

ACER: An Agglomerative Clustering Based Electrode Addressing and Routing Algorithm for Pin-Constrained EWOD Chips

Sean Shih-Ying Liu, Chung-Hung Chang, Hung-Ming Chen, and Tsung-Yi Ho, *Senior Member, IEEE*

Abstract—The problem of pin-constrained electrowetting-on-dielectric (EWOD) biochips becomes a serious issue to realize complex bio-chemical operations. Due to limited number of control pins and routing resources, additional Printed Circuit Board (PCB) routing layers may be required which potentially raises the fabrication cost. Previous state-of-the-art work has tried to develop a framework that uses a network-flow-based method for broadcast electrodeaddressing EWOD biochips. Nevertheless, greedily merging of electrical pins in previous works is at high risk of producing unroutable design. Routability should have higher priority than pin reduction. While previous works dedicated their effort on pin reduction, we have addressed our attention on routability of broadcast addressing. Experimental results demonstrate that taking routability into consideration can even have higher pin reduction. Viewed in this light, we present ACER, a routability driven clustering algorithm followed by escape routing using integer linear programming that effectively solves both pin merging and routing in broadcast addressing framework. Our proposed algorithm does not greedily focus on pin-reduction. Instead, routability is taken into consideration through agglomerative clustering. Compared to previous state-of-the-art, our proposed algorithm can further reduce required control pins by an average of 13% and route the design using 68% less wirelength.

Index Terms—Agglomerative clustering, broadcast addressing, electrowetting-on-dielectric, pin-constrained.

I. INTRODUCTION

ELECTROWETTING-ON-DIELECTRIC (EWOD) becomes a popular actuator for droplet-based digital microfluidic biochip applications. EWOD chips offer flexibility and efficiency to manipulate discrete fluidic droplet to realize several practical bio-system applications [2]–[5].

The structure of EWOD chips is a 2-D electrode array illustrated in Fig. 1(a). To realize an operation, droplets are moved across the electrodes to combine with other droplets or

separate into two droplets. Underneath each electrode, there exists an electrical pin, which is connected to a peripheral control pin through conduction wire. Wire can only be routed in orthogonal direction and cannot be crossed. For a simple design, routing from the electrical pins to the control pins can be realized in one routing layer. However, if routability is an issue, an additional routing layer is required to route the design, which consequently raises the fabrication cost.

The control pins are actuated by an external controller that supplies a limited number of signal ports. The controller generates a sequence of voltage value and fed to each control pin. The input sequence to each control pin is defined as activation sequence. Activation sequence is a combination of three values: value 1 denotes logic high value, value 0 denotes logic low value, and value X denotes don't care in which an arbitrary value can be assigned without changing original operation. The electrical pins located underneath the electrodes drive the movement of droplet. A logic high value attracts the neighboring droplet and a logic low value repels droplet away. By carefully planning the driving voltage of electrodes, droplets can be programmed to move in designated paths.

The assignment of electrodes to a control pin is generally referred as electrode addressing. The work done in [6] introduces the direct-addressing method which is a one-to-one assignment of an electrode to a control pin. However, due to limited number of signals that can be simultaneously generated from the controller, direct-addressing becomes less favorable in handling complex design. Fig. 1(b) is an example of direct-addressing.¹

In this regard, the number of control pins that can be simultaneously actuated is limited to the maximum number of signal ports supplied by the controller. This limitation of controller introduces the pin-constrained droplet-based digital microfluidic electrode addressing problem. In Fig. 1(b), 14 control pins are activated which requires a 16-bit controller to trigger the control pins. Pin-count of a design refers to the number of required activation sequences or number of control pins that needed to trigger the design.

¹The illustrated example has one routing track between two electrical pins. Routing tracks is set to 3 in simulation conducted in this paper and previous works. Squeezing three routing tracks creates difficulty to interpret the illustration. The purpose of this example is to demonstrate that poor electrode addressing leads to longer wirelength.

Manuscript received July 24, 2013; revised October 15, 2013, December 29, 2013, and March 13, 2014; accepted April 23, 2014. Date of current version August 18, 2014. This paper was recommended by Associate Editor Y. Chen.

S. S.-Y. Liu and H.-M. Chen are with the Institute of Electronic, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: sniealu.ee96g@g2.nctu.edu.tw; hmchen@mail.nctu.edu.tw).

C.-H. Chang is with the Reliability Department, Macronix Company Ltd., Hsinchu 300, Taiwan (e-mail: d19880819@gmail.com).

T.-Y. Ho is with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701, Taiwan (e-mail: tyho@csie.ncku.edu.tw).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCAD.2014.2329415

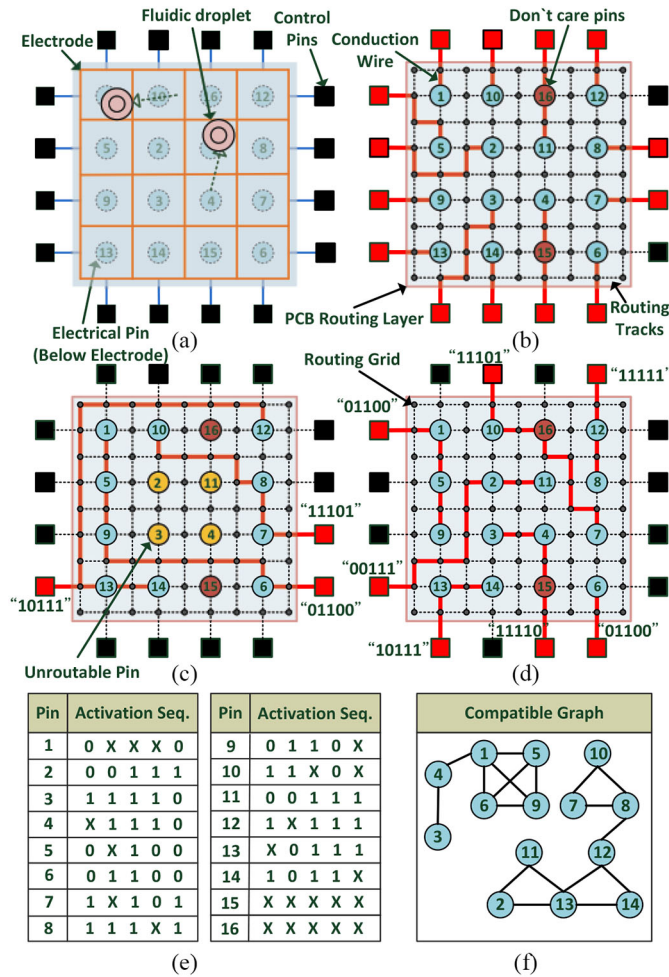


Fig. 1. Comparison on routing result of different electrode addressing framework. (a) Schematic view for EWOD Chips. (b) Routing of electrical pins after direct-addressing. (c) Routing of electrical pins after greedily minimizing total number of required control pins. (d) Routing of electrical pins after routability driven pin merging. (e) Activation sequence for each electrical pin. (f) Compatible graph constructed from the given activation sequences.

A. Using Broadcast Addressing to Satisfy the Pin-Constraint

To satisfy the pin constraint for a given design, the work done in [7] proposed the broadcast addressing method to assign several electrical pins to one control pin. In [7], a compatible graph of electrical pins is constructed indicating which sets of electrical pins can be activated by a common activation sequence. A clique partition method is performed on the compatible graph to minimize number of required activation sequences. Fig. 1(f) is an illustration of compatible graph in which each node represents an electrical pin on EWOD chip. In Fig. 1(f), the illustrated example can be triggered by a minimum of five activation sequences.

However, a given design can only be realized if all the electrical pins can be successfully routed. Minimizing total pin count may not produce a routable design. In Fig. 1(c), clique with largest degree in compatible graph is routed on each iteration. Clique connecting to pin 1, 5, 6, and 9 is first routed followed by clique 7, 8, and 10 and clique 12, 13, and 14. However, this particular electrode addressing produces an unroutable design after routing clique 12,

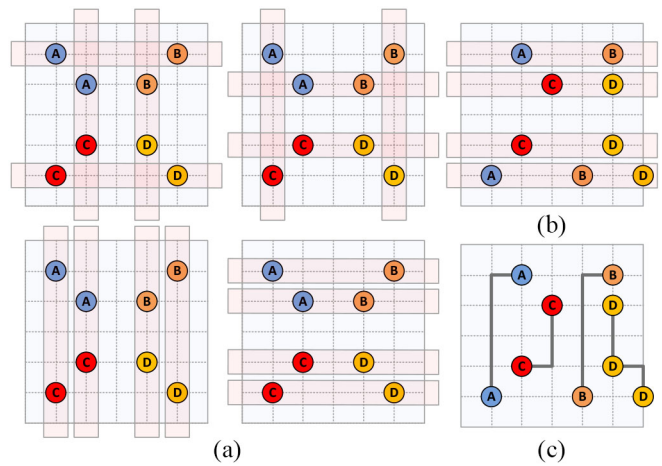


Fig. 2. Illustration of two scenarios in which finding minimum number of global tracks is ineffective. Electrical pins with same alphabet represents electrical pins are compatible. (a) Multiple solutions of minimum number of global tracks and none of the solutions have mergeable pins. (b) Unique solution of minimum number of global tracks but none of the electrical pins can be merged on any global track. (c) Feasible solution to the example illustrated in (b).

13, and 14. This shows greedily reducing pin count creates longer detour and impairs routability of the design. The electrode addressing in Fig. 1(d) produces a routable design even though it requires seven activation sequences which is two more than the optimal solution. However, keep in mind that controllers supplies 2^n number of ports. An electrode addressing that requires seven activation sequences is just as cost effective to electrode addressing that requires five activation sequences.

B. Previous Works on Broadcast Addressing and Routing

Thus, despite the benefits gained from broadcast addressing method, routability becomes the bottleneck to realize a design. The work done in [1] proposed a network flow algorithm in attempt to minimize number of pin-counts. The work done in [8] extends the work by further considering the variation of driving voltage.

In [1], a network flow algorithm is proposed to search for an initial solution. This algorithm greedily reduces pin-count by finding the minimum number of global tracks that covers all of the electrical pins. This greedy approach suffers two major drawbacks. First, the initial solution obtained from network flow is not unique, a specific solution must be chosen out of multiple solutions in order to proceed to the next stage. Second, routability of electrical pins is completely overlooked. Fig. 2 is an illustration on two common scenarios where finding minimum number of global tracks is ineffective. Fig. 2(a) illustrates not only there are multiple solutions in finding minimum number of global tracks, all solutions are infeasible. Fig. 2(b) illustrates even there exists a unique solution, the obtained solution is still infeasible. Fig. 2(c) is a feasible solution to the example illustrated in Fig. 2(b). It can be observed that finding minimum tracks as initial solution not only limits the solution space, its complete negligence to routability severely degrades its effectiveness.

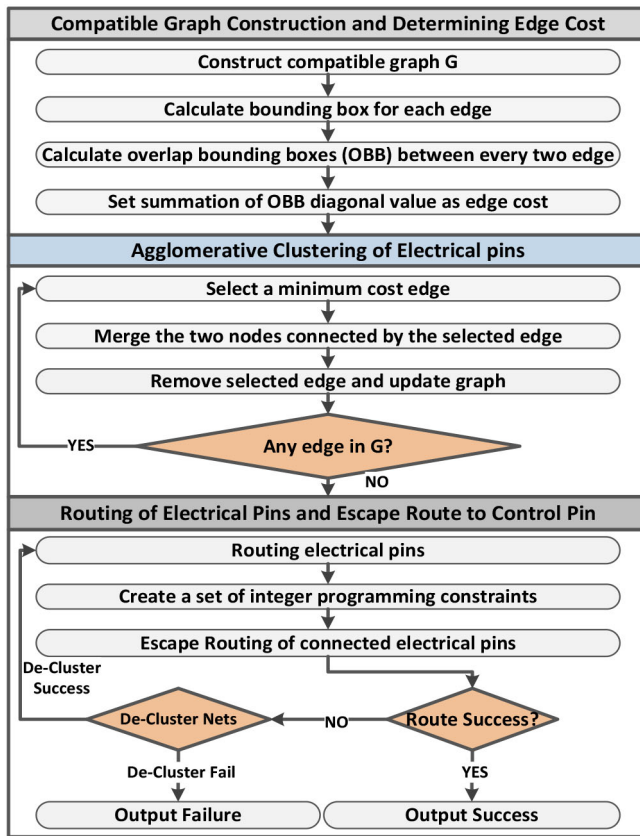


Fig. 3. Flow chart of the proposed algorithm. The proposed algorithm is divided to three stages. First stage constructs compatible graph and determines edge cost. Second stage applies agglomerative clustering on the electrical pins and determines routing order. Third stage routes the electrical pins and then routes the connected electrical pins to peripheral control pin by formulating as escape routing problem using a set of integer programming constraints.

C. Our Contributions

While previous works focus on minimizing the pin-count, we suggest that it is more important to focus on routability of the design. Greedily minimizing pin-count may produce unroutable design as illustrated in Fig. 1(c). Routability should be regarded as primary concern when minimizing pin-count. To the best of our knowledge, this is the first work that addresses routability on broadcast addressing framework. In brief, our contributions can be summarized as follows.

- 1) In this paper, we propose ACER to effectively handle both electrode addressing and routing of electrical pins. ACER is a routability driven electrode addressing algorithm based on agglomerative clustering.
- 2) Unlike previous works, our proposed method relieves burden of resolving congestion at routing stage by determining a better routing order and avoids unnecessary global track identification.
- 3) Experimental result shows that by considering routability, pin-count can be further reduced compared to the approaches that greedily minimize pin-count.
- 4) Unlike previous work which uses rip and reroute framework to resolve congestion, our approach relies on de-cluster and reroute approach.

Our proposed framework consists of two routers. The first router is based on Lee's algorithm [9] that connects the

clustered electrical pins. The second router is an escape router that connects clustered electrical pins to peripheral control pins.

Fig. 3 shows the flow chart of our proposed methodology. Our proposed framework is divided to four stages: Compatible graph construction, agglomerative clustering, routing of electrical pins and escape routing of electrical pins. The remaining parts of this paper are organized as follows. Section II formulates the problem. Section III describes agglomerative clustering of electrical pins. Section IV describes the routing of electrical pins. Section V describes how to route the connected electrical pins to peripheral control pin using integer linear programming based escape routing. Section VI presents the experimental result. Finally, Section VII concludes the paper and introduces possible future works.

II. PROBLEM FORMULATION AND DEFINITIONS

The operations on an EWOD chip are defined by the activation sequences fed to each control pin. The activation sequence controls the voltage value of the control pin on each time frame. Here we formally define the problem of pin-constrained electrode addressing and routing for broadcast-addressing framework as follows.

A. Pin-Constrained Electrode Addressing and Routing for Broadcast-Addressing EWOD Design

Given a set of electrical pins $P = \{p_1, p_2, \dots, p_k\}$ and a set of activation sequences $A = \{a_1, a_2, \dots, a_k\}$. Each electrical pin p_i has a corresponding activation sequence a_i . Minimize total pin count by triggering multiple electrical pins with mutual compatible activation sequence while ensuring that every electrical pin can be successfully routed to designated control pin without crossing.

Before the detail of the algorithm is introduced, we define the following terms that are used in graph representation and the proposed algorithm.

Definition 1: A node n_I on compatible graph represents a set of electrical pins $I = \{p_{i1}, p_{i2}, \dots, p_{in}\}$ that can be triggered by one activation sequence a_k .

Definition 2: Two nodes are said to be mutually compatible if and only if all electrical pins represented by two nodes can be triggered by one activation sequence without affecting original operation.

Definition 3: An edge, e_{I-J} , exists between node n_I and node n_J if and only if two nodes are mutually compatible.

Definition 4: A bounding box of an edge, BB_{I-J} , is defined as the minimum spanning rectangle that covers all electrical pins represented by two nodes.

Definition 5: An overlapping bounding box, $OBB_{I-J:K-L}$, is defined as the cross section area over the two bounding boxes BB_{I-J} and BB_{K-L} spanned by edge e_{I-J} and edge e_{K-L} .

III. ROUTABILITY DRIVEN CLUSTERING OF ELECTRICAL PINS

In this section, a compatible graph is constructed to represent compatibility of the electrical pins. The merging of the

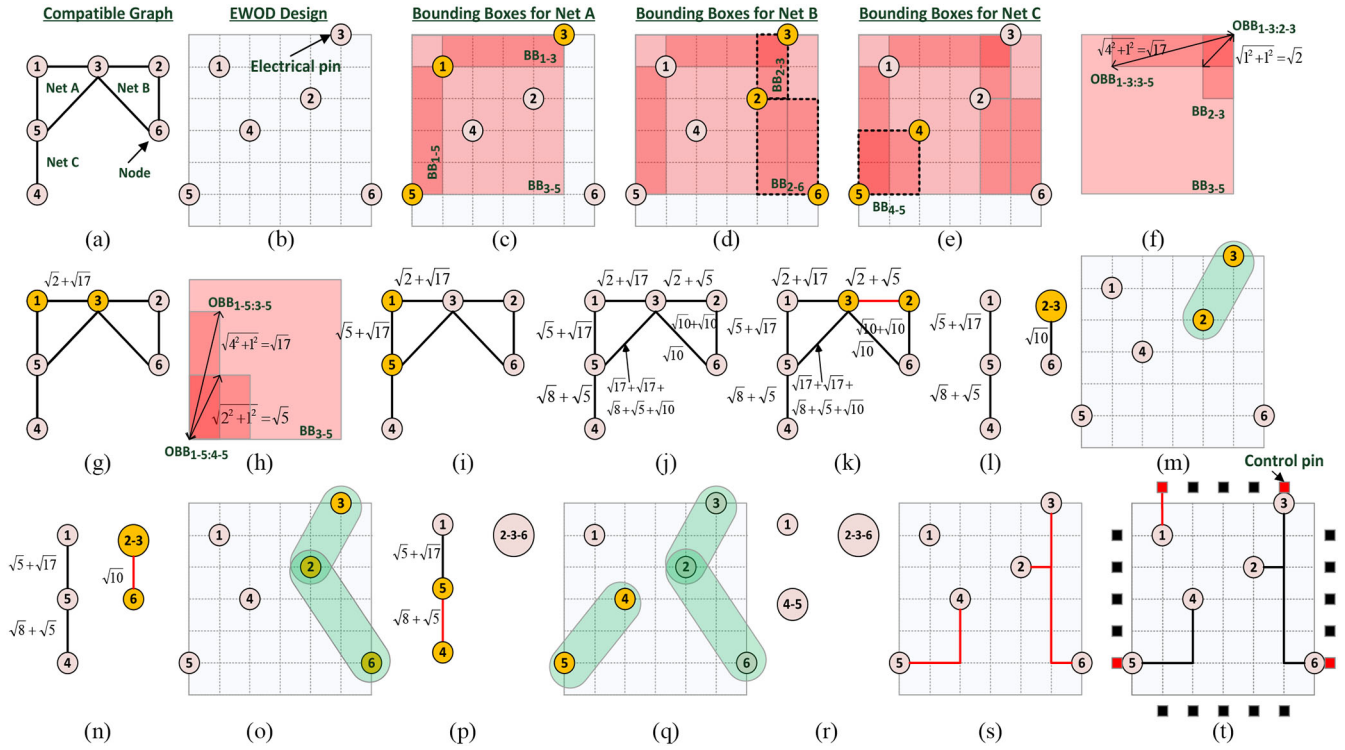


Fig. 4. Illustration of an example from compatible graph construction to escape routing of connected electrical pins. (a) Given compatible graph. (b) Position of electrical pins. (c) Bounding boxes of Net A. (d) Bounding boxes of Net B. (e) Bounding boxes of Net C. (f) and (g) Overlapping bounding boxes to edge e_{1-3} . (h) and (i) Overlapping bounding boxes to edge e_{1-5} . (j) Compatible graph with edge cost. (k)–(r) Agglomerative clustering of electrical pins. (s) Routing of electrical pins in order of 2 – 3, 2 – 3 – 6, and 4 – 5. (t) Escape routing of connected electrical pins.

electrical pins is achieved by applying agglomerative clustering. Edge cost in the compatible graph is determined to reflect routability of the electrical pins. Fig. 4 illustrates an example on constructing compatible graph and escape routing of the electrical pins.

A. Construction of Compatible Graph

The compatible graph is constructed based on the activation sequence of electrical pins. Initially, an electrical pin is represented by one node [see Fig. 4(a) and (b)]. If two electrical pins can be triggered by the same activation sequence, then an edge exists between the two nodes. The compatible graph updates the compatibility among nodes after every merging of electrical pins.

An n -degree clique in compatible graph represents that all of the connected nodes in a clique are mutually compatible and can be triggered by a single activation sequence. Since each cluster must be a n -degree clique, the problem of finding minimum number of clusters is equivalent to the minimal clique cover problem. The minimum clique cover problem is an NP-Complete problem [10]. In addition, finding minimum set of cliques neglects routability of the design which is likely to produce unroutable design as illustrated in Fig. 1(c). To consider routability while merging electrical pins, techniques in agglomerative clustering are applied.

B. Agglomerative Clustering of Electrical Pins

Agglomerative clustering is a category in hierarchical clustering which minimizes a given objective cost by clustering

elements in a bottom-up fashion [11], [12]. Clustering can aim to minimize a certain objective cost. Given a graph $G(V, E)$ in which V represents a set of vertices and E represents a set of edge. A cost is given on each edge in E . In this paper, agglomerative clustering begins by setting each vertex as an individual cluster. Each cluster begins to expand by merging adjacent cluster connected by an edge. The objective is to minimize total number of clusters by selecting a set of edges with a summation of minimum cost. Different edge cost can be configured to meet a variety of specifications and constraints. In application of EWOD design, the cluster is a node on the compatible graph and we try to model routability in the edge cost.

To merge electrical pins, edge with the least cost in the compatible graph is selected and the two connected nodes are merged. New node is created to represent the merged node. Adding new node to the compatible graph consists of three operations.

- 1) A new parent node is created to replace the two merged nodes.
- 2) New edges are created to connect the new parent node to mutual compatible nodes.
- 3) The merged nodes and all of the edges connected to merged nodes are removed from the compatible graph.

A new edge, e_{k-m} , that connects node n_k and n_m is created if and only if both child nodes of n_k have an edge connecting to n_m .

Fig. 4(c)–(e) is an illustration of bounding boxes of net A, B, and C. Fig. 4(j)–(r) is an illustration of agglomerative

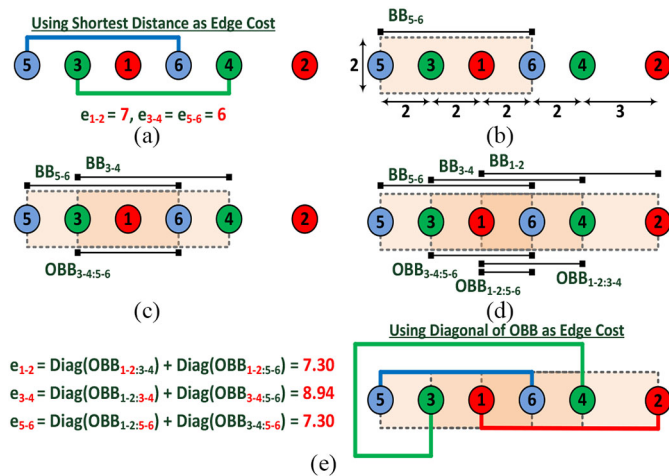


Fig. 5. Comparison on routability using diagonal distance of overlapping bounding boxes (*OBB*) and shortest distance as edge cost. Nodes that are marked with same color represents both nodes are mutually compatible. (a) Using shortest distance as edge cost. (b) Illustration of bounding box for edge e_{5-6} . (c) Illustration of bounding box for edge e_{3-4} and overlapping bounding box $OBB_{3-4-5-6}$. (d) Illustration of bounding box for edge e_{1-2} , overlapping bounding box $OBB_{1-2-5-6}$, and overlapping bounding box $OBB_{1-2-3-4}$. (e) Using diagonal distance of *OBB* as edge cost.

clustering. In Fig. 4(k), edge e_{2-3} is selected since it has the least cost on the graph. The merging of n_2 and n_3 is illustrated in Fig. 4(l). Edge e_{3-6} , e_{2-6} , node n_2 and n_3 are removed from the graph and new node n_{2-3} and new edge e_{2-3-6} are added to the graph. The process of clustering repeats until there is no more edge in the compatible graph. In Fig. 4(r), there are only three nodes left in the compatible graph, which means clustering is completed and three control pins are required.

In Fig. 4(a), without considering physical location of the electrical pins, a possible clustering solution can be n_{1-3-5} , n_{2-6} , and n_4 . However, such solution consumes additional routing resources that impairs routability. By reflecting routability in the edge cost, solution obtained after clustering can aim to maximize routability of the design.

C. Determining Edge Cost

In this subsection, we explore several methods to determine edge cost and analyze the impact of edge cost on routability of the design. Fig. 5(a) illustrates a common scenario in EWOD designs in which several compatible electrical pins are aligned in one routing track. In the illustrated example, the design is not routable if p_3 and p_4 are not routed last. In Fig. 5(a), electrical pins are merged based on shortest distance, p_1 and p_2 are merged last in which create an unroutable design. The question is not how much routing resource a pair of electrical pins consumes, it is about how much routing resource that may be potentially valuable to other pairs of electrical pins.

In this regard, Fig. 5(b)–(d) constructs *BB* and *OBB* for each pair of electrical pins. The *OBB* can be identified using line sweep method [13]. The general concept of line sweep method is to sort the x_{min} and x_{max} coordinates of bounding boxes in nondecreasing order. A virtual line sweeps from the minimum x-coordinate to check whether if there are overlaps in y-coordinate. When sweep line reaches to the end of sorted

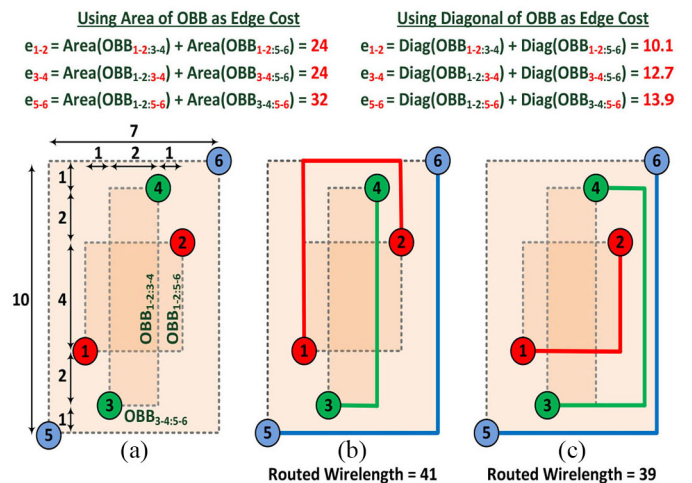


Fig. 6. Comparison on routed wirelength using area and diagonal of overlapping bounding boxes as edge cost. Nodes that are marked with same color represents both nodes are mutually compatible. (a) Original pin location and corresponding overlapping bounding boxes. (b) Routing order based on shortest distance. (c) Routing order based on diagonal distance of *OBB*.

x-coordinate list, all of the overlapping bounding boxes can be identified.

For *BB* with all of its electrical pins on the same y-coordinate, the *BB* is expanded by one routing track upward and one routing track downward and vice versa for electrical pins with same x-coordinate. The *BB* represents the potential routing region for a pair of electrical pins and *OBB* represents potential routing region shared by two pairs of electrical pins. In Fig. 5(e), merging order is determined using summation of diagonal length of every *OBB* to a pair of electrical pins.² The obtained merging order merge p_3 to p_4 after p_1 to p_2 and p_5 to p_6 are merged. This merging order in this example produces a routable design.

The reason to use diagonal length rather than area of *OBB* is because area creates ambiguity for *OBB* s with same area but different aspect ratio. Fig. 6 illustrate an example that using diagonal is better than area. In Fig. 6, BB_{1-2} covers a 4×4 area to BB_{5-6} and 2×4 BB_{3-4} . Using area of *OBB* to determine edge cost, edge e_{1-2} has an edge cost equal to 24 which is equivalent to e_{3-4} . Using diagonal value of *OBB* to determine edge cost, e_{1-2} has cost equal to $\sqrt[2]{32} + \sqrt[2]{20} = 10.1$ and e_{3-4} has cost equal to $\sqrt[2]{68} + \sqrt[2]{20} = 12.7$. Merging p_3 and p_4 prior to p_1 and p_2 requires two additional unit length.

Fig. 7 evaluates the effectiveness of edge cost by comparing pin count to total routed wirelength on testcase random-5. The merging of electrical pins is performed under the framework of agglomerative clustering. Edge cost is configured using diagonal length of *OBB*, area of *OBB*, HPWL of *OBB* and shortest distance between two nodes. Starting from initial 100 pins, pin count is gradually reduced on each step of clustering. The nature of agglomerative clustering is that it always uses the least objective cost to merge the next electrical pin.

²There is no tie-breaking strategy during clustering stage. If two edges have same cost, it implies that our methodology regard two edges have the same impact toward routability. Hence, merging priority depends on their order in queue.

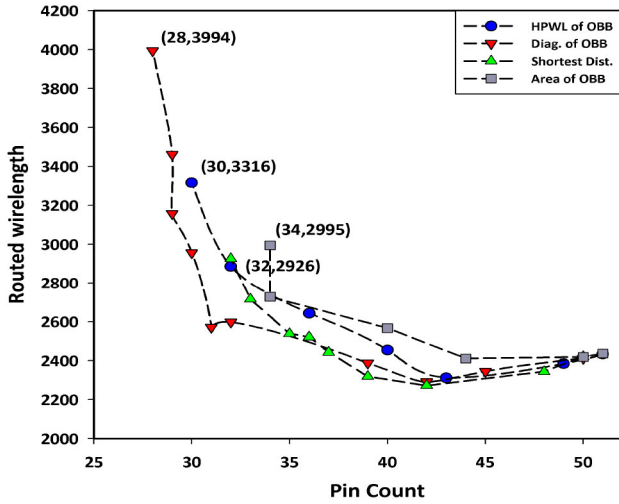


Fig. 7. Comparison on tradeoff between reductions in number of activation sequences versus total routed wirelength for testcase random-5.

Despite different configurations of edge cost, a smooth trade-off between pin count and routed wirelength can be observed on all configurations. From Fig. 7, it can be observed that using diagonal of *OBB* achieves most reduction in number of required pins. As a comparison, the work done in [1] uses 7965 unit wirelength yet still requires 37 control pins. Unfavorable electrode addressing for routing and brute force rip and reroute is probably the culprit for nearly double amount of the routed wirelength in [1].

IV. ROUTING OF ELECTRICAL PINS

When agglomerative clustering is completed, the actual routing path that connects all of the electrical pins is still undetermined. A maze router based on Lee's Algorithm [14] is implemented to connect the merged electrical pins. The routing order for each pair of electrical pins is determined based on its clustering order. In Fig. 4(q) and (r), the clustering order of electrical pins is $(p_2-p_3) \rightarrow (p_2-p_3-p_6) \rightarrow (p_4-p_5)$. The clustering order means that electrical pin p_2 and p_3 are routed first. After p_2 and p_3 are routed, it then searches for least cost path connecting to p_6 . After p_2 , p_3 , and p_6 are connected, p_4 and p_5 are routed.

To increase routing success rate, the concept on pricing routing resources in global routing [15], [16] is adopted to increase routing success rate. The concept of pricing routing resource is to avoid potentially congested region by guiding router to avoid grids with higher cost. By applying the same concept in EWOD routing, a simplified yet still effective method is adopted to price each routing grid. In our approach, every routing grid spanned by the *BB* is updated with a penalty cost inversely proportional to the diagonal of *BB*. The penalty cost is controlled by an empirical parameter β to adjust the quality between detour wirelength and routability. Equation (1) calculates the penalty cost on each grid. In our configuration, β is set to 8 for all designs.

Fig. 8 illustrates an example using testcase amino-acid-1 that how penalty cost can prevent routing failure. In Fig. 8(a),

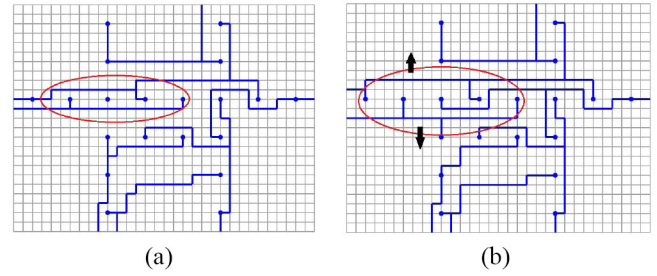


Fig. 8. Impact on routability with the penalty cost using testcase [amino-acid-1]. Wires are detoured away from potentially congested region. (a) Unroutable result without the penalty cost. (b) Routable result with the penalty cost.

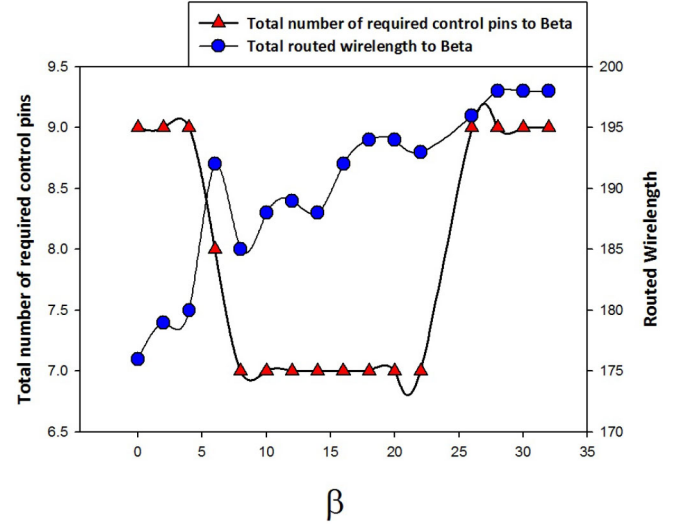


Fig. 9. Impact on routed wirelength and number of required control pins with different adjustment of β in testcase [amino-acid-1].

routing failure occurred since electrical pins are routed regardless of the congested region. By adding penalty in Fig. 8(b) to each routing grid, routing path detours away from congested region and creates additional space for electrical pins at congested region to escape

$$p(x, y) = \sum_{\forall(x, y) \in BB_j} \frac{\beta}{\text{Diag}(BB_j)}. \quad (1)$$

Fig. 9 plots the trade-off curve between reduction in number of activation sequences and total routed wirelength in response to different adjustment of β . The larger value β , the longer detour from the congested regions during maze routing. However, over-powering β creates longer detour which creates more routing obstacles when routing electrical pins.

V. ESCAPE ROUTING OF CONNECTED ELECTRICAL PINS TO CONTROL PINS

When all electrical pins are connected, the upcoming task is to route the connected electrical pins to a peripheral control pin. Fig. 4(t) is an illustration on escape routing of a set of connected electrical pins. The objective is to route every merged electrical pins to a control pin using minimal routed wirelength. Similar problem formulation can be seen in PCB escape routing [17], [18]. However, in the scenario

of broadcast-addressing EWOD design, it is an escape routing problem for multiple nets with each net containing a set of prerouted paths. Thus, techniques adopted in PCB escape routing are not directly applicable. In this section, we first formulate the multisource multisink escape routing problem, then we introduce two methods based on ILP formulation and maze routing to solve the problem.

A. Multisource Multisink Escape Routing Problem

Given a mesh based graph with a set of grid points $G = \{g_1, g_2, \dots, g_i\}$, a set of nets $N = \{n_1, n_2, \dots, n_i\}$ and a set of target grid points $T = \{t_1, t_2, \dots, t_m\}$. Each net n_i has a set of source grid points $S_{n_i} = \{s_{n_i,1}, s_{n_i,2}, \dots, s_{n_i,k}\}$ and a set of routed edges $E_{n_i} = \{e_{n_i,1}, e_{n_i,2}, \dots, e_{n_i,m}\}$ that connects all of routed grid points in net n_i . Route every net n_i from any arbitrary source grid point $s_{n_i,k}$ belong to net n_i to any arbitrary target grid point such that no routing paths can be crossed.

- 1) $G = \{g_1, g_2, \dots, g_i\}$ is a set of grid points correspond to the EWOD design.
- 2) $S_{n_i} = \{s_{n_i,1}, s_{n_i,2}, \dots, s_{n_i,k}\}$ is a set of routed grid points for net n_i .
- 3) $N = \{n_1, n_2, \dots, n_i\}$ is a set of nets. A net n_i includes a set of routing paths that connects a set of grid points $S_{n_i} = \{s_{n_i,1}, s_{n_i,2}, \dots, s_{n_i,k}\}$.
- 4) $T = \{t_1, t_2, \dots, t_m\}$ is a set of control pins. Each control pin has one edge connecting to a grid point in G .
- 5) $f(u, v)$ denotes an outward flow from g_u to g_v . A positive flow $f(u, v)$ exists from grid point u to grid point v if and only if u and v are directly adjacent. A positive flow from u to v represents a negative flow from v to u . In other words, $f(u, v) = -f(v, u)$.

B. Simultaneous Escape Routing Based on ILP Formulation

The integer linear programming formulation of the multisource multisink escape routing problem is defined as follows. Given a set of nets, a set of electrical pins and a set of control pins, minimize the total summation of absolute value of flow for all grid points. The objective function is described in (2)³

$$\min \sum_{\forall v \in Adj[u]} |f(u, v)|, \forall u \in G \quad (2)$$

$$s.t. \sum_{\forall v \in Adj[u]} f(u, v) = 0, \forall u \notin S, T \quad (3)$$

$$\sum_{\forall s_{n_i,j} \in S_{n_i}} \sum_{\forall v \in Adj[s_{n_i,j}]} f(s, v) = 1, \forall n_i \in N \quad (4)$$

$$\sum_{\forall v \in Adj[t]} |f(t, v)| = N, \forall t \in T \quad (5)$$

$$\sum_{\forall v \in Adj[u]} |f(u, v)| \leq 2, \forall u \notin S \quad (6)$$

$$|f(u, v)| \leq 1, \forall u \in G, \forall v \in G. \quad (7)$$

³Even though the problem formulation contains absolute value constraints, it is still integer linear programming because one absolute value of integer constraint can be replaced by two integer linear constraints.

Equation (3) defines the conservation of flow. For a given grid point that is not a source grid point or a target grid point, the inward flow must equal to its outward flow. Equation (4) defines the net flow for a given net n_j must equal to 1. Equation (4) forces every net to have an outward flow. When an outward flow is initialized for each net, (3) forces this outward flow from each net to be escaped to a target grid point.

Equation (5) defines the terminating condition. When the summation of absolute flow value of every control pin equals to the total number of nets, it represents that every net is successfully escaped to a control pin. Equation (6) restricts that no routing paths can be crossed by defining that the absolute value of flow for any given grid point less than or equal to 2. Equation (7) defines the capacity constraint such that the absolute value of flow between any two electrical pins is less than or equal to 1.

Fig. 10 illustrates an example of escape routing. In Fig. 10(a), the initial input contains two nets. Net 1 contains a set of routed grid points $S_{n_1} = \{g_{31}, g_{32}, g_{33}, g_{40}, g_{49}\}$ and Net 2 contains routed grid points $S_{n_2} = \{g_{35}, g_{44}, g_{51}, g_{52}, g_{53}, g_{62}, g_{71}\}$. In Fig. 10(b), (4) forces an outward flow of Net 1 from g_{31} to g_{30} and an outward flow from g_{53} to g_{54} . In Fig. 10(b), g_{30} and g_{54} violates (3). Thus, the outward flow is propagated from g_{30} to g_{29} and from g_{54} to g_{54} to a target grid point in Fig. 10(c). Equation (3) continues to propagate the outward flow initialized from each net until every outward flow reached to a target grid point. In Fig. 10(d), all constraints are satisfied and ILP escape routing is terminated.

C. Sequential Escape Routing Based on Maze Routing

Maze routing is a sequential routing technique, which can be easily adopted to solve the multisource multisink escape routing problem. The maze routing algorithm is a routing algorithm based on Lee's Algorithm [14]. A brief explanation of Lee's algorithm is summarized as follows. The algorithm begins from the source location. For all neighboring grids that are reachable from the source location within a unit distance, each grid is marked with the unit distance. Using the distance from marked grids, find all unmarked neighboring grids adjacent to the marked grid and update its distance. The algorithm progressively expands from the source location until target location is found. The original Lee's algorithm assumes one source and one target. By initiating the algorithm from multiple source locations instead of one and terminate the algorithm when any one of the multiple target locations is found, the original Lee's algorithm can be adopted to solve the multisource multisink escape routing problem.

However, a maze router route each net sequentially while solving the problem using ILP simultaneously considers all nets. Thus, a maze router guarantees to find a solution for each net but not does not guarantee to find a solution to route all nets. The reason is because maze router heavily depends on net ordering and there is no way to find a net ordering that guarantees to find a solution that can successfully route all nets if there is one that exists [19].

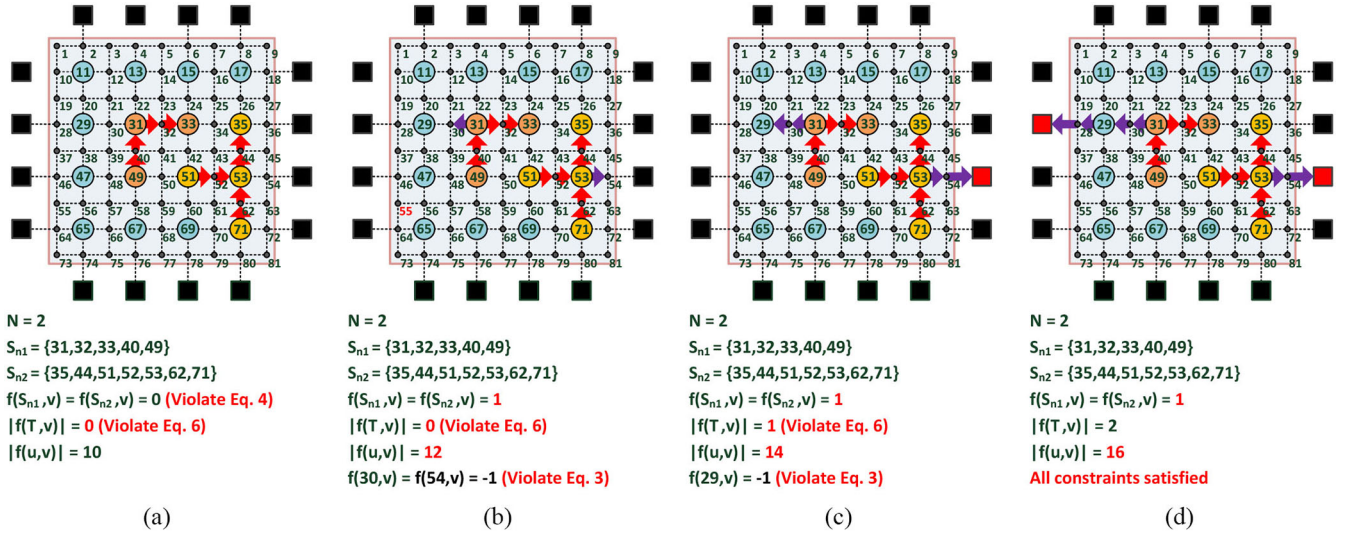


Fig. 10. Example of simultaneous escape routing based on ILP formulation. (a) Initial input with two nets. The red lines denote the prerouted paths. (b) Two outward flows are initialized from Net 1 and Net 2. (c) Outward flows are propagated away from Net 1 and Net 2. (d) Each outward flow reached a target point and escape routing terminates since all constraints are satisfied.

TABLE I

TIME DURATION FOR EACH PHASE OF THE PROPOSED FRAMEWORK AND PIN COUNT ANALYSIS AFTER ELECTRICAL PIN CLUSTERING AND AFTER ILP ESCAPE ROUTING (E.R. STANDS FOR ESCAPE ROUTING, CLUS. STANDS FOR CLUSTERING)

Case	Time			Pin Count		
	Stage 1 ¹	Stage 2 ²	Stage 3 ³	Aft. Clus.	Aft. E.R.	# of De-Clust.
amino-acid-1	0.00	0.00	0.26	7	7	0
amino-acid-2	0.00	0.00	0.27	9	9	0
protein-1	0.00	0.00	0.89	10	10	0
protein-2	0.01	0.01	0.92	23	31	5
dilution	0.00	0.00	1.58	23	23	0
multiplex	0.00	0.00	1.80	10	10	0
Random-1	0.00	0.00	0.56	9	9	0
Random-2	0.01	0.01	1.68	15	16	1
Random-3	0.01	0.01	3.65	26	27	1
Random-4	0.02	0.03	29.95	43	44	1
Random-5	0.00	0.00	67.78	28	28	0
Random-6	0.00	0.01	120.21	35	35	0
Random-7	1.01	7.05	230.77	43	45	1

¹ Stage 1: Time duration from parsing input till electrical pin clustering is completed.

² Stage 2: Time duration after electrical pin clustering till routing of electrical pins is completed.

³ Stage 3: Time duration after routing of electrical pins till escape routing of merged electrical pins is completed.

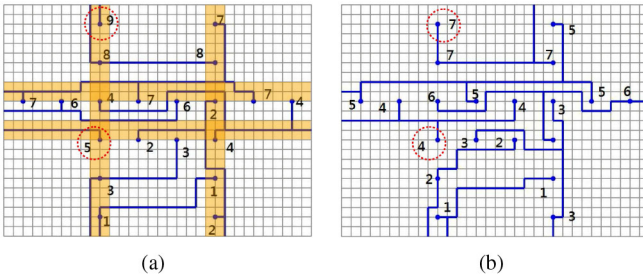


Fig. 11. Routing result for testcase [Amino-Acid-1] compared with work [1] for test case [amino-acid-1]. (a) Our work requires seven activation sequences using 185 unit of routed wirelength. (b) Work done in [1] requires nine activation sequences using 190 unit of routed wirelength.

VI. EXPERIMENTAL RESULT

In this section, experimental result of the proposed algorithm is presented. The entire algorithm is implemented with standard C++ language and experimented on AMD Athlon Dual Core machine operating at 2.6 GHz. IBM CPLEX

v12.2 [21] is used to solve the integer linear programming problem in escape routing. Input designs are obtained from [1]. There exist three routing grids between every two electrical pins and the don't care electrical pins can be arbitrary merged with any other electrical pins. Since the binaries from [1] and [6] cannot be obtained, we refer the result including execution time from [1] and reimplemented the work done in [6].

A. Analysis on Routing Behavior

From figures presented in [1], several shortcomings can be observed. Fig. 11(a) redraws the routing result from the figure presented in [1] and compares to our routing result illustrated in Fig. 11(b). Electrical pins within the red dashed circle are the don't care pins.

The orange stripes in Fig. 11(a) represents the initial tracks identified in [1]. After initial tracks are identified, electrical pins 1, 5, and 9 on the left vertical track are routed first

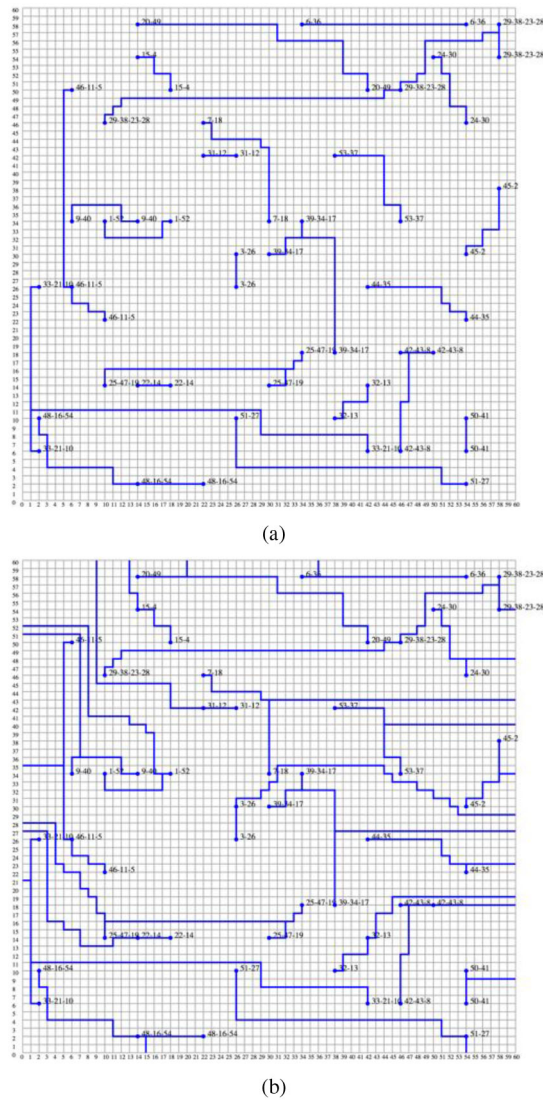


Fig. 12. Merging and routing result for testcase [dilution] using the proposed algorithm. (a) Merging result after agglomerative clustering (routed wirelength = 492 unit). (b) Routing result after escaped routing (routed wirelength = 820 unit).

[1, Fig. 7(c)]. However, electrical pin 5 and 9 are don't care pins which can be merged with other electrical pins. In our proposed algorithm, active electrical pins are dealt prior to the don't care pins. The don't care pins connects to nearest routed wire after active electrical pins are merged and routed.

Generally speaking, the work done in [1] is on a route-first-merge-later basis. Such framework leaves the burden of resolving congestion to the subsequent rip and reroute framework. Regarding Fig. 11(a), there are no obvious gain by identifying a set initial tracks. In contrast, our work is on a merge-first-route-later basis. Routability is considered at the merging stage by determining better merging priority and setting penalty cost in congested region.

Table I reports time duration of each stage using ILP escape routing. We divide total execution time into three stages. The first stage in the second column shows the time duration from parsing input until electrical pin clustering is completed. The second stage in the third column shows the time duration after

electrical pin clustering until routing of electrical pin is completed. The third stage in the fourth column shows the time duration after routing of electrical pins until ILP escape routing of merged electrical pins is completed. It can be observed that ILP escape routing takes most of the execution time and routing of electrical pins has negligible run time.

The fifth column shows pin-count after clustering of electrical pins which can be regarded as the lower bound of pin-count. The sixth column shows final pin-count after ILP escape routing. The last column shows number of de-clustering required for each testcase. In our framework, we de-cluster nets based on diagonal distance of clustered electrical pins, so each de-cluster iteration may de-cluster more than one net. Note that a straightforward approach that de-cluster one net on each iteration is also feasible and achieves equivalent solution quality. On all 13 testcases, only five cases require de-clustering and four cases only require one iteration of de-clustering.

Fig. 12 uses testcase dilution to show intermediate steps of our proposed algorithm. Fig. 12(a) is the result after electrical pins are merged and routed. Fig. 12(b) takes the merging result as input and performs multisource and multisink escape routing described in Section V.

Regarding to Fig. 11, it may seem the work done in [1] route each don't care electrical pins independently. However, in testcase amino-acid-2, there are 8 don't care electrical pins out of 24 electrical pins. The rest of 16 electrical pins can be formed to a minimum of seven independent compatible cliques. To achieve 11 pin-count on amino-acid-2 as reported in [1], the don't care pins are impossible to be routed independently. To make a fair comparison, we have double-checked with Huang *et al.* [1] to make sure the simulation constraints conducted in this paper is exactly the same in [1]. In other words, this paper and the work done in [1] merge and route all don't care electrical pins.

B. Comparison with Previous Works on Broadcast Addressing and Direct Addressing

To demonstrate the effectiveness of routability driven clustering, we compare our algorithm with [1] and direct addressing method [6]. For direct addressing method [6], escape routing is applied to route each electrical pin to a control pin.⁴ In Table II, #E denotes total number of electrical pins. #WL denotes total routed wirelength, #Pin denotes pin count and the number inside the parenthesis up scale pin count to nearest 2^n indicating the required n -bit controller. In Table II, our work has less pin count on 7 out 13 designs compared to [1] and route all the designs with less wirelength. The rest of the six designs, only three designs requires higher n -bit controller. Our proposed framework does not include a rip and reroute engine and congestion is minimized at clustering stage. The significantly more routed wirelength in [1] indicates less favorable electrode addressing to router and aggressive rip up

⁴The reimplementation of direct addressing [6] in [1] reports several route failure testcases. However, our reimplementation of [6] have successfully routed every design that was available to us. This may be due to the difference in escape routing engine.

TABLE II

COMPARISON WITH PRIOR WORKS [6] AND [1] ON TOTAL NUMBER OF REQUIRED CONTROL PINS AND ROUTED WIRELENGTH. ILP IS ESCAPE ROUTING USING INTEGER LINEAR PROGRAMMING AND MAZE IS ESCAPE ROUTING USING MAZE ROUTING ALGORITHM. THE NUMBER INSIDE THE PARENTHESIS SCALES PIN COUNT TO NEAREST 2^n . WE REFER THE RESULTS FROM [1] WHICH IS PERFORMED ON DIFFERENT PLATFORM (G.M. STANDS FOR GEOMETRIC MEAN AND TIME IS MEASURED IN SECONDS)

benchmarks	Chip Size	#E	Direct Addressing [6]			Huang et. al. [1]*			Our (ILP)			Our (Maze)		
			#Pin(2^n)	#WL	Time(s)	#Pin(2^n)	#WL	Time(s)	#Pin(2^n)	#WL	Time(s)	#Pin(2^n)	#WL	Time(s)
amino-acid-1	6X8	20	20(32)	136	0.24	9(16)	190	0.08	7(8)	185	0.26	7(8)	185	0.00
amino-acid-2	8X8	24	24(32)	144	0.32	11(16)	207	0.11	9(16)	180	0.27	9(16)	180	0.00
protein-1	13X13	34	34(64)	348	0.90	24(32)	462	0.26	10(16)	353	0.89	10(16)	348	0.01
protein-2	13X13	51	51(64)	607	0.91	25(32)	662	0.28	31(32)	566	0.94	31(32)	566	0.02
dilution	15X15	54	54(64)	582	1.40	15(16)	1178	0.17	23(32)	820	1.58	23(32)	844	0.14
multiplex	15X15	59	59(64)	851	1.24	36(64)	1444	0.36	10(16)	534	1.80	10(16)	527	0.02
random-1	10X10	20	20(32)	161	0.46	8(8)	278	0.04	9(16)	183	0.56	9(16)	183	0.00
random-2	15X15	30	30(32)	476	1.32	11(16)	614	0.10	16(16)	421	1.70	16(16)	421	0.03
random-3	20X20	60	60(64)	808	3.00	19(32)	2720	0.31	27(32)	1090	3.67	27(32)	1099	0.33
random-4	30X30	90	90(128)	1826	10.00	26(32)	5975	0.40	45(64)	4076	30.00	46(64)	2127	0.85
random-5	50X50	100	100(128)	3814	42.70	37(64)	7965	1.53	28(32)	3830	67.78	28(32)	3994	2.60
random-6	60X60	100	100(128)	3223	54.10	41(64)	8901	2.23	35(64)	2695	120.22	39(64)	3247	1.74
random-7	70X70	150	150(256)	6188	59.86	80(128)	16612	6.65	45(64)	7133	238.83	52(64)	5063	7.53
G.M.	-	-	2.68	0.90	-	1.13	1.69	-	1.00	1.00	-	1.02	0.94	-

TABLE III

COMPARISON WITH PRIOR WORK [20] ON PIN-CONSTRAINED DESIGNS WITH PRESENCE OF OBSTACLES. P_{max} INDICATES THE NUMBER OF CONTROL SIGNALS SUPPLIED BY THE CONTROLLER (G.M. STANDS FOR GEOMETRIC MEAN AND TIME IS MEASURED IN SECONDS)

benchmarks	Size	#E	P_{max}	[20]			Our (ILP-Based)		
				#Pin	#WL	Time(s)	#Pin	#WL	Time(s)
DNA-1	16X24	211	128	128	3003	3.7	128	2335	7.3
DNA-2	13X21	77	32	32	1113	0.7	32	894	2
random-1	6X8	24	16	16	271	0.0	16	206	0.28
random-2	15X15	59	32	32	991	0.4	32	893	1.47
random-3	15X15	62	32	32	1153	0.5	32	872	1.45
random-4	15X15	91	64	64	1417	0.4	64	1244	1.6
random-5	20X30	256	128	128	4742	11.4	128	3969	16.9
random-6	30X40	400	256	256	9099	26.1	256	6959	72.2
G.M.	-	-	-	1.00	1.24	-	1.00	1.00	-

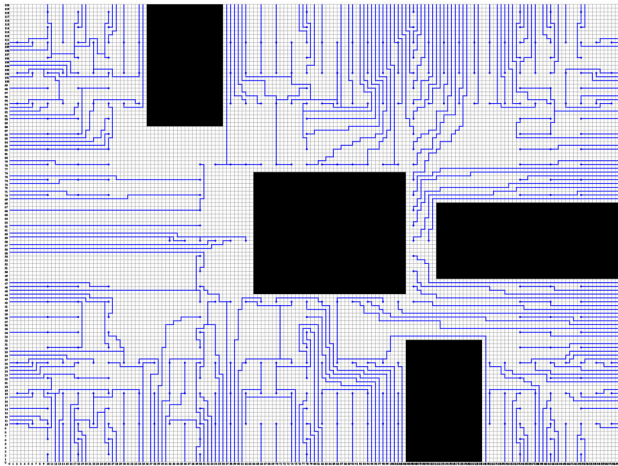


Fig. 13. Routing result of our work on test case [random06] with obstacles.

and reroute. On average, our proposed framework using ILP as escape routing engine requires 13% less control pins and routes the designs with 68% less wirelength.

In some scenarios, our agglomerative clustering may not achieve as much pin count reduction compared to previous works. The primary reason is that our work focuses on routability while previous works focus on greedily reducing pin count. From our experience, routability of the design is correlated to routing capacity of bin and pin density of the

design. For designs with sparse pin distribution or higher routing capacity per unit area, greedy approach is possible to achieve better pin reduction compared to our approach.

In terms of efficiency for each routing iteration, ILP escape routing takes most of the execution time and routing of electrical pins based on maze routing algorithm has negligible run time. However, ILP escape routing does not scale well with large design. Hence, a maze router is implemented to provide an alternative solution with much faster run time at expense of slight degradation in solution quality. Total number of required control pins is increased by an average of 2% when maze router is used instead of integer linear programming escape routing.

C. Evaluation of ACER on Design With Obstacles

To demonstrate flexibility of our algorithm. We have also experimented on designs with presence of obstacles. Table III compares our algorithm with the work done in [20]. In this set of designs, a value P_{max} is given for each design which specifies the maximum control signal that can be supplied by the controller. The merging stage in our algorithm terminates when pin-count is reduced to less than or equal to P_{max} . In comparison with the work done in [20], our proposed algorithm uses 24% less routed wirelength while satisfying the P_{max} constraint on all designs. Fig. 13 is the routing result on testcase *random06* using our proposed algorithm.

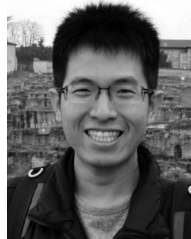
VII. CONCLUSION

In this paper, we solved both electrode addressing and routing problem for pin-constrained broadcast addressing EWOD designs. We have shown that previous works pay very little attention on routability of the design and greedily reducing pin-count adopted in previous works suffers several shortcomings that are ineffective to satisfy the pin constraint. Thus, we present ACER, a framework that uses agglomerative clustering to solve electrode addressing with consideration of routability. Compared to previous works, our proposed algorithm is simple and effective. Possible improvements include considering routing capacity and pin density into agglomerative clustering. By having better knowledge toward total routing capacity is possible to achieve more pin-count reduction. Applying network flow algorithm is also a possible direction to accelerate runtime.

In short, this paper shows that routed wirelength, routability of the design and pin-count reduction are all correlated. Shorter wirelength means less routing blockages, less routing blockages implies better routability and better routability enables more pin-count reduction. Considering routability during electrode addressing enables further reduction in pin-count compared to greedy pin-count reduction approach. Our claim is supported based on simulation results on every possible testcase with and without obstacles that we can obtain.

REFERENCES

- [1] T.-W. Huang, S.-Y. Yeh, and T.-Y. Ho, "A network-flow based pin-count aware routing algorithm for broadcast-addressing EWOD chips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 17, no. 12, pp. 1786–1799, Dec. 2011.
- [2] T.-Y. Ho, J. Zeng, and K. Chakrabarty, "Digital microfluidic biochips: A vision for functional diversity and more than Moore," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Bangalore, India, 2010, pp. 578–585.
- [3] T.-W. Huang, T.-Y. Ho, and K. Chakrabarty, "Reliability-oriented broadcast electrode-addressing for pin-constrained digital microfluidic biochips," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, San Jose, CA, USA, 2011, pp. 448–455.
- [4] M.-G. Pollackand, A.-D. Shenderov, and R.-B. Fair, "Electrowetting-based actuation of droplets for integrated microfluidics," *Lab Chip*, vol. 2, no. 2, pp. 96–101, Feb. 2002.
- [5] F. Su, K. Chakrabarty, and R. B. Fair, "Microfluidics-based biochips: Technology issues, implementation platforms, and design automation challenges," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 2, pp. 211–223, Feb. 2006.
- [6] J. Gong and C.-J. Kim, "Direct-referencing two-dimensional-array digital microfluidics using multilayer printed circuit board," *J. Microelectromech. Syst.*, vol. 17, no. 2, pp. 257–264, Apr. 2008.
- [7] T. Xu and K. Chakrabarty, "Broadcast electrode-addressing for pin-constrained multi-functional digital microfluidic biochips," in *Proc. Design Autom. Conf.*, Anaheim, CA, USA, 2008, pp. 173–178.
- [8] S.-H. Yeh, J.-W. Chang, T.-W. Huan, and T.-Y. Ho, "Voltage-aware chip-level design for reliability-driven pin-constrained EWOD chips," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, San Jose, CA, USA, 2012.
- [9] C. Y. Lee, "An algorithm for path connections and its applications," *IRE Trans. Electron. Comput.*, vol. EC-10, no. 3, pp. 346–365, Sep. 1961.
- [10] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*. New York, NY, USA: Plenum Press, 1972, pp. 85–103.
- [11] J. H. Ward, "Hierarchical grouping to optimize an objective function," *J. Amer. Statist. Assoc.*, vol. 58, no. 301, pp. 236–244, 1963.
- [12] G. J. Szekely and M. L. Rizzo, "Hierarchical clustering via joint between-within distances: Extending ward's minimum variance method," *J. Classificat.*, vol. 22, pp. 151–183, Sept. 2005.
- [13] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, 2nd ed. New York, NY, USA: McGraw-Hill, 2001.
- [14] C. Y. Lee, "An algorithm for path connections and its applications," *IRE Trans. Electron. Comput.*, vol. EC-10, no. 3, pp. 346–365, Sep. 1961.
- [15] K.-R. Dai, W.-H. Liu, and Y.-L. Li, "NCTU-GR: Efficient simulated evolution-based rerouting and congestion-relaxed layer assignment on 3-D global routing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 3, pp. 459–472, Mar. 2012.
- [16] J. A. Roy and I. L. Markov, "High-performance routing at the nanometer scale," in *Proc. Int. Conf. Computer-Aided Design*, San Jose, CA, USA, 2007, pp. 496–502.
- [17] Q. Ma, T. Yan, and M. D. F. Wong, "A negotiated congestion based router for simultaneous escape routing," in *Proc. Int. Symp. Quality Electron. Design*, San Jose, CA, USA, Mar. 2010, pp. 606–610.
- [18] L. Luo, T. Yan, Q. Ma, M. D. Wong, and T. Shibuya, "A new strategy for simultaneous escape based on boundary routing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 2, pp. 205–214, Feb. 2011.
- [19] L. Abel, "On the ordering of connections for automatic wire routing," *IEEE Trans. Comput.*, vol. C-21, no. 11, pp. 1227–1233, Nov. 1972.
- [20] J.-W. Chang, T.-W. Huang, and T.-Y. Ho, "An ILP-based obstacle-avoiding routing algorithm for pin-constrained EWOD chips," in *Proc. Asia South Pacific Design Autom. Conf.*, Sydney, NSW, Australia, 2012, pp. 67–72.
- [21] (2014, Jun. 26). *IBM ILOG CPLEX Optimizer* [Online]. Available: <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer>



Sean Shih-Ying Liu received the B.S. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2007, where he is currently pursuing the Ph.D. degree from the Institute of Electronics.

His current research interest is primarily on placement optimizations for VLSI designs.



Chung-Hung Chang received the B.S. degree in electrical engineering from National Chung Cheng University, Mianxiang, Taiwan, and the M.S. degree from the Institute of Electronics from National Chiao Tung University, Hsinchu, Taiwan.

He is with the Reliability Department, Macronix Company Ltd, Hsinchu. His current research interests include design automation for microfluidic biochips.



Hung-Ming Chen received the B.S. degree in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1993, and the M.S. and Ph.D. degrees in computer science from the University of Texas, Austin, TX, USA, in 1998 and 2003, respectively.

He is currently a Professor with the Department of Electronics Engineering, National Chiao Tung University. His current research interests include physical design automation in digital and analog circuits, beyond-die integration (off-chip electronic design automation), and 3-D integrated circuit design methodology.

Dr. Chen has served as a Technical Program Committee Member on ACM/IEEE ASP-DAC, IEEE SOCC, VLSI-DAT, and ACM ISPD.



Tsung-Yi Ho (M'08–SM'12) received the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2005.

Since 2007, he has been with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, where he is currently an Associate Professor. His current research interests include design automation for microfluidic biochips and nanometer integrated circuits. He has published several papers in top journals and conferences such as IEEE TCAD, ACM TODAES, ACM/IEEE DAC, IEEE/ACM ICCAD, and ACM ISPD. He presented eight tutorials and contributed four special sessions in ACM/IEEE conferences, all in design automations on biochips.

Prof. Ho was the recipient of many research awards, such as the Dr. Wu Ta-You Memorial Award of National Science Council in Taiwan, the Research Award for Junior Research Investigators from Academia Sinica, the Distinguished Young Scholar Award of Taiwan IC Design Society, the Outstanding Young Electrical Engineer Award of Chinese Institute of Electrical Engineering, the K. T. Li Research Award of Delta Electronics, the ACM Taipei Chapter Young Researcher Award, the IEEE Tainan Chapter Gold Member Award, the Invitational Fellowship of the Japan Society for the Promotion of Science, Japan, and the Humboldt Research Fellowship from the Alexander von Humboldt Foundation, Germany. Currently, he serves as a Distinguished Visitor of IEEE Computer Society, the Chair of IEEE Computer Society Tainan Chapter, and an Associate Editor of the *ACM Journal on Emerging Technologies in Computing Systems* and the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS.