input, the set V of connectivity pins, set U of boundary points, set I of intermediate points and set E' of edges for the components, and returns a planar connected graph with maximum possible nets of the components connected under given set of constraints. For any of the two given components C_i and C_j, the optimization model simultaneously decides on the net order for escaping and constructs two bipartite graphs $G_i(V_i, U_i, E'_i)$ and $G_j(V_j, U_j, E'_j)$.

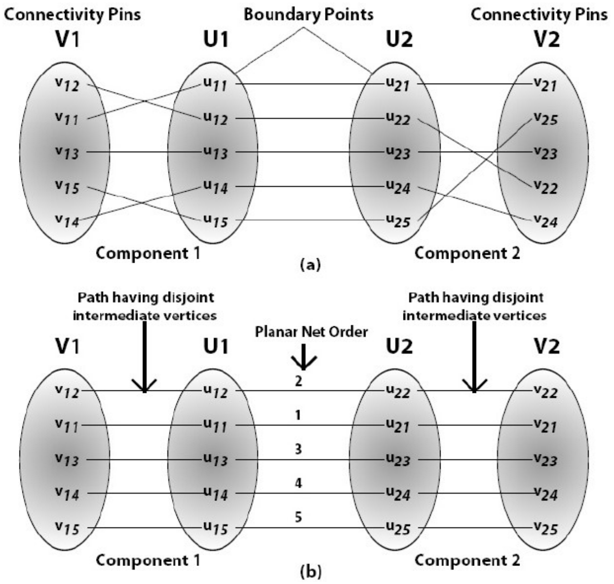


Fig.8. Planar Bipartite Graph Construction

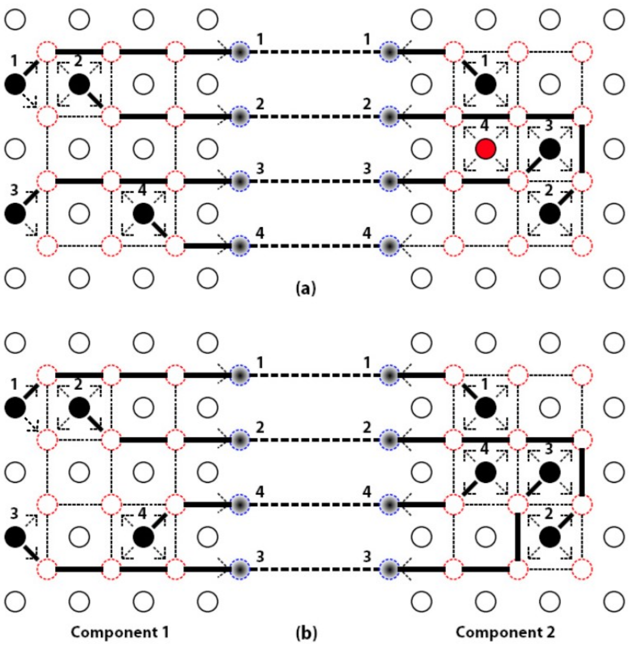


Fig.9. Net Ordering Significance

3.3 SER Model

We formulate a linear programming model that can perform simultaneous routing of nets for multiple components. Let set $C = \{c_1, c_2, \ldots, c_n\}$ be the set of n BGA components for which the routing is to be performed. As explained in the previous section, let Vc_i be the set of connectivity pins of component c_i that need routing. Let Ic_i and Bc_i be the sets of intermediate points and boundary points respectively of component c_i. The model takes following sets as input:

Set $V_{all} = \cup c_i\ (Vc_i \cup Ic_i \cup Bc_i)$, the set of all pins and points of all components.
Set $I_{all} = \cup c_i\ (Ic_i \cup Bc_i)$, the set of all intermediate and boundary points of all components.

Set $E'_{all} = \cup c_i \, E'c_i$, the set of all possible edges of all components. The edges are identified by their end points as e_{ij} where $i, j \in V_{all}$. Note that the edges between pins to intermediate points and intermediate points to escape boundary points are unidirectional while all other edges are bidirectional. Unidirectional edges make modeling easy but do not affect the routing outcome. The inter pin capacity is maintained with the number of intermediate points/edges according to the flow model, which satisfies the capacity constraint. If there are only 2 edges between adjacent pins, then maximum 2 nets can pass through.

Set $BE_{all} = \cup c_i \, BEc_i$, the set of all boundary edges of all components.

Set $N \ = \{(v_i,v_j)| \ v_i \in Vc_i, \ v_j \in Vcj\}$, The net list (pin pairs that need to be connected)

Set $NE[e_{ij}]$ be the set of neighboring edges, where $e_{ij} \in BEc_i$. Each set consists of all the neighboring boundary edges of $e_{ij} \in BEc_i$. In addition, the model takes the set T of the nets that have limit on the route length. The set consists of the ordered pair where first element is the net while second element is the length threshold given in terms of number of edges.

The model also takes the set P of the pair of nets that cannot escape together (from adjacent boundary points). This maintains the power and signal integrity for error free routing. Here, we are considering this constraint only at the boundary edges but it can be extended within the component by making sets of neighbors for all the edges. The output of the model is number of nets connected and the edges used to create the routes for the nets.

We define the decision variable $X_{(r_{ab},e_{ij})}$ of the model for all $r_{ab} \in N$, $e_{ij} \in E'_{all}$ as a binary variable. The variable acquires the value one if edge e_{ij} is used for the routing of net r_{ab}, otherwise, the value of the variable is zero.

$$X_{(r_{ab},e_{ij})} = \left\{ \begin{matrix} 1 & \textit{if edge } e_{ij} \textit{ is used for routing } r_{ab} \\ 0 & \textit{otherwise} \end{matrix} \right\}$$

The routing is constrained by connectivity constraints, planar graph constraints, net order constraints and system constraints. The constraints of each type are explained below.

3.3.1 Connectivity Constraints

This set of constraints defines basic routing of the nets. The constraints ensure that the two pins of any net are connected using the edges from set E'_{all} and there is exactly one route between the pins of every net.

Single Path Activation Constraint: Multiple edges are incident on the connectivity pins. For every connectivity pin that needs routing, at the most one incident edge can be used for the routing of a particular net. Since the edges from connectivity pins are unidirectional towards the intermediate points only, therefore we need to apply this constraint only for the edges going outside from pins in each pair of component for that net. This constraint allows no net routing for the connectivity pins if the complete routing path construction in a planar fashion is not possible.

$$\sum_{e_{ij} \in E'_{all} \,\&\, i=(a \ OR \ b)} X_{(r_{ab},e_{ij})} \leq 1 \qquad \forall r_{ab} \in N$$

Net Ends Match Constraint: The two ends of every net should have same connectivity state. This means that either both connectivity pins of a net should use one incident edge for the particular net or both pins should not use any net. This constraint combined with the subsequent constraints ensures that there are no incomplete routes. Either the complete route exists between pins of the net or no edge is used for that net.

$$\sum_{e_{ij} \in E'_{all} \,\&\, (i=a)} X_{(r_{ab},e_{ij})} = \sum_{e_{pq} \in E'_{all} \,\&\, (p=b)} X_{(r_{ab},e_{pq})}$$

$$\forall r_{ab} \in N$$

Route Completion Constraints: Above two constraints deal with the connectivity pins. If the routing of net is initiated for one pin, the other pin also needs to be connected. Furthermore, the constraints ensure that only one route is initiated for every net. This constraint deals with the intermediate and boundary points. The following two constraints ensure that a route is completed if it is initiated at the connectivity pins. These constraints also ensure that there is no unwanted and unconnected route that does not belong to any net. Following constraint ensures that if one incident edge on an intermediate point connects the point to one of the connectivity pins of a particular net (directly or through a path of edges), then another edge must be active for the same net to allow the connectivity to the second connectivity pin of the same net. This constraint also ensures that a route cannot initiate from an intermediate point, without being connected to any connectivity pin.

$$\sum_{e_{ij} \in E'_{all}} X_{(r_{ab}, e_{ij})} = \sum_{e_{jk} \in E'_{all}} X_{(r_{ab}, e_{jk})}$$

$$\forall r_{ab} \in N, \forall j \in I_{all}$$

The above constraint ensures that even number of edges are used for any net at any intermediate point. However, it does not eliminate the possibility that the same edge is used twice. If the same edge is used twice, the path construction stops at the intermediate point and the two pins of the net are not connected. Following constraint ensures that different edges are selected for any net at each intermediate point.

$$X_{(r_{ab}, e_{ij})} + X_{(r_{ab}, e_{ji})} \le 1$$

$$\forall r_{ab} \in N, \forall e_{ij}, e_{ji} \in E'_{all}$$

3.3.2 Planar Graph Constraint

The above set of constraints results in route-able nets. However, the resultant connectivity graph is not planar. Although all edges are used by at the most one net, the routes for different nets can intersect at intermediate or boundary points. Introducing a simple constraint can eliminate this situation. The constraint ensures that any point on the PCB is used by at the most one net.

$$\sum_{r_{ab} \in N} \sum_{e_{ij} \in E'_{all}} X_{(r_{ab}, e_{ij})} \le 1, \quad \forall j \in I_{all}$$

3.3.3 Net Order Constraint

All the escaped nets between any two components must be in same order to ensure SER. Actually, for a net r_{ab}, if pin *'a'* belongs to component C_i and pin *'b'* belongs to component C_j, then all the escaped nets between C_i and C_j must be in the same order. This constraint also acts as a global routing (to find the escape boundary points for nets) which forces the other constraints to find detailed routing (how to reach at the boundary points from connectivity pins) in that order to find maximal routing solution. Based on global routing, the model tries to find the path from connectivity pin to the escape boundary point by using the route completion constraints. Other constraints help in finding single path and planarity during detailed routing.

$$X_{(r_{ab}, e_{ij})} = X_{(r_{ab}, e_{ji})}$$

$$\forall r_{ab} \in N, \forall e_{ij}, e_{ji} \in BE_{all}$$

3.3.4 System Constraints

It is possible to add a variety of system constraints in the optimization model. For example, the model can be constrained to use equal length paths for a set of nets to ensure the synchronous signaling. Similarly, it is preferred that the power signal wires (nets) should not be adjacent to the data and control signal wires, specifically if the wire width is less. Presence of such wires adjacent to one another can result in signal errors. In this paper, we add these two constraints into the model as an example.

Path Length Constraint: Path lengths of certain nets can be restricted to a threshold value, allowing timing budget requirements to be incorporated. Following constraint limits the path length of the nets given in a set to the respective threshold.

$$\sum_{e_{ij} \in E'_{all}} X_{(r_{ab},e_{ij})} \leq t_{r_{ab}}, \quad \forall r_{ab} \in T$$

Power Signal Integrity Constraint: This constraint ensures that the specific pair of nets must not escape through adjacent boundary points. Keeping the safe distance between them, allows to transfer error free signals. Following constraint checks that if one boundary edge is active for one net then the second net from the pair must not be active on any of its adjacent neighboring boundary edges.

$$X_{(r_{ab},e_{ij})} + \sum_{e_{kl} \in NE[e_{ij}]} X_{(r_{cd},e_{kl})} \leq 1$$

$$\forall (r_{ab}, r_{cd}) \in P, \forall e_{ij} \in BE_{all}$$

3.3.5 Objective Function

The objective of this model is to maximize the number of nets that can be routed. Here it is noted that objective is not maximizing the links, so it does not choose longest paths. However in order to increase the number of routable nets the model tries to select shorter paths. Following expression ensures the objective:

$$\max . \sum_{r_{ab} \in N} \sum_{e_{ij} \in E'_{all}} X_{(r_{ab},e_{ij})}$$

The complete optimization model is presented in equation set 1 - 9.

$$\max . \sum_{r_{ab} \in N} \sum_{e_{ij} \in E'_{all}} X_{(r_{ab},e_{ij})} \tag{1}$$

subject to:

$$\sum_{e_{ij} \in E'_{all} \,\&\, i=(a \ OR \ b)} X_{(r_{ab},e_{ij})} \leq 1 \qquad \forall r_{ab} \in N \tag{2}$$

$$\sum_{e_{ij} \in E'_{all} \,\&\, (i=a)} X_{(r_{ab},e_{ij})} = \sum_{e_{pq} \in E'_{all} \,\&\, (p=b)} X_{(r_{ab},e_{pq})} \qquad \forall r_{ab} \in N \tag{3}$$

$$\sum_{e_{ij} \in E'_{all}} X_{(r_{ab},e_{ij})} = \sum_{e_{jk} \in E'_{all}} X_{(r_{ab},e_{jk})} \qquad \forall r_{ab} \in N, \forall j \in I_{all} \tag{4}$$

$$X_{(r_{ab},e_{ij})} + X_{(r_{ab},e_{ji})} \leq 1 \qquad \forall r_{ab} \in N, \forall e_{ij}, e_{ji} \in E'_{all} \tag{5}$$

$$\sum_{r_{ab} \in N} \sum_{e_{ij} \in E'_{all}} X_{(r_{ab},e_{ij})} \leq 1, \qquad \forall j \in I_{all} \tag{6}$$

$$X_{(r_{ab},e_{ij})} = X_{(r_{ab},e_{ji})} \qquad \forall r_{ab} \in N, \forall e_{ij}, e_{ji} \in BE_{all} \tag{7}$$

$$\sum_{e_{ij} \in E'_{all}} X_{(r_{ab},e_{ij})} \leq t_{r_{ab}}, \qquad \forall r_{ab} \in T \tag{8}$$

$$X_{(r_{ab},e_{ij})} + \sum_{e_{kl}\in NE[e_{ij}]} X_{(r_{cd},e_{kl})} \le 1 \qquad \forall (r_{ab},r_{cd}) \in P, \forall e_{ij} \in BE_{all} \tag{9}$$

Optimization models are generally NP-hard. However, for specific problem instance and the use of available solvers, the models can produce optimal results. In the following section, we solve the proposed optimization model using commercially available solvers for the example instances and compare the output results with that of commercially available PCB routing software.

4.0 EVALUATION

In this section, we first validate the optimization model proposed in Section 3.0. Subsequently, we present the performance comparison of the proposed model and Proteus auto router by Labcenter Electronics Ltd. We have used AMPL language to implement the model and AMPL modeler [30] available at NEOS server [29] to get the optimization model results. The comparison with the state of the art research work was not possible because of the unavailability of the examples being tested. The authors have contacted multiple researchers in the domain; however, none of the examples presented in the literature were made available, primarily because of proprietary designs.

4.1 Model Validation

In this section we verify that the route-ability, planar graph construction and net ordering constraints are able to achieve the desired routing for the given instances. For this purpose we test the model using the example component available in Proteus as BGA11_49_1.27. This BGA has 7x7 grid consisting of 49 pins as shown in Fig. 10. We took two components and randomly selected 6 pins from each component to be connected to form 6 nets. For better understanding, we have assumed inter-pin capacity of one (i.e. IP_C=1) in all examples and single side escaping is permitted for both components.

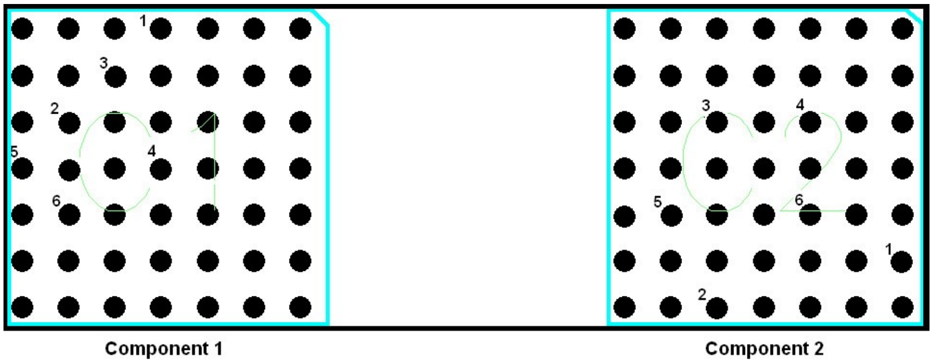


Fig.10. Pins position on BGA11_49_1.27

The example shows that the proposed model is able to perform SER for BGA based PCB components as shown in Fig. 11. The BGA has maximum 6 boundary points in each component and the model routes all the 6 nets by escaping in the same order from both components. The net order it chooses in a sequence 1,2,3,4,5 and 6. Also we can see that it is not using any edge or vertex twice, to ensure planarity. This example validates our constraints of connectivity, planar graph and net order.

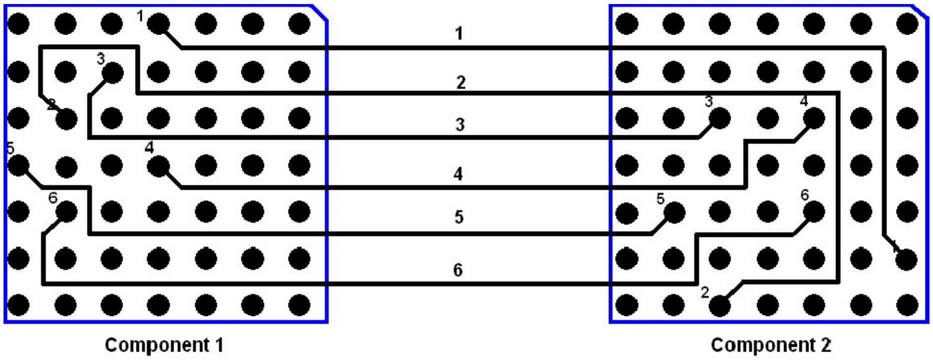


Fig.11. SER Routing with proposed model

System Constraints Validation:

We consider the same example to validate the system constraints. The model takes the set *P* of the pair of nets (1 and 2) that cannot escape together. We enable the `power and signal integrity' constraint and run the model once again, keeping all other things and constraints be the same. Again it routes all the nets in a planar fashion but with different net order. The order it follows this time is 1,3,2,4,5 and 6 as shown in Fig.12. This new order shows that model performs exactly as required, by keeping away the net 1 and 2. Hence it validates this constraint of our optimization model.

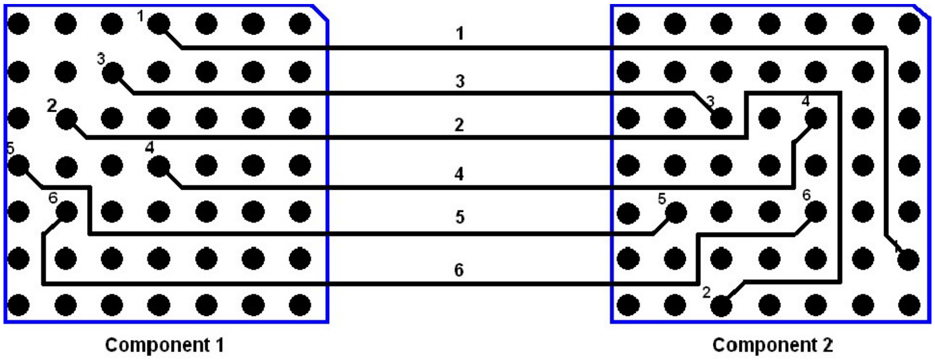


Fig.12. System Constraints effect on Routing

Similarly, we validate the `Path length' constraint. The path length of net 3 is 11 edges, as shown in Fig. 11. In component 1 we have 7 edges from pin to boundary point, whereas in component 2 we have 3 edges and one edge is between both boundary points. We restrict the length of net 3 to a threshold value of 9 (shortest path) edges. Therefore by using the proposed model, we are not only getting the optimized shorter routes but also can control the wire length, which is not possible as compare to other algorithms such as boundary escape routing [12]. Now we disable the `power and signal integrity' constraint and enable the `Path length' constraint, to see the effect of this constraint on the original routing output. Once again, the model not only route net 3 on its shortest path with path length of 9 edges, but also routes all other nets as usual in a planar way, as shown in Fig. 12. Hence, we verify that the route-ability, planar graph construction, net ordering constraints and system constraints are able to achieve the desired SER solution. In the following section, we compare the performance of the model against Proteus.

4.2 Performance Analysis

The performance comparison in this section is based on a single metric of number of route-able nets keeping in view the fact that the design constraints are met. Although the execution time is also important, as long as all nets can be routed within an acceptable time, the significance of execution time decreases. We have considered three different examples for the purpose of evaluation. Each example highlights different aspect of the proposed model as explained in respective section.

4.2.1 BGA11_49_1.27 Example

To validate the proposed model, we compare the results obtained by our proposed analytical model with the results obtained by Proteus Software. Proteus placed 5/6 nets even if we run interactively for 10 times for stable results. The routing of nets is shown in Fig. 13. It could not provide routing for 5th net. With Proteus, the only solution here to route this net, is manual routing, which is very difficult especially when we do routing for large number of pins and nets. Whereas our model, with the input of the same example, results having routing of all 6/6 nets as shown in Fig. 11.

4.2.2 BGA_18x9_17x6 Example

We test our SER optimization model on a two different grid size components. One BGA has 18x9 grid size while other has 17x6 grid size, as shown in Fig. 14. In this example we test two things: i) model testing for different size BGA's and ii) model testing for boundary connectivity pins. We randomly selected 2 boundary connectivity pins and 16 others from each component. Left component has 17 inter pin capacity boundary points and 2 boundary connectivity pins, results in 19 escape boundary points in total. Right component has 16 inter pin capacity boundary points and 2 boundary connectivity pins, results in 18 escape boundary points in total. Therefore for SER, there are 18 maximum possible nets that can be escaped, simultaneously from both components.

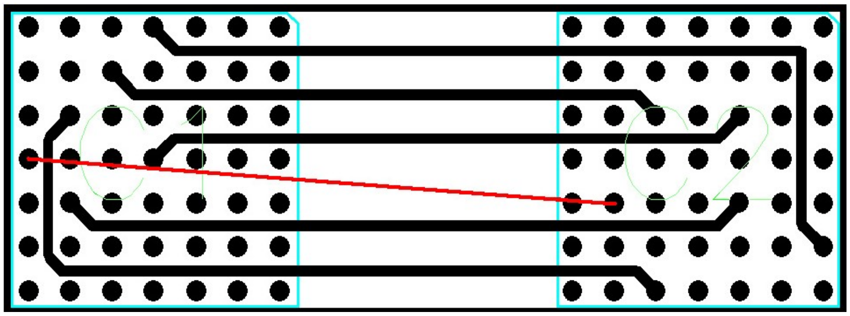


Fig.13. SER Routing with Proteus

We compare the results obtained by our proposed model with the results obtained by Proteus. In first run, Proteus placed 12/18 nets and after 10 runs it placed 15/18 nets as shown in Fig. 14. It could not provide routing for 11th, 12th and 16th net. We can see once again that manual routing is very difficult in this case. However when we run our model for this example, we get results having routing of all 18/18 nets as shown in Fig. 15.

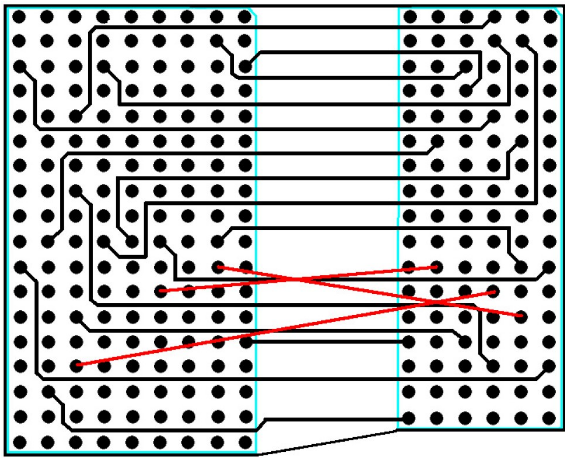


Fig.14. Proteus SER Routing for different size BGA's

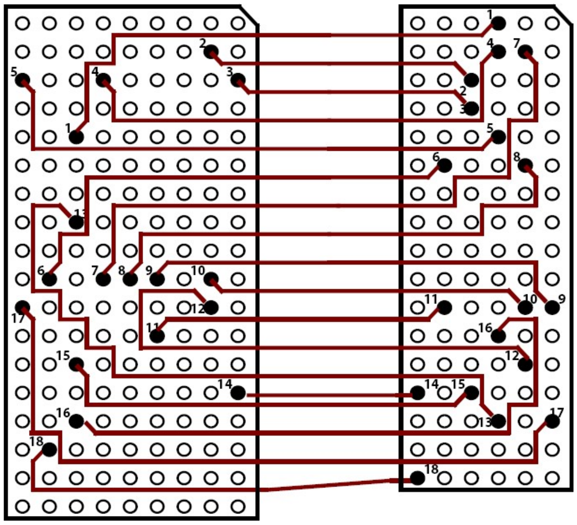


Fig.15. Model SER Routing for different size BGA's

4.2.3 BGA_98x49_86x50 Example

We also validate our SER optimization model on a relatively bigger example, which requires connectivity of 100 nets. Among two different grid size components, one BGA has 98x49 grid size while other has 86x50 grid size, as shown in Fig. 16. In this example we test two things i) what is the behavior of model when we have large number of nets ii) difference of routed nets with increased complexity. We randomly selected 100 pins from each component. Left component has 97 inter pin capacity boundary points and 19 boundary connectivity pins, results in 116 escape boundary points in total. Right component has 85 inter pin capacity boundary points and 15 boundary connectivity pins, results in 100 escape boundary points in total. Therefore for SER, there are 100 maximum possible nets that can be escaped, simultaneously from both components.

We compare the results obtained by our proposed model with the results obtained by Proteus. In first run, Proteus placed 65/100 nets as shown in Fig. 16. Even after 10 runs (stable results), it placed 82/100 nets. It could not provide routing for 18 nets and the manual routing is almost impossible in this case. However, when we run this example with our model, again we get results having routing of all 100/100 nets with same escaping net order and without violating any design rule.

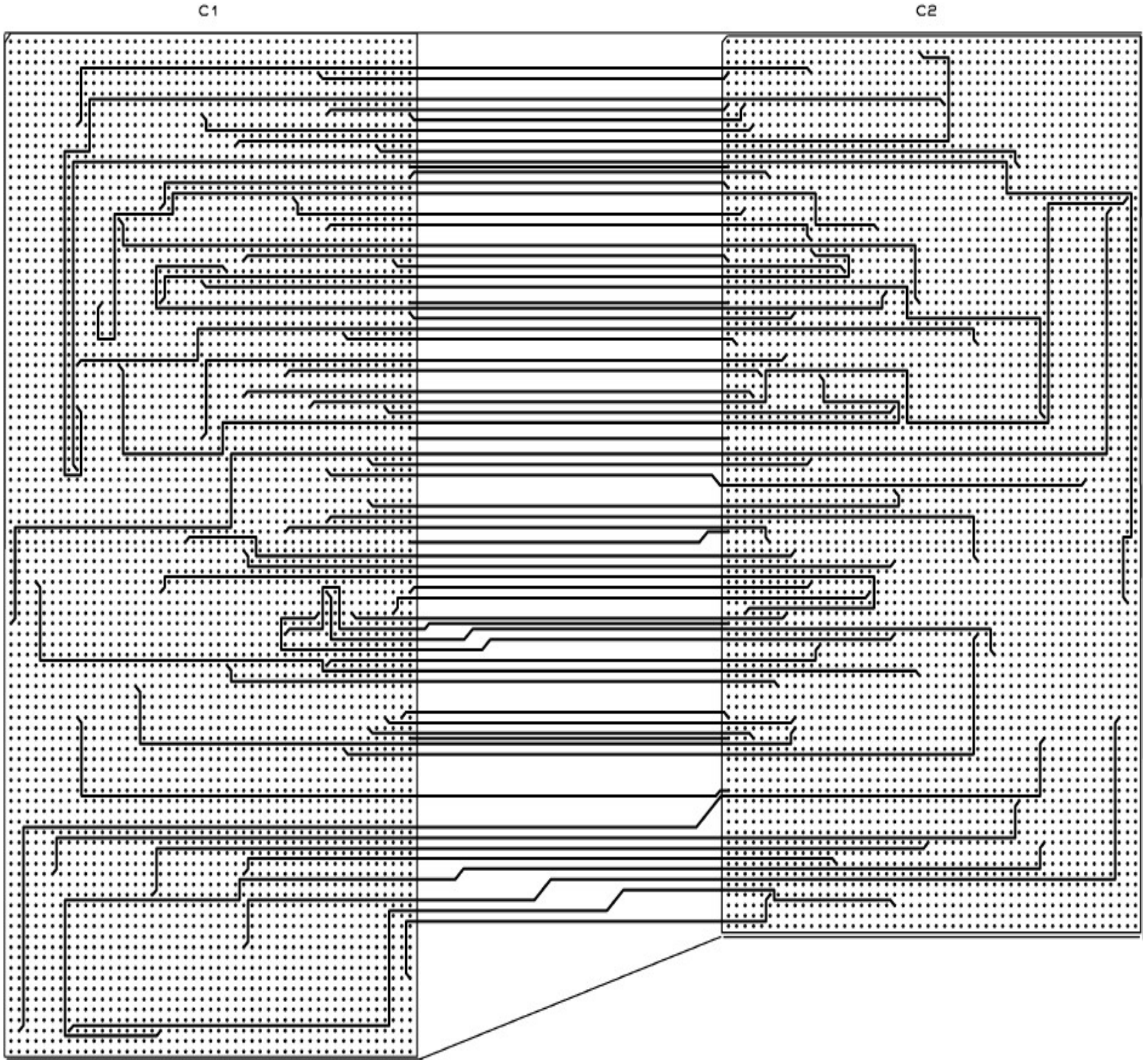


Fig.16. Proteus SER Routing for 100 nets BGA's

Here, we compare the results achieved by different experiments discussed so far. Our proposed optimization model performed 100% routing in all the five examples as shown in Table 1, however Proteus could not route all nets in experiment 2, 3, 4 and 5. Here it is important to note that manual routing of even for a single left behind net is very difficult, and needs lots of time in rearranging of all the nets once again. It is also clear from the results that as the grid size increases or number of required nets increases, the routing performance by Proteus decreases and left behind nets increases as shown in Fig. 17. However proposed model outperforms in all cases even for larger examples. The running times of larger examples are significantly high for model but we can overcome this deficiency by using high computing systems. The main important thing is routing of all the nets or maximal possible nets to avoid very tedious task of manual routing of left behind nets. Also, Proteus multiple passes produce different results in all examples. This indicates that a heuristic is being used which produces different results with different starting point, and we cannot confirm that the results are optimal, whereas proposed model produces same results all the time, in all experiments.

Table 1: Experiment results and comparison with Proteus

Ex. No.	C1 Size	C2 Size	No. of Nets	Proteus (Routing)			Model	
				1st Run	10th Run	%age	Routing	%age
1	5x4	5x4	5	4/5	5/5	100%	5/5	100%
2	7x7	7x7	6	4/6	5/6	83%	6/6	100%
3	18x9	17x6	18	12/18	15/18	83%	18/18	100%
4	48x40	44x44	50	36/50	41/50	82%	50/50	100%
5	98x49	86x50	100	65/100	82/100	82%	100/100	100%

It is obvious that the proposed optimization model is able to route upto 20% more nets compared to best case routing from Proteus. We observe that the execution time for the proposed model is relatively higher compared to the execution time for Proteus on same instance. Optimization model with improved execution time remains as a future development of the proposed work.

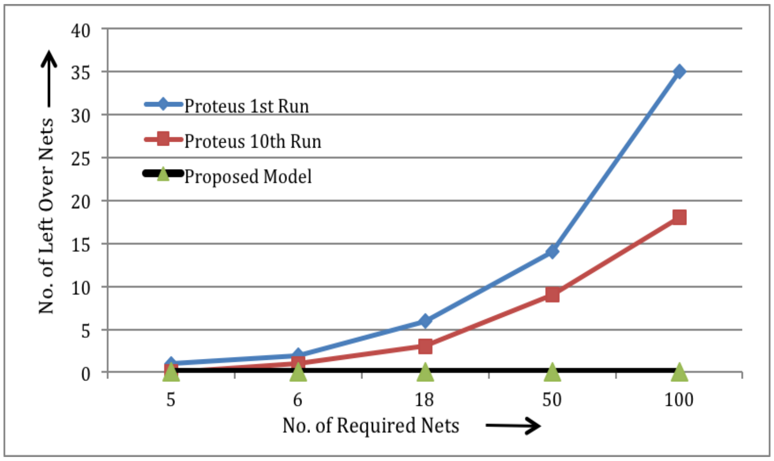


Fig.17. Model Comparison with Proteus

5.0 CONCLUSION

This research has proposed the use of optimization models for solving the problem of Simultaneous escape routing of multiple components. The problem of SER has been mapped onto planar bipartite graph construction problem using disjoint node and edge paths. The mapping enables the mathematical analysis of the SER problem using graph theory, which is the future work of current research. Flow models for different inter pin capacities has been proposed which helps in finding the diagonal capacities and disjoint paths to ensure planarity. Also mainly an optimization model has been proposed for planar bipartite graph construction problem and has been validated to solve the SER problem. To the best of our knowledge, this is the first work which uses optimization model to solve the problem of SER by considering all necessary constraints. Comparison with commercially available software shows that the optimization model can route upto 20% extra nets. In the future, we plan to derive minimum requirements in terms of inter-pin capacity, escaping sides and number of escape points to achieve planar simultaneous escape routing. These properties shall be used to propose time efficient optimization model. Subject to the availability of benchmark examples, we also plan to test the performance of the proposed model over benchmark examples.

REFERENCES

[1] M. M. Ozdal, “Routing algorithms for high performance VLSI packaging,” *PhD thesis*, University of Illinois at Urbana Champaign, 2005.

[2] J. C. Whitaker, *Microelectronics*, 2nd ed., CRC Press, 2005.

[3] L. A. Coldren, S. W. Corzine, and M. L. Mashanovitch, *Diode lasers and photonic integrated circuits*, John Wiley and Sons, 2012.

[4] G. Marcantonio, “Ball grid array (BGA) integrated circuit packages,” U.S. Patent 5,796,170, issued August 18, 1998.

[5] D. Y. Chong, B. Lim, K. J. Rebibis, S. Pan, S. Krishnamoorthi, R. Kapoor, A. Y. Sun, and H. Tan. “Development of a new improved high performance flip chip BGA package,” *In Proc. Electronic Components and Technology Conference*, vol. 2, pp. 1174-1180, 2004.

[6] R. Plieninger, M. Dittes, and K. Pressel. “Modern IC packaging trends and their reliability implications,” *Microelectronics Reliability 46*, No. 9, 2006.

[7] M. R. Garey and D. S. Johnson, *Computer and Intractability: a Guide to the Theory of NP-Completeness,* WH Freeman and company, New York, 1979.

[8] H. Xiang, X. Tang, and D. E. Wong, “An algorithm for simultaneous pin assignment and routing,” *in Proc. IEEE/ACM International conference on Computer-Aided Design*, pp. 232-238, 2001.

[9] M. M. Ozdal, and M. D. Wong, “Simultaneous escape routing and layer assignment for dense PCBs,” *in Proc. IEEE/ACM International Conference on Computer-Aided Design*, pp. 822-829, 2004.

[10] M. M. Ozdal, and M. D. Wong, “Algorithms for simultaneous escape routing and layer assignment of dense PCBs,” *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 8, pp. 1510-1522, 2006.

[11] M. M. Ozdal, and M. D. Wong, and P. S. Honsinger, “Simultaneous escape-routing algorithms for via minimization of high speed boards,” *IEEE Trans.Computer Aided Design of Integrated Circuits and Systems*, vol. 27, no. 1, pp 84-95, 2008.

[12] L. Luo, T. Yan, Q. Ma, M. D. Wong, and T. Shibuya, “B-Escape: A simultaneous escape routing algorithm based on boundary routing,” *in Proc. ACM 19th International Symposium on Physical Design*, pp. 19-25, 2010.

[13] Q. Ma, T. Yan, and M. D. Wong. “A negotiated congestion based router for simultaneous escape routing,” *in Proc. 11th International Symposium on Quality Electronic Design (ISQED)*, pp. 606-610, 2010.

[14] H. Kong, T. Yan, and M. D. Wong. “Optimal simultaneous pin assignment and escape routing for dense PCBs,” *in Proc. 15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 275-280, 2010.

[15] M. Tanaka, and Y. Nakagiri, “An approach to pin assignment in printed circuit board design,” *ACM SIGDA Newsletter 10*, no. 2, pp. 21-33, 1980.

[16] H. Brady, “An approach to topological pin assignment,” *IEEE Trans. Computer Aided Design of Integrated Circuits and Systems*, vol. 3, no. 3, pp. 250-255, July 1984.

[17] T. Y. Tsai, R. J. Lee, C. Y. Chin, C. Y. Kuan, H. M. Chen, and Y. Kajitani, “On routing fixed escaped boundary pins for high speed boards,” *In Proc. IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE),* pp. 1-6, 2011.

[18] Y. Tan, Q. Ma, and M. D. Wong, “Advances in PCB routing,” *IPSJ Trans. System LSI Design Methodology*, vol. 5, Feb. 2012.

[19] M. A. Shayegan, S. Aghabozorgi, and R. G. Raj, “A Novel Two-Stage Spectrum-Based Approach for Dimensionality Reduction: A Case Study on the Recognition of Handwritten Numerals,” Journal of Applied Mathematics, vol. 2014, Article ID 654787, 14 pages, 2014. doi:10.1155/2014/654787.

[20] Q. Ma, and M. D. Wong, “NP-Completeness and an approximation algorithm for rectangle escape problem with application to pcb routing,” *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 9, pp. 1356-1365, 2012.

[21] Y. K. Ho, H. C. Lee, and Y. W. Chang, “Escape routing for staggered-pinarray PCBs,” *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 9, pp. 1347-1356, 2013.

[22] K. Wang, S. Dong, H. Wang, Q. Chen, and T. Lin, “Mixed-crossing-avoided escape routing of mixed-pattern signals on staggered-pin-array PCBs,” *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 4, pp. 571-584, 2014.

[23] J. T. Yan, "Length-constrained escape routing of differential pairs." *Integration, the VLSI Journal* 48, pp. 158-169, 2015.

[24] H. Kong, Q. Ma, T. Yan, and M. D. Wong, "An optimal algorithm for finding disjoint rectangles and its application to PCB routing, " *in Proc. ACM/IEEE Design Automation Conference (DAC)*, pp. 212-217. 2010.

[25] NB. Anuar, H. Sallehudin, A. Gani, and O. Zakari. "Identifying false alarm for network intrusion detection system using hybrid data mining and decision tree." *Malaysian journal of computer science* 21, no. 2, pp. 101-115, 2008.

[26] J. T. Yan, T. Y. Sung, and Z. W. Chen, “Simultaneous escape routing based on routability driven net ordering,” *in Proc. IEEE International System-On-Chip (SOC) Conference (SOCC)*, pp. 81-86, 2011.

[27] YM. Alginahi, MN. Kabir, and AI. Mohamed. "Optimization of High-Crowd-Density Facilities Based on Discrete Event Simulation." *Malaysian Journal of Computer Science,* 26, no. 4, 2013.

[28] E. Ahmed, A. Gani, M. Sookhak, S. H. Hamid, and F. Xia. "Application optimization in mobile cloud computing: Motivation, taxonomies, and open challenges." *Journal of Network and Computer Applications,* 52, pp. 52-68, 2015.

[29] J. Czyzyk, M. Mesnier, and J. More, "The NEOS Server," *IEEE Journal on Computational Science and Engineering*, vol. 5, issue 3, 1998.

[30] R. Fourer, D. M. Gay, B. W. Kernighan, *AMPL: a modeling language for mathematical programming,* Duxbury Press/Brooks/Cole, 2002.