Mobility based Net Ordering for Simultaneous Escape Routing

Kashif Sattar
University Institute of Information Technology
Arid Agriculture University
Rawalpindi, Pakistan

Anjum Naveed
School of Electrical Engineering and Computer Science (SEECS)
National University of Sciences and Technology (NUST)
Islamabad, Pakistan

Aleksandar Ignjatovic
School of Computer Science and Engineering (CSE)
University of New South Wales (UNSW)
Sydney, Australia

Muhammad Zeeshan
School of Electrical Engineering and Computer Science (SEECS)
National University of Sciences and Technology (NUST)
Islamabad, Pakistan

***Abstract*—With the advancement in electronics technology, number of pins under the ball grid array (BGA) are increasing on reduced size components. In small size components, a challenging task is to solve the escape routing problem where BGA pins escape towards the component boundary. It is often desirable to perform ordered simultaneous escape routing (SER) to facilitate area routing and produce elegant Printed Circuit Board (PCB) design. Some heuristic techniques help in finding the PCB routing solution for SER but for larger problems these are time consuming and produce sub-optimal results. This work propose solution which divides the problem into two parts. First, a novel net ordering algorithm for SER using network theoretic approach and then linear optimization model for single component ordered escape routing has been proposed. The model routes maximum possible nets between two components of the PCB by considering the design rules based on the given net ordering. Comparative analysis shows that the proposed net ordering algorithm and optimization model performs better than the existing routing algorithms for SER in terms of number of nets routed. Also the running time using proposed algorithm reduces to $O(2^{NE/2}) + O(2^{N\bar{E}/2})$ for ordered escape routing of both components. This time is much lesser than $O(2^{NE})$ due to exponential reduction.**

***Keywords*—Net ordering; optimization model; ordered escape routing; PCB routing; simultaneous escape routing**

I. Introduction

With the advancement in technology, the demand for small size electronic components with larger number of pins increases [1]. These components can be Integrated Circuits (ICs) or BGA type with hundreds of connectivity pins [2]–[4] in the form of grid. Usage of such components necessitates sophisticated design and advanced technologies for producing small sized electronic circuit boards. Multiple BGAs need to be connected on the PCB, which require connectivity of pins by routing of nets. This kind of routing has two parts, one is escape routing which is from pin to the boundary of the component and the other is area routing which is between the components as shown in Fig. 1. Presence of BGA components on smaller sized PCBs significantly increases the routing complexity.

Routing in PCBs is the problem of connecting the pins of different components on the PCB to make it operational. The circuit wire lying on the PCB, between the pins pair (that needs to be connected) is called a net. Routing of nets becomes more complex, if there are more number of nets and needs ordered escape routing. Ideally, the routing algorithm shall connect all nets as a planar graph (nets don't cross each other) and the nets shall be entirely contained within the area of the PCB. Given the high density of components on smaller sizes PCBs, and hundreds of nets to route. A single layer is not enough to route all nets in a planar fashion particularly if ordered escaping is required. Therefore, multiple routing layers are used to ensure planarity as shown in Fig. 1.

PCB routing for given set of connectivity pins is similar problem as constructing of planar graph for a given set of nodes, which is known to be NP-Hard [5]. BGA components have small size balls/pins under the surface, in the form of a grid. This high density grid reduces the net escaping space and make the routing more complex. Therefore, PCB routing is divided into two parts: 1) escape routing where routing is to be done from pin to the component boundary; and 2) area routing where the net between one component to other component is established as shown in Fig. 1. To facilitate the area routing, escape routing is required in some particular order. Therefore ordered escape routing is a major challenge in PCB routing, which becomes even more challenging for simultaneous escape routing (SER) [6]–[12]. In SER pin escaping for multiple components is done simultaneously in a specific order to reduce the complexity of area routing as shown in Fig. 2. To reduce the complexity of SER, the problem has been divided into two parts i.e. ordered escape routing of two components separately. In this case, net order has necessarily required.

Focus of this research is to find the net order for simultaneous escape routing. In this paper, an algorithm to find the net order has been proposed. This reduces the problem into half and reduces the complexity of routing by many times. In this way the area routing becomes very simple and it is just a

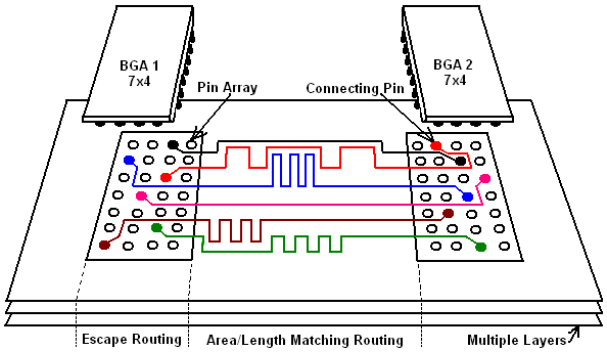

Fig. 1. PCB routing.

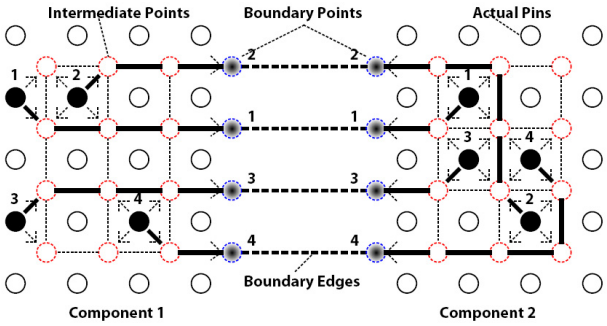

Fig. 2. Simultaneous escape routing.

matter of connectivity of straight wires without any crossing to maintain planarity. The proposed algorithm finds the net order and do ordered escape routing for each component according to that net order. For ordered escape routing optimization model has been used, which helps in detailed routing from a connectivity pin to an escape boundary point. 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. Following are the contributions of this work:

- In the first step an algorithm to find the optimal net order has been proposed. This algorithm helps by converting the complex problem of simultaneous escape routing into simple and single component ordered escape problem. Ultimately this exponentially reduces the complexity of the problem.
- In the second step ILP optimization model for ordered escape routing has been modified, which helps in finding the ordered net escape from a single component.

Collectively these two contributions help in solving SER problem in a much better way.

The rest of the paper is organized as follows: In Section 2, some basic techniques used for planar routing and advancements in this area with a review of the latest literature has been discussed. Section 3 presents proposed algorithm to find a net order. In Section 4, optimization model for ordered escape routing under the constraints of planar routing and escape capacity has been proposed. In Section 5, algorithm/model validation and comparative evaluation of the proposed algorithm has been performed. Section VI concludes this work.

II. Related Work

Routing of nets on PCB, is divided into two main categories, one is area and the other is escape routing [7]. The objective of escape routing is to establish the planar routes by escaping the nets from pins to the component boundary 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 [13], [14].

Escape routing is of three types [15]: 1) Unordered escape routing where there is no particular sequence of pin escaping towards component boundary from a single component [15]–[21]; 2) ordered escape routing where there is a particular sequence (order) of pin escaping towards component boundary from a single component [22] (one of the example of this type of routing is Bus escape which is also called as maximum disjoint subset problem [23]); and 3) simultaneous escape routing where pins from multiple components escaped in the same order. This problem relatively more complex as compare to other escaping problems. To find the solution of SER is time consuming process, particularly for larger problems. This time can be reduced if the escape net order is known in advance. Escape net order is a sequence of nets to escape from the boundary of a component. This prior net ordering information converts the problem of SER into ordered escape routing of a single component. Focus of this work is to find the optimal net order. The following subsections discuss the related literature in detail.

A. Simultaneous Escape Routing

The initial research work proposed for SER consists of two parts. Unordered escaping of pins proposed in the first component whereas for the second component, ordered escape routing proposed to match the pins of first component to get SER solution. However, this approach might results in an incomplete routing. Ozdal et al. [7] carried out the first well known SER research. Graph based approach was used to find maximal escape nets. This approach gives better results only if the pins of both components are closely aligned to each other. However, complex problems of SER cannot be solved using fixed escape patterns used in this approach. Ozdal et al. [8] also proposed randomized algorithm for large scale problems in their extended work. However, proposed monotonic escape of nets in one direction cannot produce optimal results and also it limits its applications. For routing pattern generation, Ozdal et al. [9] also used congestion driven router, however for large scale SER problems the graph of escape patterns becomes complicated.

Simultaneous Pin Escape [12] is a flow model based approach. Authors show that for a particular probelm if construction of planar graph is possible then this approach guarantees planar routing. The existence of a solution can be determined by the relation between maximum flow and number of escape pins of the BGA component [12], [16]. This approach must provide a planar solution if maximum flow is greater or equal to the escape pins; however, for higher inter

pin capacities 'no net crossing' cannot be ensured and exact value of maximal flow is not known. A bus oriented escape routing with consecutive constraints has been proposed for SER on multiple layers [24]. The focus of this research is to reduce the CPU time, however if the pins are randomly placed on each component then its very difficult to form buses and also unable to get optimal routing solution due to escaping of nets in the form of buses.

B. Net Ordering Algorithms

B-Escape is a routing algorithm proposed by Luo et al. [10]. This algorithm uses dynamic net ordering approach and performs well when compared with Allegro PCB router [10]; however it consumes significant time to find a suitable net order and in reordering of nets again and again. Rout-ability Driven Net Order proposed by Yan et al. [25] is a multi-step approach. An important step is identification of a net ordering, which is derived on the basis of 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. [12]. The drawback is that the second step is totally dependent on the output of the first step i.e. net ordering. It does not explore all possible paths and may lead to sub-optimal results in complex problem instances.

Escaped Boundary Pins Routing for High-Speed Boards proposed by Chin et al. [26] is an algorithm for finding static net ordering by dynamic pin sequence (DPS). Another work proposed by Kumtong et al. [27] is based on set sequence (SS) technique. While these algorithms can better utilize the routing space and adapt to wire length and shape requirements, their focus is on boundary pin routing and are not suitable for BGA type of PCB components which need escaping from inside the components; their use can also be more challenging, particularly in case of SER.

Most of the research on SER is based on basic constraints of capacity and construction of planar graph. Also most of them are using heuristic approaches to get the routing solution. This research considers 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, clustering of computational resources to perform complex computational tasks and using net ordering techniques, the drawback can easily be overcome. In this research an algorithm for optimal net ordering has been proposed which reduces the computational time and outperforms the existing algorithms.

III. Algorithm for Net Ordering

This section, first define the terms used in the algorithm and then discuss the tile model and net ordering algorithm in detail. It is also assumed that there are only two components and only one side escape is possible from the side of component, which is facing towards the other component.

$Rows_{Ci}$: Number of rows of pins for component Ci.

$Orthogonal_{Cap}$: This capacity refers to the number of nets that can pass through between two neighboring pins vertically or horizontally.

$Diagonal_{Cap}$: It refers to the number of nets that can pass through between two neighboring diagonal pins.

Cap_{Ci} : Escape routing capacity for component Ci.

SER_{Cap}: Simultaneous Escape Routing capacity.

$BoundaryPins_{Ci}$: Number of pins on Escape boundary line which needs connectivity for the component Ci.

$URDF\_Mobility_{Pi}$: Number of nets that can pass from Up, Rear, Down and Front side of pin Pi.

A. Net Ordering Algorithm

This proposed algorithm performs four basic functions in sequence as discussed in Algorithm 1, which are $SER\_Capacity()$, $Initial\_NetOrder()$, $URDF\_Mobility\_Values()$ and $Final\_NetOrder()$ respectively. This algorithm produces the optimal net order for SER, if it exists. Then escape routing for each component performed separately according to that net order with the help of an optimization model.

Algorithm 1 Net Ordering Algorithm

```
1:  procedure NET_ORDERING()
2:      Routable ⇐ SER_Capacity()
3:      if !Routable then
4:          print 'Complete Routing not possible' and exit
5:      Pins_C1[n] ⇐ 'n' Connectivity pins of component 1
6:      Pins_C2[n] ⇐ 'n' Connectivity pins of component 2
7:      I[n] ⇐ Initial_NetOrder(Pins_C1[n], Pins_C2[n])
8:      M1[n][4] ⇐ URDF_Mobility_Values(Pins_C1[n])
9:      M2[n][4] ⇐ URDF_Mobility_Values(Pins_C2[n])
10:     F[n] ⇐ Final_NetOrder(I[n], M1[n][4], M2[n][4])
11:     if F[n] = Null then
12:         print 'Complete Routing not possible' and exit
13:     else
14:         return F[n]
```

1) SER Capacity Function: $SER\_Capacity()$ function has been discussed in Algorithm 2. First calculate the escape capacity of each component. Here $BoundaryPins_{Ci}$ has been added in Cap_{Ci}, because they can escape directly from the component boundary without effecting the other nets or orthogonal capacity. Obviously the SER capacity must be the minimum escape capacity of both components. Then it is also verified whether the routing with the required number of nets $'n'$ is possible or not.

Consider an example instance of two BGA components of size 6x4, having 5 connectivity pins as shown in Fig. 3. Also consider left component as $C1$ and right component as $C2$. $Rows_{C1}$ and $Rows_{C2}$ are 6, whereas $Orthogonal_{Cap}$ is 1. Since there is no boundary connectivity pin, therefore Cap_{C1} is 5 and Cap_{C2} is also 5. According to Algorithm 2, SER_{Cap} is also 5. There are 5 pins which need connectivity (escape routing) and these are less than or equal to the SER_{Cap}, therefore the Algorithm 2 returns true and move on the next step of the main Algorithm 1.

Algorithm 2 SER Capacity Algorithm

procedure SER_CAPACITY()
  $Cap_{C1} = (Rows_{C1} - 1)xOrthogonal_{Cap} + BoundaryPins_{C1}$
  $Cap_{C2} = (Rows_{C2} - 1)xOrthogonal_{Cap} + BoundaryPins_{C2}$
  $SER_{Cap} = min(Cap_{C1}, Cap_{C2})$
  if $SER_{Cap} >= n$ **then**
    return $true$
  else
    return $false$

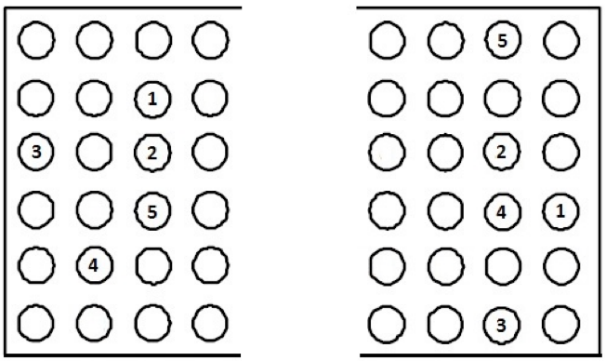

Fig. 3. BGA component of size 6x4.

2) Initial Net Order: To find the initial net order according to the position of pins in the grid of the component, $Initial\_NetOrder()$ function in Algorithm 3 has been discussed. Also some rules for rearrangement of the initial net order has been defined.

In the first step, convert each row in a vertex for both components as shown in Fig. 4. Then for the rows having multiple connectivity pins apply the rules defined in Step 2 to Step 13 of Algorithm 3. In this example there is Pin-3 on non-escape boundary of left component and also there is one pin above that row whereas there are two escape boundary points above that row. Therefore select Pin-3, first in order. Similarly select Pin-1, first in order in the right component. In the third step connect all pins to their corresponding pins with edges, as shown in Fig. 4.

3) URDF Mobility Values: In the third basic function of Algorithm 1, find the $URDF$ mobility values of a pin according to the position of pin in the component as shown in Fig. 5.

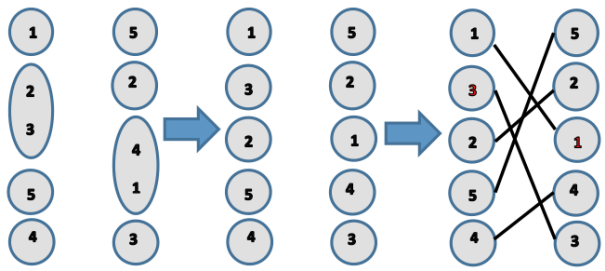

Fig. 4. Three step process to find initial net order.

Algorithm 3 Initial NetOrder Algorithm

procedure INITIAL_NETORDER($Pins_{C_1}[n], Pins_{C_2}[n]$)
  Read row wise all pins from top to bottom and put them in column form, separately for each component.
  if there are multiple pins in a single row **then**
    if there are multiple pins in the first row **then**
      order must be from the escape side to non-escape side.
    else if there are multiple pins in a last row **then**
      order must be from the non-escape side to the escape side.
    else if there is a pin on the non-escape boundary column and the number of escape boundary points are greater than the number of pins above that row **then**
      select this pin first and name it as prioritized rear boundary pin.
    else if In cases, where the same multiple pins are in a single row in both components **then**
      select from the escape side to the non-escape side.
    else
      For all other rows order should be according to the row number of the corresponding pin in the other component.
  Now there are two separate columns having number of rows equals to the number of pins. This is called initial net ordering that facilitates in finding the final net ordering. Connect pin in the left column with its corresponding pin in the right column with edges as shown in the last step of Fig. 4.

Algorithm 4 URDF Mobility Values Algorithm

procedure URDF_MOBILITY_VALUES($Pins_{C_i}[n]$)
  Find the Up, Rear, Down and Front ($URDF$) mobility values of a pin by identifying its position in the grid.
  $U = i, R = j, D = k$ and $F = l$ implies that there are $(i/Orthogonal_{Cap})$ number of rows on the up side of the connectivity pin, $(i/Orthogonal_{Cap})$ number of columns on the rear side, $(k/Orthogonal_{Cap})$ number of rows on the down side and $(l/Orthogonal_{Cap})$ number of columns on the front side (escape side) of the pin respectively, in a component. These values help in finding the number of nets that can possibly be passed through, from that side of the pin.

According to the example of Fig. 3, Pin-1 is in second row and third column of the left component, which implies that there are one row on the upside of the pin, four rows on the down side, two columns on the rear side and one column on the front side. Since $Orthogonal_{Cap} = 1$ is being considered, therefore the $URDF$ values are $U =$ (No. of rows on Up side $*Orthogonal_{Cap}) = 1$, $R = 2$, $D = 4$ and $F = 1$ respectively as shown in Fig. 5. Furthermore, $U = 1$ implies that only one net can pass through the upside of Pin-1 and similarly for the other values.

4) Final Net Order: The last function of Algorithm 1finds the final net order as shown in Fig. 6, by eliminating a crossing of edges among two columns. To eliminate the cross, there are four possibilities. For example, consider the first

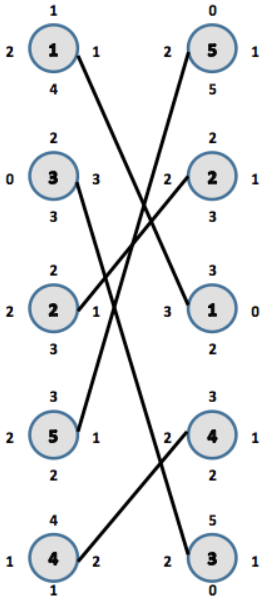

Fig. 5. Finding the URDF mobility values.

crossing between Pin-1 and Pin-5, as shown in Fig. 5. The first possibility is to move the Pin-5 in left column to the top of the Pin-1 in the left column. Second possibility is to move Pin-1 in the left column to the bottom of the Pin-5 in the same column. Similarly, the third possibility is to move the Pin-1 in the right column to the top of Pin-5 of the right column and finally the fourth possibility is to move Pin-5 of the right column to the bottom of Pin-1 in the same column. Try all the four possibilities in some order, till selection of one, which is allowed by $URDF$ values as discussed in Algorithm 5. Failure of all the possibilities shows that routing of both nets is not possible in this scenario.

In final net ordering process, initially considers first pin of each column, which is Pin-1 and Pin-5 as shown in Fig. 6. Since the row of Pin-5 in its component is higher (smaller row number), therefore select Pin-5 and try to move its corresponding pin from left component towards upside. But on the way there is Pin-3, which has rear value, $R = 0$. This implies that Pin-5 cannot route from rear side of Pin-3. Therefore, first move Pin-3 towards downside of Pin-5. Since the rear and down values of Pin-5 are greater than zero, hence route Pin-3 from the rear side of Pin-5. After the routing of Pin-3, route Pin-5 towards upside at the level of selected Pin-5 of right component as shown in Fig. 6. In this way eliminate all the crossings and get the final nets order if exists. The proposed algorithm not only converts the SER problem into much simplified ordered escape routing problem by providing the net order, but also provides the global routing of each net. In the next section optimization model for detailed routing is being used.

IV. Optimization Model for Ordered Escape

Optimization models are very helpful to solve real life problems [28], [29]. These models provide optimal solution by considering all constraints. This section presents the modified Integer linear program based optimization model proposed in

Algorithm 5 Final NetOrder Algorithm

procedure FINAL_NETORDER($I[n], M_1[n][4], M_2[n][4]$)
Consider a current variable, which initially points to the first pin of each column. In each iteration, the current variable moves to the next row. Let Pin-A be from left column and Pin-B be from the right column.
Select the pin according to the following rules:
if Escape boundary connectivity pins are already fixed with their escape boundary points **then**
select it first and arrange its counter pin.
else if A and B both have the same position in the corresponding ordering of pins of the two components and no other pin is being blocked by routing this net **then**
these pins are already in order. Only need to update the mobility values of the left over unordered pins. If, by this choice, any pin is being blocked then reselect that pin first.
else if A or B is a prioritized rear boundary pin **then**
select that pin first. If, by this choice, any pin is being blocked then reselect that pin first.
else if If A or B is not in the component's first row and the next row pin (w.r.t current row) of both columns are the same i.e C **then**
select C
else
select the pin whose row number is lesser.
Route the selected pin (e.g A) by moving its counter Pin-A' (or both in case C is selected) towards upside in a column to reach at current row level by these rules:
if First pin to move upward **then**
move the counter Pin-A' (both in case of C) upward from the rear side of all the pins, till reaching at current row level to eliminate the edge crossing.
else if on the way any pin have R=0 or U=0 **then**
first move that pin below the counter pin A' (counter pin must have R>0 and D>0). If there is more than 1 pin which faces this situation then order of moving down pins is from escape side to non-escape side.
if routing is not possible by selecting Pin-A and still Pin-B remians **then**
select Pin-B and goto Step 15
if routing is not possible by selecting any pin A or B **then**
return false
else
update the mobility values. The value in each direction will be either the distance between the pin and the net passing through in that direction or connected pin whichever is lesser. Move the current variable to the next row and goto Step 3.
if eliminate all crossings by reordering pins **then**
this is our final net order and the function returns true.
else
return false

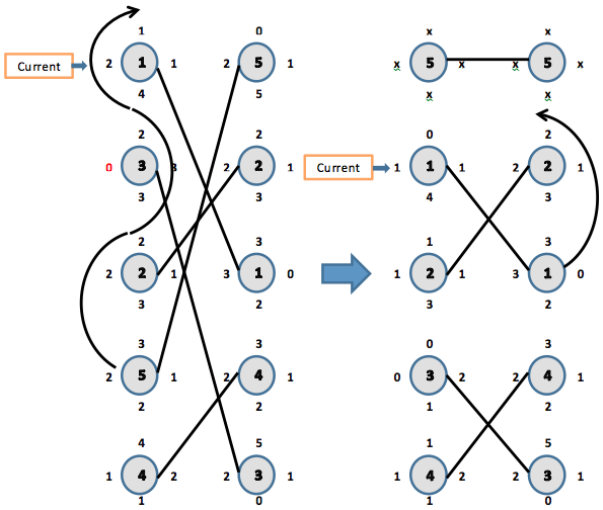

Fig. 6. Finding the final net order.

our earlier work [30]. System constraints from the model has been removed because these are useless once net order has been finalized. The graph $G(V, E)$ serves as input to this model, where V is a set of vertices. This set consists of boundary points, intermediate points and connecting pins. All the possible edges are in set E and the nets use these edges for escape towards the boundary of the component.

Based on the given constraints, the proposed model returns another graph $G'(V', E')$, which shows the maximum number of possible escaped nets, with $G'(V', E') \leq G(V, E)$, which also implies that $V' \leq V$ and $E' \leq E$. Linear programming model has been formulated that can perform escape routing of nets from a component according to the given net order identified in the previous section. These are the sets used as input in the proposed model:
N= Set of required nets.
I = All intermediate and boundary points.
V = All intermediate points, boundary points and connecting pins.
BE= All edges at the component's boundary.
E = All possible edges of a BGA. The proposed model contains decision variable $X_{(n_a,e_{ij})}$, which is a Boolean variable for all nets $n_a \in N$ and for all edges $e_{ij} \in E$. If the solver assign true or 1 to the decision variable then that edge e_{ij} for a particular net n_a for escape routing is being used, otherwise the solver assigns false or zero value to the decision variable.

$$X_{(n_a,e_{ij})} = \left\{ \begin{array}{l} 1 \text{ if } e_{ij} \text{ edge is being used for net } n_a \\ 0 \text{ if } e_{ij} \text{ is not being used for } n_a \end{array} \right\}$$

By using this optimization model, routing is constrained by the constraints of connectivity, planar graph, net order and system constraint. Each constraint is important in terms of achieving error free routing designs [30]. The following set of equations (1)-(6), completely represents the optimization model.

$$\max . \sum_{n_a \in N} \sum_{e_{aj} \in E} X_{(n_a,e_{aj})} \quad (1)$$

subject to:

$$\sum_{e_{ij} \in E \,\&\, i=a} X_{(n_a,e_{ij})} \leq 1 \qquad \forall n_a \in N \quad (2)$$

$$\sum_{e_{ij} \in E} X_{(n_a,e_{ij})} = \sum_{e_{jk} \in E} X_{(n_a,e_{jk})} \qquad \forall n_a \in N, \forall j \in I \quad (3)$$

$$X_{(n_a,e_{ij})} + X_{(n_a,e_{ji})} \leq 1 \qquad \forall n_a \in N, \forall e_{ij}, e_{ji} \in E \quad (4)$$

$$\sum_{n_a \in N} \sum_{e_{ij} \in E} X_{(n_a,e_{ij})} \leq 1 \qquad \forall j \in I \quad (5)$$

$$\sum_{e_{ij} \in BE} X_{(n_a,e_{ij})}.j \leq \sum_{e_{kl} \in BE} X_{(n_{a+1},e_{kl})}.l \qquad \forall n_a \in N \quad (6)$$

V. Evaluation

Proposed net ordering algorithm has been implemented in C++ and for detailed routing, optimization model has been implemented in AMPL language [31]. Initially, the proposed algorithm and optimization model has been validated in Sub-section V-A and V-B, respectively. Subsequently the performance is being evaluated in Sub-section V-C.

A. Algorithm Validation

This section verify that the algorithm proposed in Section III, gives the optimal net order if SER has a valid routing solution. Furthermore if a particular scenario has no complete routable solution then it return false. Algorithm validated for this purpose on BGA component as shown in Fig. 3. Both components $C1$ and $C2$ have 6 rows from A to F and 4 columns from 1 to 4. There are 5 connectivity pins in each component that needs escape routing to connect with each other. To simplify the explanation, in this example assume $Orthogonal_{Cap} = 1$ and $Diagonal_{Cap} = 2$. Twelve (12) scenarios has been considered with randomly selected 5 escape pins in each component. In the first scenario, first pin is at location B4 in $C1$ and location E2 in $C2$. Similarly see the location of other pins in Fig. 7. The net order obtained by our proposed algorithm is 1,3,2,4 and 5. Then run the same scenario in commercially available Proteus auto router by Labcenter Electronics Ltd. The result obtained by Proteus is exactly in the same order as obtained by our algorithm as shown in Fig. 7.

Complete validation results can be seen in Table I. First column is the scenario number, second and third columns are the pin positions in each component respectively from Pin-1 to 5. Fourth column is the net order obtained by the C++ program for the proposed algorithm and the fifth column is the net order obtained by the Proteus router. Results validate the algorithm for all scenarios. In some cases the algorithm returns false, which implies that complete routing for all the nets is not possible as shown in Scenario 7 of Table I. Running the same scenario in Proteus, resulted in the un-routable solution as shown in Fig. 8. Routing of Pin-1 from B3 of component 1 to F2 of component 2 is not possible, which validates the results.

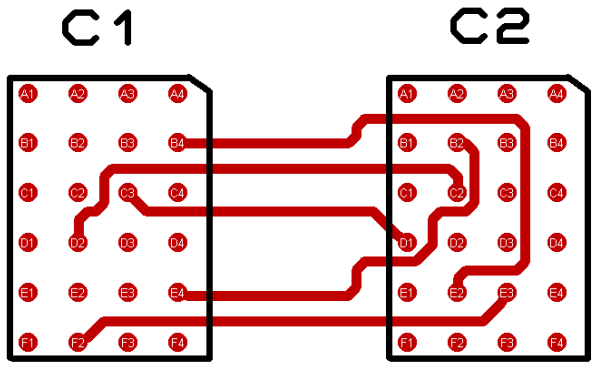

Fig. 7. SER in BGA 24_1.5.

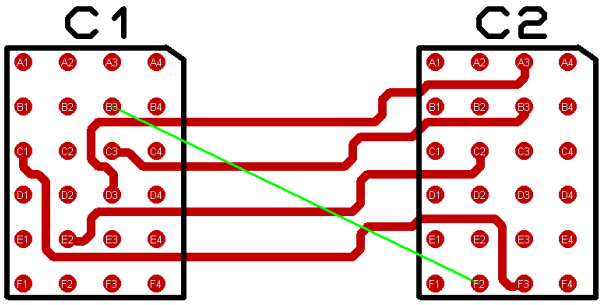

Fig. 8. Proteus incomplete routing.

B. Model Validation

Consider Scenario 1 of Table I for the validation of model. The net order obtained by the proposed algorithm is 1, 3, 2, 4, and 5. This net order converts the SER problem into two single components ordered escape routing problems. This example clearly shows that the optimization model efficiently performs ordered escape routing for PCB components based on BGA, as shown in Fig. 9. There are maximum 5 boundary points in the BGA component and model chooses to route all five nets by escaping in a given order. It is evident that this solution ensures planarity by not selecting any vertex or edge twice. The routing of nets is exactly the same as obtained by Proteus auto router as shown in Fig. 7. This example performs comprehensive validation of proposed optimization model and ensures all constraints of connectivity, net order and planar graph.

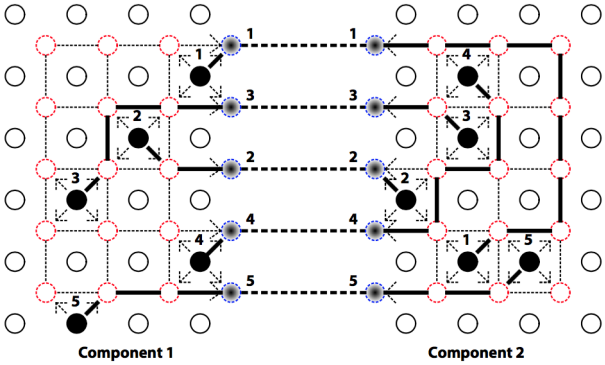

Fig. 9. Ordered escape routing with proposed model.

TABLE I. VALIDATION RESULTS AND COMPARISON WITH PROTEUS

S.No.	C1 Pins	C2 Pins	NetOrder		Accuracy
			Algorithm	Proteus	
1	B4, C3, D2, E4, F2	E2, D1, C2, B2, E3	1, 3, 2, 4, 5	1, 3, 2, 4, 5	100%
2	B3, B4, C1, E2, D3	D4, C3, F3, D2, A3	5, 2, 1, 4, 3	5, 2, 1, 4, 3	100%
3	B3, C3, C1, E2, D3	D4, C3, F3, D3, A3	5, 1, 2, 3, 4	5, 1, 2, 3, 4	100%
4	B3, C3, C1, E2, D3	D2, C3, F3, D3, A3	5, 1, 2, 4, 3	5, 1, 2, 4, 3	100%
5	C2, C1, D1, E2, B1	B3, A2, F2, E4, D3	1, 5, 2, 4, 3	1, 5, 2, 4, 3	100%
6	D3, E2, C3, C1, B3	A3, F2, C3, F3, D4	1, 5, 3, 4, 2	1, 5, 3, 4, 2	100%
7	B3, C3, C1, E2, D3	D2, C3, F3, D3, A3	Routing of all nets not possible	Routing of all nets not possible	100%
8	B3, C3, C1, E2, D2	E2, B3, F3, C2, A3	5, 1, 4, 2, 3	5, 1, 4, 2, 3	100%
9	D3, D4, C1, D1, D2	A3, D3, C3, E3, B3	5, 3, 1, 2, 4	5, 3, 1, 2, 4	100%
10	A3, C3, D2, E3, F2	E2, D2, C2, B2, E3	1, 3, 4, 2, 5	1, 3, 4, 2, 5	100%
11	D3, C3, F2, A3, C2	F2, C2, E2, B2, D3	4, 2, 5, 1, 3	4, 2, 5, 1, 3	100%
12	C2, F3, B3, C4, E1	D4, A1, E1, C2, A4	Routing of all nets not possible	Routing of all nets not possible	100%

C. Performance Analysis

The performance comparison in this section is to find the optimal net order by comparing the results of our proposed mobility based net ordering (MBNO) algorithm with the routability driven net ordering (RDNO) algorithm [25]. Consider three different scenarios of the same examples discussed in [25] for the purpose of evaluation. In this example there are 9 rows and 10 columns for both components, having 90 pins in total. There are 14 nets, which need connectivity and escaping of pins for SER. Also single side escape is permitted with $Orthogonal_{Cap}$ equal to 2 and the $Diagonal_{Cap}$ equal to 3. Each scenario highlights different aspect of the proposed

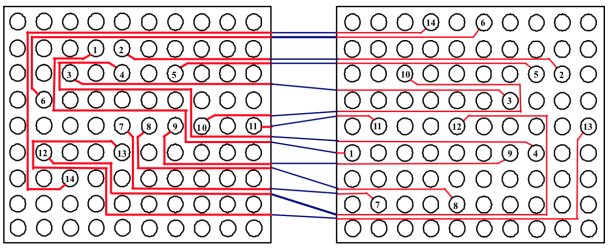

Fig. 10. Routability driven Net ordering [25].

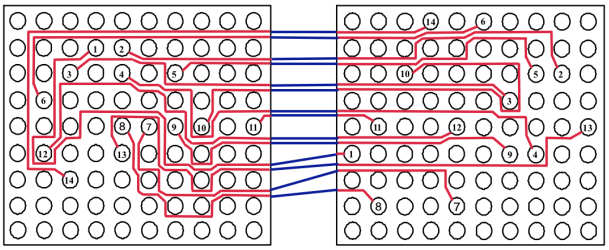

Fig. 11. Proposed Mobility based Net ordering.

algorithm as explained in respective sub-section.

1) BGA_9x10 Default case Scenario-1: To validate, results obtained by our proposed MBNO algorithm has been compared with the results obtained by RDNO algorithm. For the default example discussed in [25] both provided the optimal net order. Fig. 10 shows the output of RDNO algorithm and the net order is: 14, 6, 2, 5, 3, 10, 11, 4, 1, 9, 7, 8, 12, 13. Whereas Fig. 11 shows the output of the proposed algorithm and the net order is: 14, 6, 2, 5, 3, 10, 4, 11, 12, 9, 1, 13, 7, 8. Results show that proposed algorithm not only provides optimal net order but also provides more compaction to the escaping nets by utilizing all escape boundary points in a sequence, which can accommodate more nets within the limits of SER_{Cap}.

2) BGA_9x10 Swap pin Scenario-II: In this scenario swap the pin number 6 and 14 with each other in right component of Fig. 10, whereas rest of the things remain exactly the same. The net order remains the same using RDNO algorithm but after placing net number 14 as a first net, it blocks pin number 6 of the right component as shown in Fig. 12 and ends with incomplete routing. However our proposed MBNO algorithm provides a complete routing as shown in Fig. 13, with the following net order: 6, 14, 2, 5, 3, 10, 4, 11, 12, 9, 1, 13, 7, 8.

3) BGA_9x10 Back boundary pin Scenario-III: This scenario place the pin number 3 at the back boundary of left component of Fig. 10, whereas rest of the settings remain exactly the same in both components. The net order remains the same using RDNO algorithm but after placing net number 14 and 6 as a first and second net respectively, it blocks pin number 3 of the left component as shown in Fig. 14 and ends with an incomplete routing. However our proposed MBNO algorithm first provides the partial routing to the blocked pin i.e. pin number 3, as shown in Fig. 15. Then the algorithm does routing for the pin number 14 and ultimately provides optimal net order for complete routing: 14, 6, 2, 5, 10, 4, 11, 1, 9, 12, 3, 13, 8, 7.

Detailed routing for the given net order is shown in Fig. 16. Results show that how efficiently proposed algorithm solves the SER problem based on the mobility values for each pin.

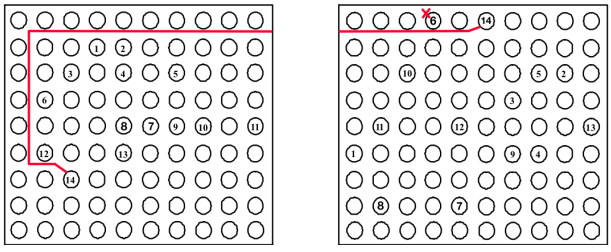

Fig. 12. RDNO routing for swap pin scenario.

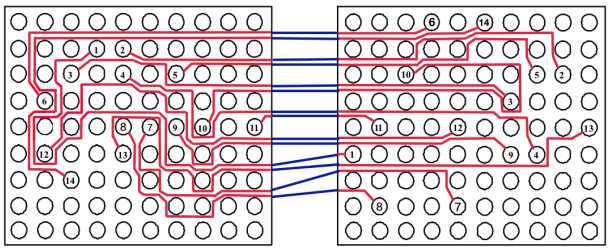

Fig. 13. MBNO routing for swap pin scenario.

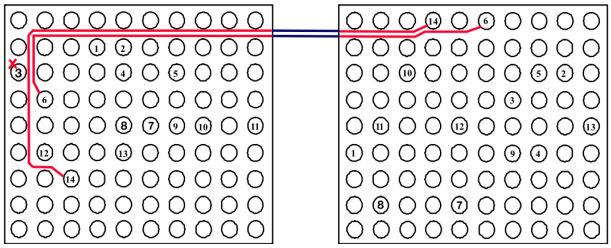

Fig. 14. RDNO routing for back boundary pin scenario.

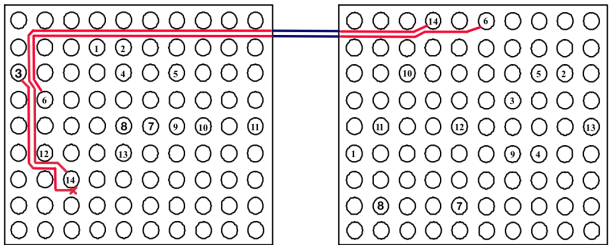

Fig. 15. MBNO partial routing for back boundary pin scenario.

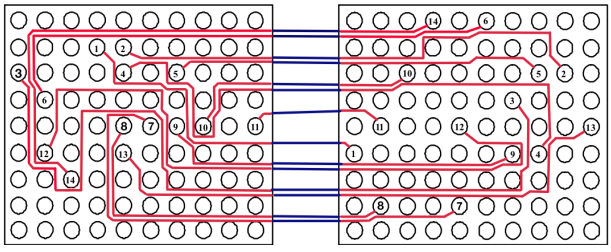

Fig. 16. MBNO complete routing for back boundary pin scenario.

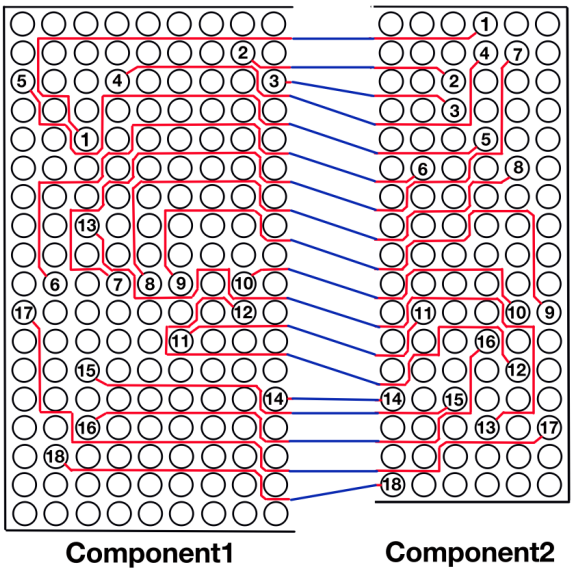

Fig. 17. MBNO results for varying size BGA′s scenario.

TABLE II. EXPERIMENT RESULTS AND COMPARISON WITH RDNO ALGORITHM

Scenario No.	C1 Pins	C2 Pins	No. of Nets	Net Routing	
				RDNO	MBNO
I	9x10	9x10	14	14/14	14/14
II	9x10	9x10	14	Incomplete	14/14
III	9x10	9x10	14	Incomplete	14/14
IV	18x9	17x6	18	Incomplete	18/18

4) Varying Size BGA's Scenario-IV: The scenarios discuss so far are variations of the same example having two BGA components of the same size. Now consider another example having different size BGA′s. Component 1 is of size 18x9 and component 2 is of size 17x6. There are 18 nets that need SER connectivity. Again assume single side escape with $Orthogonal_{Cap}$ equals to 1 and the $Diagonal_{Cap}$ is 2. Fig. 17 shows, that the proposed algorithm generates the optimal net order for successful completion of SER for all the nets.

Here, results have been compared by different methods discussed so far. Proposed MBNO algorithm performed 100% routing in all of the four scenarios by generating optimal net order, however RDNO [25] could not route all nets in Scenario-II, Scenario-III and Scenario-IV as shown in Table II. It is important to note that manual routing of even single net left out by an automated method is very difficult to route and is a very time consuming process requiring rearranging of all the routed nets. It is also clear from the results that based on the mobility values of each pin there are better routing results because it avoids any connectivity pin blockage while finding out the next net in the optimal net order.

VI. CONCLUSION

This research work propose the use of net order, in order to solve the SER problem in a way superior to the previous state of art methods. A mobility based net ordering algorithm has been proposed that finds the optimal net order and converts the problem of SER into two simpler problems of finding an ordered escape routing for each component separately. Once "connectivity pins" escape from each component, according to the net order obtained by the proposed algorithm, then area routing becomes quite simple by connecting nets of both components in an order preserving way. Net order obtained by proposed algorithm for 12 representative and randomly generated scenarios has been validated and compared it with the net order obtained by Proteus auto router. Proposed ILP optimization model has been updated to get the detail routing results. The detailed routing results are similar with the results obtained by Proteus auto router. The time for SER optimization model is exponential, i.e., $O(2^{NE})$. Where N is the number of nets that needs to be routed and E is the set of all type of edges. This means addition of only one more variable (net or edge), doubles the computation time due to exponential behaviour, i.e., $O(2^{NE+1})$. Similarly by decreasing the edges results in exponential reduction in time. In this case, edges are being reduced to almost half (assuming both components have equal edges). Therefore the new time is $O(2^{NE/2})+O(2^{NE/2})$ for ordered escape routing of both components. This time is much lesser than $O(2^{NE})$ due to exponential reduction. Results shows that proposed algorithm finds routable net order for all scenarios whereas previous algorithm fails to route all nets according to their derived net order.

Now a days, there are different types of BGA components based on their pin arrays. These types include, square pin array, triangle pin array, diamond pin array and hexagonal pin array. The proposed models in this research focus only on BGA pins forming square shape grids, in future we can apply this research on other available BGA grid shapes like triangle, diamond and hexagonal to see the increase in capacity for the same size of components. Also in future we are extending the algorithm for higher number of components.

REFERENCES

[1] M. M. Ozdal, "Routing algorithms for high-performance vlsi packaging," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2005.

[2] G. Marcantonio, "Ball grid array (bga) integrated circuit packages," Aug. 18 1998, uS Patent 5,796,170.

[3] D. Y. Chong, B. Lim, K. J. Rebibis, S. Pan, S. Krishnamoorthi, R. Kapoor, A. Sun, and H. Tan, "Development of a new improved high performance flip chip bga package," in *Electronic Components and Technology Conference, 2004. Proceedings. 54th*, vol. 2. IEEE, 2004, pp. 1174–1180.

[4] R. Plieninger, M. Dittes, and K. Pressel, "Modern ic packaging trends and their reliability implications," *Microelectronics Reliability*, vol. 46, no. 9, pp. 1868–1873, 2006.

[5] M. Garey and D. Johnson, "Appendix: a list of np-complete problems," p. 202, 1979.

[6] H. Xiang, X. Tang, and D. Wong, "An algorithm for simultaneous pin assignment and routing," in *Proceedings of the 2001 IEEE/ACM international conference on Computer-aided design*. IEEE Press, 2001, pp. 232–238.

[7] M. M. Ozdal and M. D. Wong, "Simultaneous escape routing and layer assignment for dense pcbs," in *Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*. IEEE Computer Society, 2004, pp. 822–829.

[8] ——, "Algorithms for simultaneous escape routing and layer assignment of dense pcbs," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 25, no. 8, pp. 1510–1522, 2006.

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

[10] L. Luo, T. Yan, Q. Ma, M. D. Wong, and T. Shibuya, "B-escape: a simultaneous escape routing algorithm based on boundary routing," in *Proceedings of the 19th international symposium on Physical design*. ACM, 2010, pp. 19–25.

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

[12] H. Kong, T. Yan, and M. D. Wong, "Optimal simultaneous pin assignment and escape routing for dense pcbs," in *Design Automation Conference (ASP-DAC), 2010 15th Asia and South Pacific*. IEEE, 2010, pp. 275–280.

[13] M. Tanaka and Y. Nakagiri, "An approach to pin assignment in printed circuit board design," *ACM SIGDA Newsletter*, vol. 10, no. 2, pp. 21–33, 1980.

[14] H. N. Brady, "An approach to topological pin assignment," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 3, no. 3, pp. 250–255, 1984.

[15] T. Yan, Q. Ma, and M. D. Wong, "Advances in pcb routing," *Information and Media Technologies*, vol. 7, no. 2, pp. 535–543, 2012.

[16] T. Yan and M. D. Wong, "A correct network flow model for escape routing," in *Design Automation Conference, 2009. DAC'09. 46th ACM/IEEE*. IEEE, 2009, pp. 332–335.

[17] Q. Ma and M. D. Wong, "Np-completeness and an approximation algorithm for rectangle escape problem with application to pcb routing," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 31, no. 9, pp. 1356–1365, 2012.

[18] Y.-K. Ho, H.-C. Lee, and Y.-W. Chang, "Escape routing for staggered-pin-array pcbs," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 32, no. 9, pp. 1347–1356, 2013.

[19] 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," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 33, no. 4, pp. 571–584, 2014.

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

[21] J. McDaniel, Z. Zimmerman, D. Grissom, and P. Brisk, "Pcb escape routing and layer minimization for digital microfluidic biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 1, pp. 69–82, 2017.

[22] F. Jiao and S. Dong, "Ordered escape routing for grid pin array based on min-cost multi-commodity flow," in *Design Automation Conference (ASP-DAC), 2016 21st Asia and South Pacific*. IEEE, 2016, pp. 384–389.

[23] H. Kong, Q. Ma, T. Yan, and M. D. Wong, "An optimal algorithm for finding disjoint rectangles and its application to pcb routing," in *Design Automation Conference (DAC), 2010 47th ACM/IEEE*. IEEE, 2010, pp. 212–217.

[24] J.-T. Yan, "Layer assignment of escape buses with consecutive constraints in pcb designs," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 22, no. 3, p. 45, 2017.

[25] J.-T. Yan, T.-Y. Sung, and Z.-W. Chen, "Simultaneous escape routing based on routability-driven net ordering," in *SOC Conference (SOCC), 2011 IEEE International*. IEEE, 2011, pp. 81–86.

[26] C.-Y. Chin, C.-Y. Kuan, T.-Y. Tsai, H.-M. Chen, and Y. Kajitani, "Escaped boundary pins routing for high-speed boards," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 32, no. 3, pp. 381–391, 2013.

[27] W. Kumtong, P. Danklang, and W. Sriborrirux, "Pin set sequence selection guideline routing for printed circuit board routing," in *Knowledge and Smart Technology (KST), 2015 7th International Conference on*. IEEE, 2015, pp. 126–130.

[28] Q.-D. Ngo, Y. Hadj-Said, U. Maulik *et al.*, "Automatic generation of model for building energy management," *International Journal of Advanced Computer Science & Applications*, vol. 1, no. 7, pp. 442–454, 2016.

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

[30] K. Sattar and A. Naveed, "Ordered escape routing using network flow and optimization model," in *Automation, Robotics and Applications (ICARA), 2015 6th International Conference on*. IEEE, 2015, pp. 563–568.

[31] R. Fourer, D. Gay, and B. Kernighan, "Ampl: A modeling language for math. programming," *Duxbury Press/Brooks/Cole Publishing*, 2002.