Optimal Wiring Topology for Electromigration Avoidance Considering Multiple Layers and Obstacles

Iris H.-R. Jiang
Department of Electronics Engineering & Institute of Electronics
National Chiao Tung University
Hsinchu, Taiwan
huiru.jiang@gmail.com

Hua-Yu Chang
Freelance
Taipei, Taiwan
huayu.chang@gmail.com

Chih-Long Chang
Department of Electronics Engineering & Institute of Electronics
National Chiao Tung University
Hsinchu, Taiwan
paralost.ee96@g2.nctu.edu.tw

ABSTRACT
Due to excessive current densities, electromigration may trigger a permanent open- or short-circuit failure in signal wires or power networks in analog or mixed-signal circuits. As the feature size keeps shrinking, this effect becomes a key reliability concern. Hence, in this paper, we focus on wiring topology generation for avoiding electromigration at the routing stage. Prior works tended towards heuristics; on the contrary, we first claim this problem belongs to class P instead of class NP-hard. Our breakthrough is, via the proof of the greedy-choice property, we successfully model this problem on a multi-source multi-sink flow network and then solve it by a strongly polynomial time algorithm. Experimental results prove the effectiveness and efficiency of our algorithm.

Categories and Subject Descriptors
B.7.2 [**Integrated Circuits**]: Design Aids – Placement and Routing, B.8.1 [**Integrated Circuits**]: Performance and Reliability – Reliability, Testing, and Fault-Tolerance

General Terms
Algorithms, Reliability.

Keywords
electromigration, routing, network flow, linear programming.

1. INTRODUCTION
As technology advancing, the shrinking feature size results in the increasing current density [1]. When the current density of a wire exceeds the process limitation, a mass of atoms inside of the metal are forced to migrate along the direction of the current, and the gradual transport eventually causes a permanent failure (e.g., open- or short-circuit defect). This phenomenon is referred to as electromigration (EM). Since this failure is triggered after a circuit has been operating for months, or even years, EM reliability is measured in terms of the mean time to failure (MTF), depending on the current density and the working temperature of a wire [2]. Hence, in modern analog and mixed-signal designs, EM has become a prevalent reliability issue for signal or power lines [3]. Recently, extensive research endeavors have been dedicated to EM reliability, e.g., [4-11].

EM reliability relies on the following two techniques:

1. An accurate EM simulator/analyzer: Because EM manifests itself only after a circuit has been used for a period of time, the defect chips cannot be filtered out during product testing. A superior EM simulator/analyzer is desirable to characterize the realistic current values for each terminal/wire and to identify the wires that are potentially threatened by EM [4-6].
2. A good wiring topology for EM: To diminish the EM risk, a thin wire should be widened according to its current density. Conventional EM fixing is applied at post-layout, which may consume tremendous routing resources and induce a large amount of layout change. On the contrary, if a good wiring topology considering EM is applied to a router, we may immune EM with much fewer routing resources [7-10].

Typically, the routing resource is measured by the total wire area. Under EM consideration, the wire width should vary according to its current density. Consider an instance given by [9, 10]—a signal net with three current sources and four current sinks—as shown in Figure 1(a). The current value of each current source/sink is indicated by a positive/negative value. Here, the allowable current in a unit width wire is normalized to 1. If the minimum wirelength routing is applied, followed by EM fixing by widening wires at post-layout, Figure 1(b) depicts the result of wire area 182. (The number/arrow beside each wire segment indicates its current value/direction.) The post-layout modification may consume a high routing resource and even produce DRC errors. Although we can refine it, a significant amount of layout change is inevitable. Considering EM during routing instead, prior works [9] and [10] propose heuristics and have the lower wire area 154 (Figure 1(d)) and 144 (Figure 1(f)), respectively. In fact, we find there is an optimal solution of wire area 142 (Figure 1(h)).

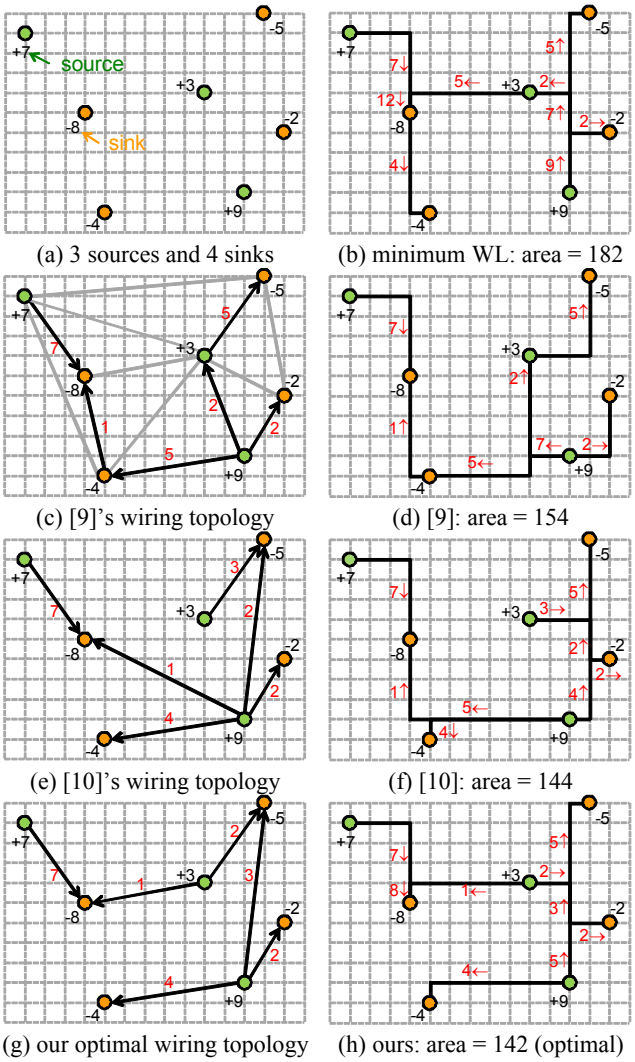
Hence, based on this case, it would be beneficial to consider electromigration at routing. This task is accomplished by wiring topology generation first and then detailed routing [7-10]. The wiring topology determines the connection among all terminals, gives the current-correct wire widths, and minimizes the wire area. The detailed router converts each two-terminal connection to a detailed routing path and composes the whole rectilinear Steiner

ISPD'10, March 14–17, 2010, San Francisco, California, USA.

tree. If the routing resource of the detailed route can completely be abstracted into the wiring topology, then an optimal wiring topology implies an optimal routing tree.

In this paper, therefore, we focus on the wiring topology for electromigration avoidance problem: Given the maximum tolerable current over a unit-area wire for each layer, the feasible minimum wire width, a set of current sources and a set of current sinks, sources/sinks are associated with characterized current values (e.g., DC, RMS, peak, average currents), our goal is to find a wiring topology that can offer a sufficient current for each wire segment and consumes the least wire area.

Prior works of wiring topology generation tended towards heuristics [7-10]. Among them, [9] and [10] deliver the best results so far. In [9], first of all, the Delaunay triangulation (DT) is constructed so that all current sources and sinks are connected [12]. Secondly, a source or sink at the DT boundary is selected as the starting point, and it is connected to its nearest neighbor in DT. Considering the two possible current directions of the connection, it is then greedily connected to the nearest neighbor with the best result and the wire width is determined accordingly. The process is repeated and, finally, the resulting tree is formed. Figure 1(c) illustrated the wiring topology and DT done by [9]. However, the tree edge can be provided only from DT edges; thus, if the solution must include a non-DT edge, [9] fails. On the other hand, [10] first ranks pair-wise area benefits between sources and sinks. Then, [10] iteratively and greedily assigns wire widths starting from the source-sink pair with the largest area gain. (see Figure 1(e)(f)) However, [10] does not consider the competition among all sources. In addition, the area gain computation is time-consuming, so [10] only reported the results of small cases in their paper. Moreover, checking the corresponding detailed routes of [9] and [10] shown in Figure 1, the detailed routes can be further improved. However, this possibility of improvement cannot be interpreted to their heuristics.



(a) 3 sources and 4 sinks (b) minimum WL: area = 182

(c) [9]'s wiring topology (d) [9]: area = 154

(e) [10]'s wiring topology (f) [10]: area = 144

(g) our optimal wiring topology (h) ours: area = 142 (optimal)

Figure 1. EM routing. (a) An instance with 3 current sources and 4 current sinks used by [9, 10]. (b) A routing of minimum wirelength: area 182, wirelength 33. (c)(d) Generated by prior work [9]: area 154. (e)(f) Generated by prior work [10]: area 144. (g)(h) Ours: area 142 (optimal).

On the contrary, after exploring the complexity of this problem, we first claim this problem belongs to class P instead of class NP-hard [13]. Our breakthrough is, via the proof of the greedy-choice property [14], we successfully model this problem on a multi-source multi-sink flow network, and then it can be solved in a strongly polynomial running time. Experimental results prove our theoretical contribution. Compared with the state-of-the-art works [9, 10], our algorithm outperforms them by large margins in terms of effectiveness and efficiency, especially for large-scale cases.

2. PROBLEM FORMULATION AND PROPERTIES

This section gives the problem formulation, derives the properties that help us to abstract the routing resource of a detailed route, and proves the greedy choice property.

2.1 Problem Formulation

In this paper, we consider the wiring topology for electromigration avoidance problem as follows.

> ***Problem: Wiring Topology for Electromigration Avoidance (TEA):*** Given a set $S = \{s_1, s_2, \ldots, s_m\}$ of m current sources, a set $T = \{t_1, t_2, \ldots, t_n\}$ of n current sinks, each current source i (sink j) is associated with its flow $f_{s_i}(f_{t_j})$, the maximum tolerable current density $J_{\max}$ and the minimum feasible wire width $w_{\min}$ for each routing layer/via, construct a wiring topology to connect all current sources and sinks in $S+T$, such that the current for each wire segment is sufficient, the total wire area of its corresponding detailed routing tree is minimized, and Kirchhoff's current conservation law is satisfied everywhere.

$J_{\max}$ and $w_{\min}$ for each layer (via) can be provided by technology file. On the other hand, without loss of generality, the thickness h of a wire is a layer-specific constant. To minimize the routing resource (wire area), we can fully utilize $J_{\max}$ for each layer (via):

$$J_{\max} = \frac{I}{A} = \frac{f}{wh},$$

$$\Rightarrow f = J_{\max} wh = (J_{\max} h)w,$$

$$\Rightarrow f \propto w. \qquad (1)$$

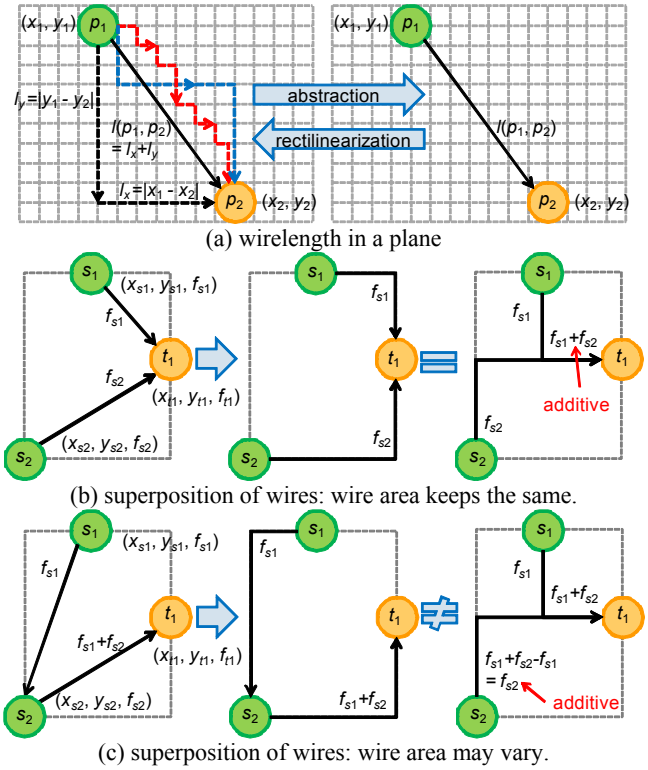
Since J_{max} and h are constant for a specific layer (via), the flow (current value) f is proportional to the wire width w. Hence, for a unit length wire, the wire width offering one unit current consumes routing resource $1/J_{max}$. Therefore, area optimality can be achieved by minimizing not only the wirelength of a wire segment but also the flow within it.

In addition, the feasible wire width should be greater than w_{min}, so the resulting wire width should be $\max\{f_{opt}, w_{min}\}$, where f_{opt} is the optimal flow assigned. If f_{opt} is smaller than w_{min} everywhere, EM is not a concern. Usually, w_{min} is smaller than f_{opt} for signal lines of large currents in analog circuits or for power supply lines. Assume w_{min} is narrower than the required wire width for a unit current for each layer. Furthermore, $f_{s_i}(f_{t_j})$ is represented by a positive (negative) value for a current source (sink). The characterized current values can be DC, RMS, peak, average currents under different EM conditions and for various signal types.

According to Kirchhoff's current conservation law, a legal TEA problem must satisfy the following criterion, i.e., the total current values from sources equal those to sinks.

$$\sum_{i=1}^{m} f_{s_i} + \sum_{j=1}^{n} f_{t_j} = 0. \quad (2)$$

This criterion can be verified at the very beginning. Since our goal is to dispatch all currents generated from sources and to maintain a balance, in some cases, the resulting topology forms a forest; we may add some dummy wires in between trees if necessary. In addition, our problem formulation is equivalent to that in [9, 10].



(a) wirelength in a plane

(b) superposition of wires: wire area keeps the same.

(c) superposition of wires: wire area may vary.

Figure 2. (a) Wirelength in the rectilinear system. (b)(c) Wire segments can be superpositioned.

2.2 Wirelength and Wire Area

First of all, consider an obstacle-free plane. For rectilinear routing, only vertical and horizontal lines are allowed. Hence, as depicted in Figure 2(a), the wirelength (Manhattan distance) $l(p_1, p_2)$ between two arbitrary points $p_1(x_1, y_1, f_1)$ and $p_2(x_2, y_2, f_2)$ can be computed as follows.

$$l(p_1,p_2) = l_x + l_y = |x_1 - x_2| + |y_1 - y_2|. \quad (3)$$

It can be seen that each possible practical route between arbitrary two sources/sinks can be abstracted by the slant wire segment of length equal to their Manhattan distance. Moreover, due to coordinate independence, l_x and l_y are considered separately.

However, the routing cost varies for layers (vias), and obstacles may exist. In both case, there is a *one-to-one mapping* between the slant wire segment and the shortest routing path, i.e., the path with the minimum *accumulated area cost* for one unit current within this connection. Based on Section 2.1, the area cost of a wire segment of length l on a specific layer (via) is l/J_{max}. Taking the summation of the area costs along a path, we obtain its accumulated area cost. Hence, this abstraction still works. In addition, this abstraction can be also applied for non-uniform temperature distribution.

Typically, the consumed routing resource can be computed as the total wiring area. For practical routes, if two wire segments overlap at some layer, as shown in Figure 2(b)(c), the overlapped parts can be *superpositioned*, i.e., *additive* considering current directions. Based on superposition, a current can be split into pieces, planned individually, and then combined together.

Our goal is to find the property of the optimal wiring topology. Moreover, it is desirable that the wire area estimated by the wiring topology is equal to that at detailed routing.

2.3 Greedy-Choice Property

As mentioned in Section 1, to minimize the routing resource, i.e., wire area, not only the wirelength for each wire segment but also the magnitude of its flow should be considered. Here, we prove the greedy-choice property of the TEA problem. This property simplifies the direction of a flow thus allowing us to develop the optimal wiring topology in Section 4.

First of all, we figure out the generic form of the greedy-choice property for wire connections, as shown in Figure 3(a), where two sources s_1, s_2 individually connect one sink t_1. Symmetrically, this generic form can extend to the case with two sinks plus one source, as shown in Figure 3(b). Any more complicated case is composed of many copies of the generic form. By the following theorem and corollary, we can largely simplify the routing problem. We start from an obstacle-free plane, and then extend to a general multi-layer space with obstacles.

Theorem: Greedy-Choice Property: There exist two sources s_1, s_2 and one sink t_1. Let the partial flows of s_1 and s_2 be sunken by t_1 in the optimal wiring topology:

$$f_{s_1} + f_{s_2} + f_{t_1} = 0. \quad (4).$$

The greedy-choice property of the partial wiring topology for these flows is in the form that the wires directly connecting from s_1 to t_1 as well as from s_2 to t_1. Considering the current directions, the flow within each wire segment can be assigned accordingly.

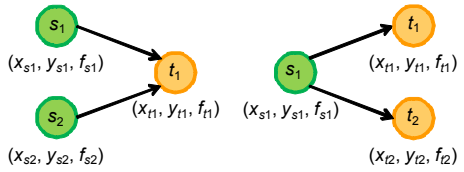


Figure 3. The generic form for the greedy-choice property. (a) 2 sources s_1, s_2 & 1 sink t_1. (b) 2 sinks t_1, t_2 & 1 source s_1.

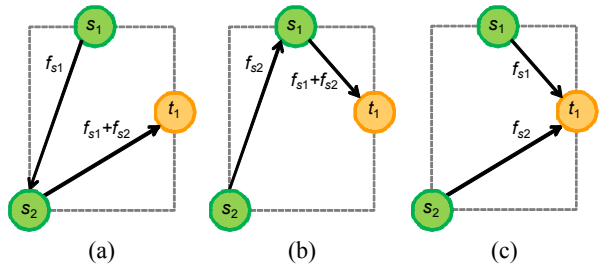


Figure 4. The greedy-choice property. (a) $\{s_1 \rightarrow s_2 \rightarrow t_1\}$. (b) $\{s_2 \rightarrow s_1 \rightarrow t_1\}$. (c) $\{s_1 \rightarrow t_1\}$ plus $\{s_2 \rightarrow t_1\}$.

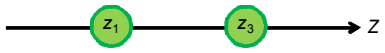


Figure 5. Three points z_1, z_2 and z_3 on an axis, where z_1 and z_3 are fixed, while z_2 is movable.

Proof: Without loss of generality, wire connections among s_1, s_2, and t_1 fall into three options as shown in Figure 4:

1. s_1 first connects s_2, and then s_2 connects t_1 (see Figure 4(a)).
2. s_2 first connects s_1, and then s_1 connects t_1 (see Figure 4(b)).
3. s_1 and s_2 connect t_1 individually (see Figure 4(c)).

Assume the optimal wiring topology is of wire area $A_{\min}$. In the following, we will claim that option 3, connecting the sources to the sink individually (Figure 4(c)), is the greedy choice. Please note that the following analysis is general and can be applied to any geometry relationship. (In addition, option 1 and option 2 are identical, so the details of option 2 are omitted here.)

The wire area for option 1 is as follows.

$$A(s_1 \rightarrow s_2 \rightarrow t_1) = f_{s_1} \cdot l(s_1, s_2) + (f_{s1} + f_{s_2}) \cdot l(s_2, t_1)$$
$$= f_{s_1} \cdot (|x_{s_1} - x_{s_2}| + |y_{s_1} - y_{s_2}|) + (f_{s1} + f_{s_2}) \cdot (|x_{s_2} - x_{t_1}| + |y_{s_2} - y_{t_1}|)$$
$$= f_{s_1} \cdot (|x_{s_1} - x_{s_2}| + |x_{s_2} - x_{t_1}|) + f_{s_2} \cdot |x_{s_2} - x_{t_1}|$$
$$+ f_{s_1} \cdot (|y_{s_1} - y_{s_2}| + |y_{s_2} - y_{t_1}|) + f_{s_2} \cdot |y_{s_2} - y_{t_1}|. \quad (5)$$

Also, the wire area for option 3 is as follows.

$$A(s_1 \rightarrow t_1) + A(s_2 \rightarrow t_1) = f_{s_1} \cdot l(s_1, t_1) + f_{s_2} \cdot l(s_2, t_1)$$
$$= f_{s_1} \cdot (|x_{s_1} - x_{t_1}| + |y_{s_1} - y_{t_1}|) + f_{s_2} \cdot (|x_{s_2} - x_{t_1}| + |y_{s_2} - y_{t_1}|)$$
$$= f_{s_1} \cdot |x_{s_1} - x_{t_1}| + f_{s_2} \cdot |x_{s_2} - x_{t_1}| + f_{s_1} \cdot |y_{s_1} - y_{t_1}| + f_{s_2} \cdot |y_{s_2} - y_{t_1}| \quad (6)$$

If option 3 is optimal, $A_{\min} = A(s_1 \rightarrow t_1) + A(s_2 \rightarrow t_1)$, we are done. Otherwise, option 1 (or option 2) is optimal:

$$A_{\min} = A(s_1 \rightarrow s_2 \rightarrow t_1) \le A(s_1 \rightarrow t_1) + A(s_2 \rightarrow t_1). \quad (7)$$

Based on superposition and coordinate independence, the following analysis on the wire area is individually applied to each axis. Consider three arbitrary points, z_1, z_2 and z_3, on an axis. Assume z_1 and z_3 are fixed, while z_2 is movable, as shown in Figure 5. Considering z_2 is located beyond or in between z_1 and z_3, we have

$$|z_1 - z_2| + |z_2 - z_3| \ge |z_1 - z_3|. \quad (8)$$

Substituting z_1, z_2, z_3 with the coordinates in Equation (5),

$$|x_{s_1} - x_{s_2}| + |x_{s_2} - x_{t_1}| \ge |x_{s_1} - x_{t_1}|, \; |y_{s_1} - y_{s_2}| + |y_{s_2} - y_{t_1}| \ge |y_{s_1} - y_{t_1}|.$$

After simplifying it,

$$A(s_1 \rightarrow s_2 \rightarrow t_1) \ge A(s_1 \rightarrow t_1) + A(s_2 \rightarrow t_1). \quad (9)$$

Hence, combining Inequalities (7) and (9),

$$A_{\min} = A(s_1 \rightarrow s_2 \rightarrow t_1) \le A(s_1 \rightarrow t_1) + A(s_2 \rightarrow t_1), \text{ and}$$
$$A(s_1 \rightarrow s_2 \rightarrow t_1) \ge A(s_1 \rightarrow t_1) + A(s_2 \rightarrow t_1).$$
$$\Rightarrow A_{\min} = A(s_1 \rightarrow s_2 \rightarrow t_1) = A(s_1 \rightarrow t_1) + A(s_2 \rightarrow t_1). \quad (10)$$

If option 1 (or option 2) is optimal, option 3 is also optimal. Thus, option 1 (or option 2) can be replaced by option 3 without loss of optimality. The greedy-choice property thus follows. □

Corollary: The greedy-choice property holds at a general multi-layer space with obstacles.

The detailed proof of the corollary is omitted. By the above theorem and corollary, we can consider wire connections *only* from sources to sinks. The TEA problem can then be viewed as a pairing problem. Moreover, the wire area keeps the same after the slant edges of the greedy-choice property are rectilinearized (see Figure 2(b)), so we can precisely calculate the wire area directly on the wiring topology.

3. GRAPH MODELING

3.1 Overview of the Electromigration Avoidance Routing Tree

Figure 6 outlines how to construct a routing tree considering electromigraiton avoidance. First of all, the wirelength and the flow bound for the connection between any pair of source and sink are computed and recorded by a multiple-source multiple-sink flow network. Secondly, sources and sinks are paired, the flow on each connection is determined, and a wiring topology is

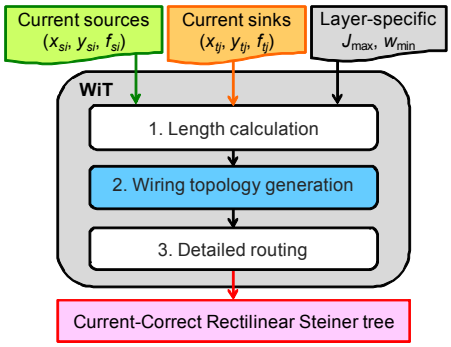


Figure 6. The EM avoidance routing tree construction.

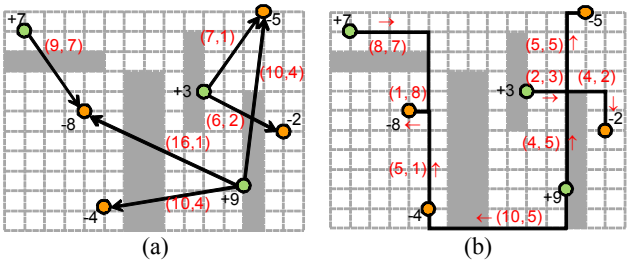


Figure 7. (a) The instance in Figure 1(a) with 4 obstacles (grey blocks) and its obstacle-avoidance wiring topology (optimal area: 178). (b) The corresponding detailed routing tree (area: 178). The pair (length, flow) is attached beside each wire.

generated. Finally, each connection is rectilinearized by the stored shortest path. If we can find a pairing of optimal wire area, we can then preserve the optimality, because rectilinearization does not affect the solution quality.

3.2 Length Calculation

Considering multiple routing layers and avoiding obstacles, length calculation in Figure 6 should compute the shortest path length for each possible connection by Dijkstra's algorithm [14]. This shortest path (with the minimum *accumulated area cost* for a unit-flow) is stored, and if this pair is included in the wiring topology, it will be retrieved at rectilinearization. Figure 7 depicts the resulting wiring topology and routing tree of the instance in Figure 1(a) with 4 extra obstacles added. In addition, after rectilinearization, the wire area still keeps optimal.

3.3 Flow Network

As depicted in Figure 8, a flow network for an instance with m current sources (s_1, s_2, …, s_m) and n current sinks (t_1, t_2, …, t_n) is a directed graph. Each current source/sink is represented by a vertex. For each pair of source and sink, we have a directed edge (s_i, t_j) from s_i to t_j associated with a triple ($l_{i,j}$,$f_{i,j}$, $c_{i,j}$):

1. $l_{i,j}$: wirelength between s_i and t_j.
2. $f_{i,j}$: flow from s_i to t_j.
3. $c_{i,j}$: capacity (upper bound) of flow $f_{i,j}$, $c_{i,j} = \min(f_{s_i}, f_{t_j})$.

The capacity $c_{i,j}$ and wirelength $l_{i,j}$ can be obtained during length calculation. In addition, two dummy vertices, the super source s_s and the super sink t_t, are added. One edge from s_s to each distinct source s_i is associated with $(0, f_{s_i}, f_{s_i})$, while one edge from each distinct sink t_j to t_t is associated with $(0, f_{t_j}, f_{t_j})$. In addition, one edge (s_s, t_t) is also added associated with $(0, f_{s_s}, f_{s_s})$, where f_{s_s} is the total flow from sources. The objective is to minimize total wire area. Since only the flow from source to sink is considered based on the greedy-choice property, the flow of each pair-wise connection is bounded by the smaller magnitude of its related source and sink. It can be seen that the greedy-choice property largely simplifies the flow network. There are no edges either between sources or between sinks. Thus, the central part of the flow network, i.e., the subgraph induced by sources and sinks, is an $m \times n$ bipartite graph.

4. NETWORK-FLOW-BASED APPROACH

We can reduce the TEA problem to the multiple-source multiple-sink minimum-cost maximum-flow problem based on the greedy-choice property. We further develop an optimal algorithm—WiT based on the concept of negative cycle detection. Furthermore, the WiT algorithm has a strongly polynomial running time [15], i.e., the running time is not related to the values of flow and length.

4.1 Initial Solution - Greedy Method

First of all, we shall devise an algorithm that can quickly assign an initial flow into each possible wire connection of good quality. For efficiency, we select the greedy method; for effectiveness, we choose the minimum wirelength as the greedy rule. The wirelength between any pair of source and sink can be viewed as the area cost for assigning one-unit flow into this connection. Conceptually, we may greedily start from a pair with minimum wirelength (instead of with minimum area). The pair-wise wirelength can be stored in a minheap, and then the greedy method can be done in O((mn)lg(mn)) time [14].

4.2 Flow and Residual Graphs

To construct an optimal wiring topology, we start from the flow network created in Section 3. The initial flow within each pair-wise wire connection is assigned by our greedy method described in Section 4.1. As shown in Figure 9(a), if the flow within the wire between source s_i and sink t_j is $f_{i,j}$, subject to the capacity $c_{i,j}$, then we may push in no more than $c_{i,j}$-$f_{i,j}$ flow, or we can return at most $f_{i,j}$ back. One-unit flow injection contributes $+l_{i,j}$ wire area; on the contrary, one-unit flow removal has $-l_{i,j}$ impact on wire area. Based on this idea, the residual graph can be constructed as Figure 9(b). In this case, a forward edge from s_i to t_j is constructed of residual capacity $c_{i,j}$-$f_{i,j}$ and of wirelength $l_{i,j}$. Simultaneously, a backward edge from t_j to s_i is also introduced of residual capacity

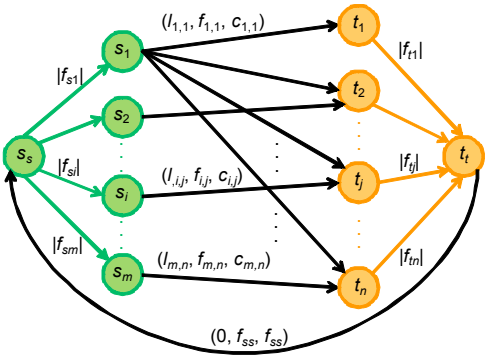


Figure 8. The flow network.

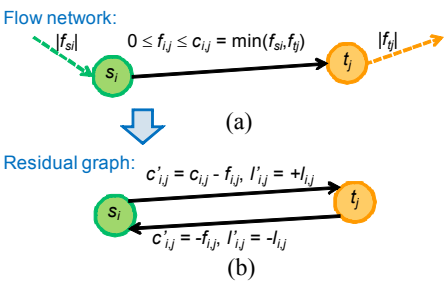


Figure 9. (a) The flow network. (b) The residual graph.

$-f_{i,j}$ and of wirelength $-l_{i,j}$. The positive/negative sign (+/-) of capacity and wirelength implies the possible direction and area impact of its residual flow.

We summarize the residual graph construction as follows. For each edge (s_i, t_j) between source s_i and sink t_j with flow $f_{i,j}$ and of wirelength $l_{i,j}$ in the flow network, two edges are constructed in the corresponding residual graph as follows, where the residual flow/capacity/wirelength is represented by $(l'_{i,j}, f'_{i,j}, c'_{i,j})$.

1. A forward edge (s_i, t_j) has a residual capacity $c'_{i,j} = c_{i,j} - f_{i,j}$, $0 \le f'_{i,j} \le c_{i,j} - f_{i,j}$, and has a wirelength $l'_{i,j} = +l_{i,j}$.
2. A backward edge (t_j, s_i) has a residual capacity $c'_{i,j} = -f_{i,j}$, $-f_{i,j} \le f'_{i,j} \le 0$, and has a wirelength $l'_{i,j} = -l_{i,j}$.

Moreover, there are two special cases:

1. If the original flow in the flow network is 0, only the forward edge is added.
2. If the original flow reaches its capacity, only the backward edge is added.

After applying this process to each edge in the flow network, the residual graph is constructed.

4.3 Wire Area Optimization

Based on the residual graph introduced above, we can analyze the possibility to optimize a solution. Figure 10(a) illustrates an initial flow network, and the corresponding residual graph can be derived. Figure 10(b) highlights the negative cycle inside the residual graph, $\{s_1, t_1, s_2, t_2, s_1\}$ of length -2 (= 7 + (-12) + 10 + (-7)). If one-unit flow is injected along the direction of this cycle, then this negative cycle can be removed, and the optimal solution is found.

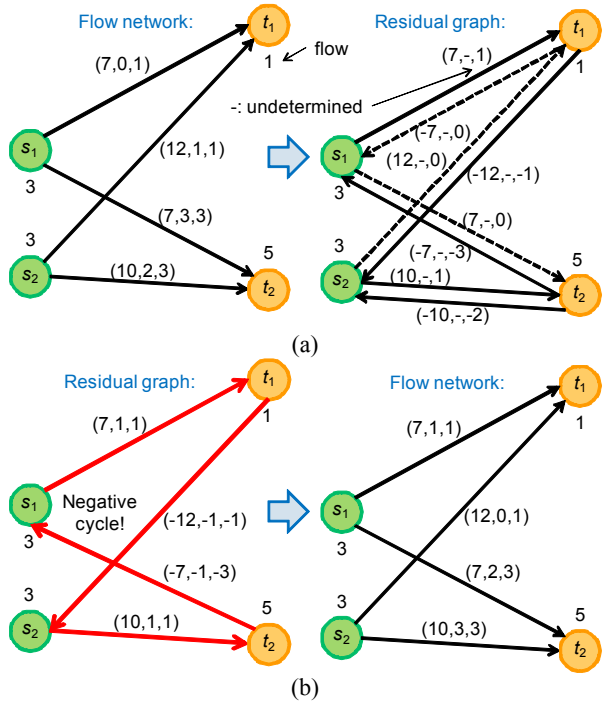


Figure 10. Negative cycle detection. (a)The flow network for a given flow assignment and its residual flow network (dashed lines means non-edges). (b) One negative cycle is removed, and the optimal assignment is obtained.

Therefore, if there exists a negative cycle inside the residual graph, we can reduce the wire area by modifying the flow on each edge within this cycle. A negative cycle $C_N = \{s_i, t_j, \ldots, s_i\}$ is a cycle that starts & ends at the same source and has a negative sum of wirelength L_C on its constituent edges. Moreover, the minimum magnitude of flow among its constituent edges is chosen as the feedback flow f_f. The feedback flow f_f is injected into the negative cycle in the residual graph, then the forward (backward) edges in C_N obtain (return) f_f flow; thus, the negative cycle is then removed. Doing so can deliver the area reduction A_C:

$$A_C = L_C \cdot f_f = \sum_{(s_i,t_j)\in C_N \cup (t_j,s_i)\in C_N} l_{i,j} \cdot \min\{|c_{i,j}|\} < 0. \quad (11)$$

Hence, wire area optimization can be done by iteratively injecting a feedback flow to a negative cycle until there is no negative cycle in the residual graph. Each negative cycle can be detected and removed in O(mn) time. The following theorem concludes this discussion.

Theorem: Negative-Cycle Removal: If the flow assignment of a flow network is optimal, there exists no negative cycles in its residual graph.

4.4 Optimal Network-Flow-Based Algorithm

Figure 11 lists the WiT algorithm. If the input instance is legal, i.e., satisfying Equation (2), the initial flow is assigned. The initial assignment can be done by the greedy method in Section 4.1 or by prior works. The solution is then iteratively refined by negative cycle detection and removal. The optimal algorithm is done in O((mn)lg(mn)+rmn)-time with r iterations for refinement. It can be shown that the number of iterations is regardless of the flows, thus it is a strongly polynomial time algorithm. (The proof is similar to [15].)

5. EXPERIMENTAL RESULTS

We implemented our algorithm in C++ language with LEDA package [16] and executed the program on a PC with an Intel® Core™2 CPU T9400 of 2.53 GHz frequency and 4 GB memory under Windows Vista™ Business 64 bit Service Pack 1 OS. We also implemented prior works on the same platform.

Since [9, 10] do not handle multiple layers and obstacles, for fair comparison, experiments are conducted on planar cases without obstacles. Totally fourteen testcases are adopted. The number of terminals k (including m current sources plus n sinks) ranges from 7 to 850. The first one is directly extracted from [10] (as shown in Figure 1.). The following three are tailored as special cases. The

```
WiT(S, T)
1. construct the flow network and calculate paths and lengths
2. if the given flow at sources/sinks is legal then
3.     find an initial flow assignment
4. while there exists a negative cycle in the residual graph do
5.     remove the negative cycle
6.     update the flow network
7.     update the residual graph
8. rectilinearize the wiring topology by stored paths
```

Figure 11. Optimal flow assignment: the WiT algorithm.

last ten are from [17]; five are from industry, five are random. However, [17] does not have current information; we randomly inject a positive or negative current to each terminal, the current value is normally distributed within the range [-*k*, +*k*]. In practical cases, these values can be extracted from EM simulators. Although the numbers are random in our experiments, the optimality of our method is always guaranteed.

Table 1 compares the WiT algorithm with state-of-the-art works. The WiT algorithm always refines *any* non-optimal solution to the optimal solution. The non-optimal solutions can be (but not limited to) [10] and our initial solution. In experiments, the number of refinement iterations depends on the initial solution quality; however, the upper bound of iterations does not. It can be seen that the number of refinement iterations for our initial solution is much fewer than that for [10], even 0 for some cases. [9] possibly generates infeasible solutions especially for large cases, so we did not try to fix it. Figure 12 shows the results conducted by [9] for IND3. Because the connection between source S3 and sink T3 is missing in the DT, it fails to complete the wiring topology. Figure 1 lists the results for INP1 of all methods listed in Table 1. Interestingly, our fast initial solution is optimal for INP1. Figure 13 illustrates the results of RT05. It can be seen that [9] only connects a relatively small part for tree construction. [10] can complete the wiring topology of large wire area using over two hour runtime, while we can deliver the optimal solution in 90 second runtime.

6. CONCLUSION

In this paper, we focus on wiring topology generation for avoiding electromigration at the routing stage for analog and mixed-signal designs. The major contribution is that we *first* claim this problem belongs to class P instead of class NP-hard. Based on the proof of the greedy-choice property, we successfully model this problem on a multi-source multi-sink flow network and solve it in a strongly polynomial time. Experimental results prove the efficiency and effectiveness of our algorithm.

7. ACKNOWLEDGMENTS

Special thanks go to Prof. Jens Lienig at Dresden University of Technology in Germany and Manager Wen-Hao Chen at TSMC, Ltd. in Taiwan for their valuable suggestions and comments.

8. REFERENCES

[1] ITRS. http://www.itrs.net.

[2] J. R. Black. Electromigration - a brief survey and some recent results. *IEEE Trans Electron Dev.*, vol. 16, no. 4, 1969, pages 338-347.

[3] N. E. Weste and D. M. Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*, 3rd ed., Addison Wesley, 2005.

[4] Cadence Virtuoso. http://www.cadence.com.

[5] Synopsys PrimeRail. http://www.synopsys.com.

[6] R. H. Tu, E. Rosenbaum, W. Y. Chan, C. C. Li, E. Minami, K. Quader, P. K. Ko, C. Hu. Berkeley reliability tools - BERT. *IEEE Trans. Computer-Aided Design*, vol. 12, 1993, pages 1524-1534.

[7] T. Adler and E. Barke. Single step current driven routing of multiterminal signal nets for analog applications. In *Proc. Design, Automation and Test in Europe (DATE-00)*, pages 446-450, 2000.

[8] T. Adler, H. Brocke, L. Hedrich, and E. Barke. A current driven routing and verification methodology for analog applications. In *Proc. Design Automation Conf. (DAC-00)*, pages 385-389, 2000.

[9] J. Lienig and G. Jerke. Current-driven wire planning for electromigration avoidance in analog circuits. In *Proc. Asia and South pacific Design Automation Conf. (ASP-DAC-03)*, pages 783-788, 2003.

[10] J.-T. Yan, and Z.-W. Chen. Electromigration-aware rectilinear Steiner tree construction for analog circuits. In *Proc. Asia Pacific Conf. on Circuits and Systems (APCCAS-08)*, pages 1692-1695, 2008.

[11] J. Lienig. Introduction to electromigration-aware physical design. In *Proc. Int. Symp. on Physical Design (ISPD-06)*, pages 39-46, 2006.

[12] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*, 3rd ed., Springer-Verlag, 2008.

[13] M. R. Garey and D. S. Johnson. The rectilinear Steiner tree problem is NP-complete. *SIAM J. Appl. Math*., vol. 32, no. 4, 1977, pages 826-834.

[14] T. H. Cormen, C. E. Leiserson, R. E. Rivest, and C. Stein. *Introduction to Algorithms*, 2nd ed., MIT Press, 2001.

[15] A. V. Goldberg and R. E. Tarjan. Finding minimum-cost circulations by canceling negative cycles. *J. Assoc. Comput. Mach.*, vol. 36, no. 4, 1989, pages 873-886.

[16] The LEDA package. http://www.algorithmic-solutions.com.

[17] C.-W. Lin, S.-Y. Chen, C.-F. Li, Y.-W. Chang, and C.-L. Yang. Obstacle-avoiding rectilinear Steiner tree construction based on spanning graphs. *IEEE Trans. Computer-Aided Design*, vol. 27, no. 4, 2008, pages 643-653.

Table 1. Comparison on area, runtime, refinement iterations between WiT and state-of-the-art works.

Testcase		INP1	INP2	INP3	INP4	IND1	IND2	IND3	IND4	IND5	RT01	RT02	RT03	RT04	RT05
#Terminals		7	16	10	16	10	10	10	25	33	75	180	303	475	850
[9]	Area	154	122	210	32	6,661	79,400	FAIL	23,112	31,466	FAIL	FAIL	FAIL	FAIL	FAIL
	Runtime (s)	0.001	0.002	0.001	0.002	0.038	0.016	-	0.016	0.040	-	-	-	-	-
[10]	Area	144	98	128	40	5,319	79,000	6,181	16,000	20,814	189,437	10,638,884	3,360,716	6,475,941	59,207,240
	Runtime (s)	0.001	0.002	0.001	0.002	0.001	0.001	0.001	0.003	0.005	0.120	4.958	45.553	297.806	8,573.453
	Refined area	**142**	**90**	**116**	**32**	**5,301**	**74,200**	**5,513**	**13,728**	**17,044**	**144,071**	**7,151,286**	**2,325,806**	**4,205,757**	**37,318,054**
	Gain	2	8	12	8	18	4,800	668	2,272	3,770	45,366	3,487,598	1,034,910	2,270,184	21,889,186
	#Iterations	**1**	**2**	**1**	**1**	**1**	**3**	**15**	**20**	**29**	**147**	**622**	**1,938**	**4,288**	**9,238**
	Refine runtime (s)	0.000	0.000	0.001	0.000	0.000	0.001	0.001	0.002	0.003	0.028	0.496	4.741	28.883	440.429
WiT	Initial area	142	94	116	32	5,301	78,200	5,629	15,092	19,624	156,489	8,139,250	2,840,420	4,831,845	45,677,120
	Initial runtime (s)	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	0.015	0.042	0.109	0.425
	Final area	**142**	**90**	**116**	**32**	**5,301**	**74,200**	**5,513**	**13,728**	**17,044**	**144,071**	**7,151,286**	**2,325,806**	**4,205,757**	**37,318,054**
	Gain	0	4	0	0	0	4,000	116	1,364	2,580	12,418	987,964	514,614	626,088	8,359,066
	#Iterations	**0**	**1**	**0**	**0**	**0**	**3**	**2**	**17**	**37**	**78**	**415**	**949**	**1,197**	**3,487**
	Total runtime (s)	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	0.016	0.297	1.794	6.489	89.497

Remarks: 1. #Terminals = #Sources + #Sinks.
Area: wire area. Refined area: area after refinement. Initial area: area of our initial solution. Final area: area of WiT.
Gain: area reduction by refinement. Refine runtime: runtime of refinement. #Iterations: number of refinement iterations.
2. Refinement is negative-cycle removal in the WiT algorithm.
3. [9] may generate infeasible solutions (labeled as FAIL). [9] is not refined because it is infeasible for some cases.

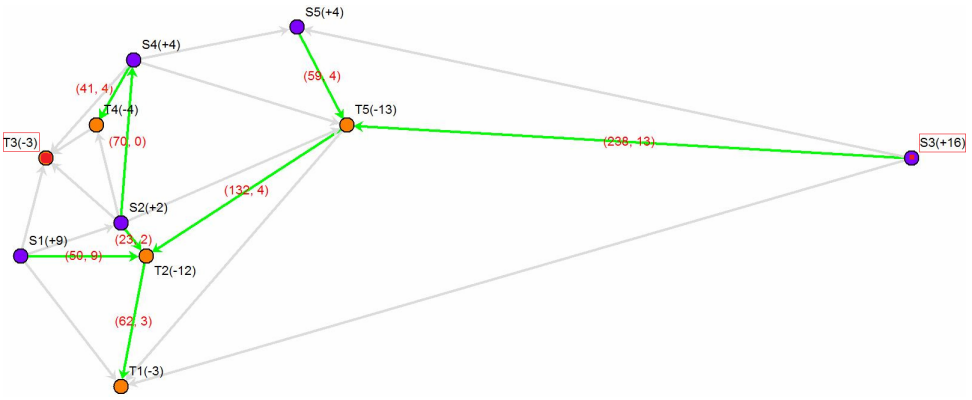
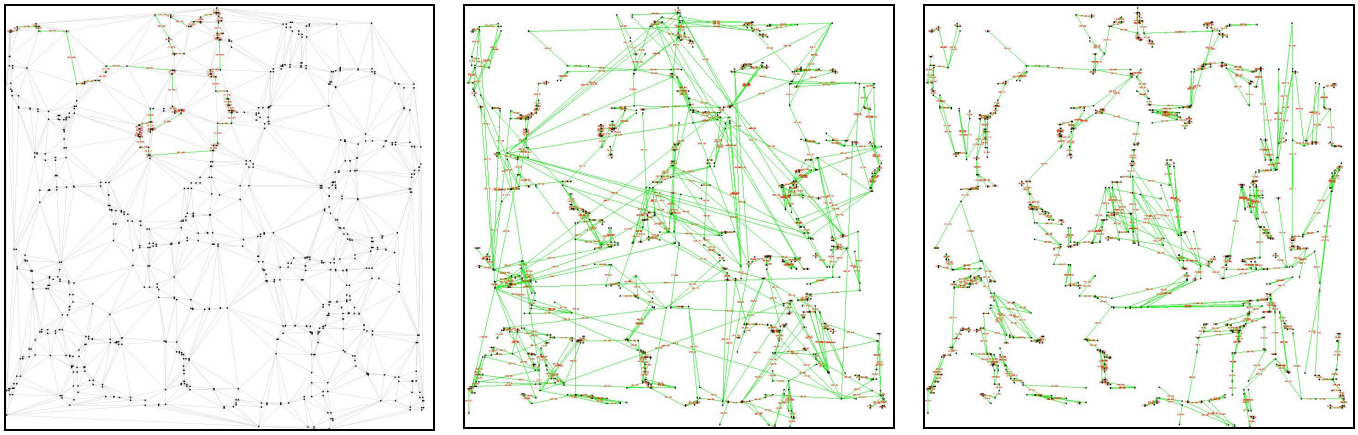


Figure 12. IND3 by [9]. If sink T3 connects source S3, it is feasible.



(a) [9]: failed (b) [10]: area = 59,207,240 (c) Ours: area = 37,318,054

Figure 13. RT05.