On Minimum-Cost Rectilinear Steiner Distance-Preserving Tree

Jun-Dong Cho†

ABSTRACT

Given a signal net $N = s, 1,\ldots, n$ to be the set of nodes, with $s$ the source and the remaining nodes sinks, an MRDPT (minimum-cost rectilinear Steiner distance-preserving tree) has the property that the length of every source to sink path is equal to the rectilinear distance between the source and sink. The minimum-cost rectilinear Steiner distance-preserving tree minimizes the total wire length while maintaining minimal source to sink length. Recently, some heuristic algorithms have been proposed for the problem of finding the MRDPT. In this paper, we investigate an optimal structure on the MRDPT and present a theoretical breakthrough which shows that the min-cost flow formulation leads to an efficient $O((n^2 \log m)^2)$ time algorithm. A more practical extension is also investigated along with interesting open problems.

1. Introduction

With the scaling of device technology and die size, interconnection delay now contributes up to 70% of the clock cycle in dense, high performance circuits. This paper considers the problem of optimizing performance-driven system design minimizing construction cost, signal delay and clock skew. The difference in arrival times between a single pulse arriving at two different clocked components is referred to as clock skew.

There have been many works related to the high-performance interconnection physical designs [11, 6]: Skew and delay optimization for reliable buffered clock trees [15], Clock-period constrained buffer insertion in clock trees [20], Buffer distribution algorithm for high-speed clock routing [2], Activity-driven clock design for lower-power circuits [19], Bounded-skew clock and Steiner routing [4], Buffer insertion and sizing under process variations for lower power clock distribution [23], Performance-driven Steiner tree
constructions [12, 5, 10, 21, 16].

Our algorithm provides a cost-and delay-minimum clock tree topology to be used in the above specific clock network design applications. We define a signal net \( N = \{s, 1, \ldots, n\} \) to be the set of nodes, with \( s \) the source and the remaining nodes sinks. The signal net \( N \) is to be embedded in an underlying graph \( G = (V, E) \) with \( N \in V \). The Hanan's grid [9] is generated by drawing a horizontal straight line and a vertical straight line crossing each node in \( N \). The graph \( G \) is associated with the intersections of the Hanan's grid as a set of nodes \( V \) and has variable edge costs; each edge \((i, j) \in E\) has a cost \( d(i, j) \) equal to the routing cost between node \( i = (x_i, y_i) \) and node \( j = (x_j, y_j) \), i.e., the rectilinear distance between the two nodes \( (|x_i - x_j| + |y_i - y_j|) \). A set of Hanan's nodes denoted by \( H \) is the set of intersections excluding sinks. The Steiner tree [22] is a routing tree \( T \) in \( G \) soaks \( N \). A Steiner tree for a set \( N \) may contain at most \( n - 2 \) other node set \( S \in H \) called Steiner nodes on the plane. The cost of \( T \) is defined to be \( \text{cost}(T) = \sum_{(i, j) \in T} d(i, j) \). The Minimum Rectilinear Steiner Tree (MRRST) problem is given a set of \( N \) of \( n \) nodes in the plane, to determine a set \( S \) of Steiner nodes such that the tree cost over \( N \cup S \) has minimum rectilinear cost. The problem are known to be NP-hard since a long time [7]. The MRST approach for interconnecting the terminals of a clock net is not necessarily the best one in terms of various applications [16].

The shortest distance between the source and a given sink \( i \) in \( G \) is denoted as \( r_i \). The shortest distance between the source and given sink \( i \) in \( T \) is denoted as \( w_i \). The problem of min-cost rectilinear Steiner Distance-preserving tree (MRDPT) [18] (see Figure 1b), which are also called Min Cost Rectilinear Steiner Arborescence [17] and Min Cost Shortest Path Steiner Tree [11], seeks a minimum-cost tree that has a special property such as \( w_i = r_i \) for every sink \( i \). It is known that finding the BRDPT in general graphs is NP-hard [3]. The complexity of the problem in a planar graph has not been known. See [11] for the history. In this paper, we present an efficient \( O((n^2 \log n)^2) \) algorithm to identify MRDPT in \( G \) that is defined as above.

This paper is organized as follows. Section 2 will formulate the MRDPT problem using a 0-1 integer Linear Program. Section 3 will explore a transformation of the 0-1 integer Linear Program into a corresponding min-cost flow network. Section 4 will extend the algorithm to construct a more general version of the MRDPT called Load-constrained Multiple-Source Minimum Rectilinear Distance-Preserving Forest. Finally, conclusion will be in Section 5.

(Fig. 1) MRST, MRDPT and 1-Q MRDPT
2. Problem Formulation

A typical approach of finding MRDPT as in [18] is as follows. 1) partition the plane into quadrants \( Q_0, Q_1, Q_2, \) and \( Q_3 \). The partitioning of the plane divides the sinks into one-quadrant MRDPT problem (1Q MRDPT) as depicted in Figure 1c.

2) Solve the 1Q MRDPT problem for \( Q_0, Q_1, Q_2, \) and \( Q_3 \), independently, obtaining an MRDPT \( T(Q_0), T(Q_1), T(Q_2), \) and \( T(Q_3) \).

3) Merge the solutions for each quadrant, thus obtaining the MRDPT T. [18] presented a dynamic programming heuristic on 1Q MRDPT. Given the solutions for each quadrant, the MRDPT T can be found in polynomial time.

In this paper, we are only concerned with finding 1Q MRDPT (refer to Figure 1c), \( T(Q_0) \), whose sinks are in quadrant \( Q_0 \) (i.e., right upper corner of the plane). Given two sink nodes \( i \) and \( j \), \( i \) is said to dominate \( j \), denoted \( i \rangle j \), if \( x_i \rangle x_j \) and \( y_i \rangle y_j \). Similarly, \( i \) and \( j \) are said to be independent if \( i \) does not dominate \( j \) and \( j \) does not dominate \( i \), denoted \( i \not\rangle j \). A layer \( L \) is a set of independent nodes. We shall call the farthest layer from the source outmost layer. We denote by \( H' \) a set of reduced Hanan’s nodes that excludes the intersections outside the outmost layer (see Figure 1c). The underlying graph of the 1Q MRDPT (we shall call it just MRDPT) is called flow graph, \( G^d=(V=(H' \cup N), A) \), such that there is a directed arc in \( A \) from \( i \) to \( j \) in \( V \) if \( r_j \leq r_i \) \( x_j \leq x_i \) and \( y_j \leq y_i \). That is, every arc is oriented toward the source \( s \). Thus, arcs are embedded using some monotone or “staircase” path between the source to any sink.

A tree is a directed in-tree rooted at node \( s \) if the unique path in the tree from any node to node \( s \) is a directed path called s-path. Observe that every node in the directed in-tree has outdegree 1. Thus the problem of MRDPT is to find a directed in-tree from \( G^d \) with min-cost.

Let us first review the NP-completeness proof of the MRDPT in general graphs, in this paper, especially transformed from the problem of the Minimum Edge-Cost Flow (MECF) as follows.

Minimum Edge-Cost Flow (MECF):

- Let \( G=(V, A) \) be a directed network with a nonnegative cost \( w(i, j) \in Z^+ \) and a capacity \( c(i, j) \in Z^+ \) associated with every arc \( (i, j) \in A \).

With given requirement flow \( f \in Z^+ \) and bound \( B \in Z^+ \), the problem is to ask for a minimum arc cost flow that is to be shipped from \( s \) to \( t \). The variable \( f(i, j) \) denotes a flow on arc \( (i, j) \in A \).

(1) capacity constraint: \( f(i, j) \leq c(i, j) \), \( \forall (i, j) \in A, \) i.e., where \( c(i, j) \) indicates the maximum amount of flow that can go through the arc \( (i, j) \).

(2) flow conservation constraints: for \( v \in V \), let the net flow at \( v \) be defined as \( f(v) = \sum f(i, j) - \sum f(j, i) \), \( \forall v \in V \), \( f(t) = 0 \), \( \forall v \in V \) and \( f(s) = -f \), \( f(t) = f \), where \( f \) is the require amount of flow to be shipped.

(3) if \( A' = \{(i, j) \in A : f(i, j) \neq 0\} \), then \( \sum_{(i, j) \in A'} w(i, j) \leq B \).

The problem is known to be NP-complete [8]. Note that our goal in MRDPT is to maximize the overlap of s-path segments for cost savings so as to minimize the total tree cost among all direct in-trees. In other words, the cost savings of an MRDPT can be represented as \( Z^* = \sum_{(i, j) \in A'} w(i, j) \times f(i, j) - 1 \).

Theorem 2.1 The min-cost Steiner distance-preserving tree problem in general graphs is NP-complete.

Proof: Let us assume that all the paths from a sink to the source are disjoint yielding the tree cost of \( Z^* = \sum_{v \in sink} r_v \). We know that the \( Z^* \) is the known upper bound on the cost of MRDPT. Then, the tree cost of an MRDPT \( T, \) \( Z_T = Z^* - Z_T' = \sum_{(i, j) \in A'} w(i, j) \times f(i, j) - \sum_{(i, j) \in A'} w(i, j) \times f(i, j) \).
\[ z = \sum_{(i, j) \in A} w(i, j), \text{ where } A' = \{(i, j) \in A : f(i, j) \neq 0\}. \]

Since the solution to the MRDPT is to find a minimum \( Z \), for all rectilinear Steiner distance-preserving trees, it is equivalent to find an MECF in \( G^d \), the theorem holds.

However, it is hard to reduce MRDPT in planar graphs to MECF directly since MECF is not restricted to planar graphs. The problem of MECF is solvable in polynomial time if (3) is replaced by \( \sum_{(i, j) \in A} w(i, j) f(i, j) \leq B \), which becomes the well-known Min Cost Flow (MCF) Problem.

**Min Cost Flow (MCF):**
- The MCF is, given amount of flow, to ask for a min-cost flow, minimizing \( z = \sum_{(i, j) \in A'} w(i, j) f(i, j) \) to be shipped from \( s \) (source) to \( t \) (sink), satisfying the capacity and flow conservation constraints [1].

Note that all paths from a sink to the source is of the same length in \( G^d \). Therefore, the objective function of the MCF is of the same value over all feasible solutions to the rectilinear Steiner distance-preserving tree. However, the object function of MRDPT (i.e., maximize \( \sum_{(i, j) \in A} w(i, j) (f(i, j) - 1) \)) is very similar to the object function of MCF (i.e., minimize \( \sum_{(i, j) \in A} w(i, j) f(i, j) \)). The MRDPT has a special property that the underlying graph \( G_A \) does not contain a directed cycle and it is planar. Motivated by the facts, in this paper, we will explore the optimality structure of MRDPT by transforming the flow graph \( G^d \) into a transformed flow graph \( G^R \) such that an MCF on \( G^R \) yields an MRDPT.

We formally formulate the problem in 0-1 integer linear programming as follows (refer to Figure 3).

**0-1-LP-MRDPT:**
- Minimize

\[ z = \sum_{(i, j) \in (G \cup H)} (w(i, j) x(i_1, j) + w(i_2, j) x(i_2, j)) + w(j, o_1) x(j, o_1) + w(j, o_2) x(j, o_2), \]

subject to

**Constraint 1:**
\[ x(i_1, j) = x(i_2, j) = x(j, o_1) = x(j, o_2) = 0, 1, \quad \forall j \in N \cup H, \]

**Constraint 2:**
\[ x(j, o_1) + x(j, o_2) = 1, \quad \forall j \in N \]

**Constraint 3:**
\[ 2 \geq 2(x(j, o_1) + x(j, o_2)) \geq (x(i_1, j) + x(i_2, j)) \leq (x(j, o_1) + x(j, o_2)), \quad \forall j \in H, \]

where \( x(i_1, j) \) and \( x(i_2, j) \) are \( 0, 1 \) variables associated with two incoming arcs entering node \( j \) and \( x(j, o_1) \) and \( x(j, o_2) \) are associated with two outgoing arcs emanating from node \( j \), respectively.

We will refer to \( i_1 \) or \( i_2 \) as a tail of \( j \) and \( o_1 \) or \( o_2 \) as a head of \( j \), respectively. In the first constraint, \( x(i_1, j) \) is being an edge of MRDPT. The second constraint is obvious and due to the fact that every sink node in the directed in-tree has an out-degree 1. The third constraint for the nodes in \( H \) is due to the other properties of MRDPT that can be represented as following two if-then-else statements.

\[
\begin{align*}
&\text{if } x(j, o_1) + x(j, o_2) = 0 \text{ then } x(i_1, j) + x(i_2, j) = 0 \\
&\text{if } x(i_1, j) + x(i_2, j) = 0 \text{ then } x(j, o_1) + x(j, o_2) = 0
\end{align*}
\]

Thus, we have the following theorem.

**Theorem 2.2** The solution to 0-1-LP-MRDPT generates an MRDPT.

**Proof:** There are six cases that does not satisfy the properties of the rectilinear distance-preserving tree as shown in Figure 2. The case of \( a \) is avoided by Constraint 2 for \( N \) and Constraint 3 for \( H \), and the cases of \( b, c, d, g \) for \( N \) and the cases of \( b, c, d, f \) for \( H \) are avoided by Constraint 3 of the 0-1-LP-MRDPT formulation, respectively. Thus, they never occur in the solution of the LP. While satisfying the constraints, minimizing the cost function \( z \) is equivalent to finding an MRDPT.
3. Transformation to Min-Cost Flow Network

Let us test now if the 0-1-LP-MRDPT has a special structure that permits us to solve the problem more efficiently than general-purpose linear programs.

Suppose that a matrix A is a 0-1 matrix satisfying the property that all of the 1's in each column appear consecutively (i.e., with no intervening zeros). This problem can be transformed into a min-cost flow problem [1]. However, we have not yet been able to find the property in our MRDPT. We also could not transform the MRDPT to the linear assignment (minimum edge-weighted matching in a complete bipartite graph) problem. In this section, we present a novel network transformation that makes the MRDPT tractable using the minimum cost flow.

Suppose we are given a graph \( G^B = (V^B = V_1 \cup V_2, A^B) \) (refer to Figure 3), where \( A^B \) denotes the set of arcs connecting nodes in \( V_1 \) to the nodes in \( V_2 \), with a set of arc weights \( \{w(i, j) : (i, j) \in A^B\} \) and a set of arc capacities \( \{c(i, j) : (i, j) \in A^B\} \). To construct the transformed \( G^B \) we model each arc of \( G^A \) as one of nodes in \( G^B \), and introduce two dummy nodes \( I \) and \( O \) (resp. \( I(j) \) and \( O(j) \)) such that \( I(j) \) (resp. \( O(j) \)) is incident to both \( j \) and \( j' \)'s two incoming (resp. outgoing) arcs in \( G^A \). That is, the transformed graph \( G^B \) is obtained directly from \( G^A \) as follows:

\[
V_1 = \{I(j) \cup O(j) \mid \forall j \in (N(s) \cup H) \cup I(s)\} (1) \\
V_2 = \{x(i, j) \cup x(j, k) \cup j \mid \forall j \in (N \cup H) \land \forall i, k\} (2)
\]

where \( i \) is a tail of \( j \) and \( k \) is a head of \( j \).

Next, we introduce two new nodes, a source node \( s \) and a sink node \( t \). For each node \( u \in V_1 \), we add an arc \((s, u)\) with capacity \( c(s, u) \), and for each node \( v \in V_2 \), we add an arc \((v, t)\) with capacity \( c(v, t) \).

Each arc \((u, v)\) in \( G^B \) connects two nodes \( u \in V_1 \) and \( v \in V_2 \) when

\[
(u, v) = (I(j), j), \forall j \in (N \cup H), \text{or} \quad (3) \\
(u, v) = (I(j), x(i, j)), \forall j \in (N \cup H), \text{or} \quad (4) \\
(u, v) = (O(j), j), \forall j \in H, \text{or} \quad (5) \\
(u, v) = (O(j), x(j, k)), \forall j \in (N \cup H), \text{or} \quad (6)
\]

where \( i \) is a tail of \( j \) and \( k \) is a head of \( j \).

Capacity \( c(u, v) \) and weight \( w(u, v) \) for each edge \((u, v)\) in \( G^B \) are assigned as follows, \( \forall j \in (N \cup H) \):

(Fig. 2) Infeasible patterns in the MRDPT

---

1The linear assignment problem can be solved using \( O(|V|^3) \) arithmetic operations by linear programming as done for example in [14].
(Fig. 3) An instance of $G^A$ and its transformed network flow graph $G^B$ (to simplify the illustration and without loss of generality, we ignored the two Hanan’s node; one in the upper-left corner and one in the lower-right corner, respectively.)

\[
c(s, u) = 2 \text{ if } (u = I(j))
\]
\[
c(s, u) = 1, \text{ otherwise}
\]
\[
c(u, t) = 2 \text{ if } (v = j)
\]
\[
c(u, t) = 1, \text{ otherwise}
\]
\[
c(u, v) = 2 \text{ if } (u \in I(j)) \text{ and } (v = j)
\]
\[
c(u, v) = 1, \text{ otherwise}
\]
\[
w(u, v) = d(i, j) \text{ if } (u = O(j)) \text{ and } (v = x(j, k))
\]
\[
w(u, v) = 0, \text{ otherwise}
\]

where $k$ is a head of $j$.

In an MRDPT solution $T$ after applying max flow on $G^B$, an interpretation of the max flow obtained is as follows:

1. if $f_{ij} \geq 1$ then $j \in H$ is being a Steiner node in $T$ or $j \in N$ is a sink node, where
   * if $f(I(j), j) = 1$, then $j$ has one incoming arc
An initial rectilinear distance preserving tree $T_1$ generated from obtaining a maxflow on $G^B$.

\[ f = f(s) = f(t) = \sum_{(i, j)} f(i, j) + \sum_{(j, k)} f(j, k) = 3 \times |N \cup H| - 1. \]

Here $f(j)$ implies the node flow balance constraints on node $j$ such that $f(j) = \sum_{i} f(i, j) - \sum_{k} f(j, k)$, where $i$ is a tail of $j$ and $k$ is a head of $j$. Let us also denote by $y(u, v)$ then 0-1 variable such that $y(u, v) = 1$ when $f(u, v) \geq 1$, $y(u, v) = 0$, otherwise.

**Lemma 3.1** A max-flow with $f$ flows on $G^B$ satisfying the condition.
\[ y(I(j), j) + y(O(j), j) = 1, \ \forall j \in H, \]

yields a rectilinear Steiner distance-preserving tree.

**Proof:** Refer to Figures 2, 3 and 4. It suffices to show that the graph constructed by applying a max-flow with the above interpretation satisfies: 1) for any sink there is a path joining it to the source; 2) it is a tree, i.e., no cycle exists; 3) there exists no infeasible patterns in Figure 2 in the graph. By sending a unit flow to each \( O(j) \), only one of outgoing arcs of \( O(j) \) should have a flow; thus 1) and 2) are satisfied and infeasible pattern of Figure 2a is avoided. Next, consider the remaining infeasible patterns to show 3) is true. Note that we send one unit flow to \( O(j) \) and two unit flows to each \( I(j) \) to generate all possible patterns. To avoid the infeasible patterns among them, we take advantage of the capacities assigned to an incoming arc and three outgoing arcs of \( I(j) \) and \( O(j) \), respectively and consider the following two cases. If \( y(I(j), j) + y(O(j), j) = 2 \), for \( j \in H \) (this is not the case for \( j \in N \) since there is no arc \( y(O(j), j) \) introduced in \( G^A \); thus the infeasible patterns of Figure 2c, d, and g are avoided.) in the max-flow, then one
of the infeasible pattern shown in Figure 2b, and d exists in the graph. Whereas, if \( y(I(j), j) + y(O(j), j) = 0 \), for \( j \in H \) in the max-flow, then the infeasible pattern shown in either Figure 2e or Figure 2f exists in the graph. These infeasible patterns can be avoided by enforcing \( y(I(j), j) + y(O(j), j) = 1 \) for each \( j \in H \).

Now, we shall show that the MRDPT can be found in polynomial time by obtaining a min-cost flow by sending \( f \) flows from node \( s \) to node \( t \) and satisfying the condition

\[ y(I(j), j) + y(O(j), j) = 1, \forall j \in H, \]

in the transformed network \( G^B \). We shall call the algorithm \textit{MCF-based MRDPT}. The algorithm maintains a feasible solution (that is, a rectilinear Steiner distance-preserving tree) in the transformed network \( G^B \) and at every iteration attempts to improve its objective function value. The following specifies this algorithm.

**Algorithm MCF-based-MRDPT:**

**Input:** A set of sinks and a source.

**Output:** An MRDPT;

begin

- construct a \( G^A \) and transform it into \( G^B \);
- establish an MRDPT by finding a min-cost flow in \( G^B \), minimizing \( z = \sum_{(u, v) \in A^B} w(u, v) f(u, v) \) while sending \( f \) flows to be shipped from \( s \) (source) to \( t \) (sink), satisfying the capacity and flow conservation constraints, also satisfying \( y(I(j), j) = y(O(j), j) = 1, \forall j \in H \);

end

**Theorem 3.1** The \textit{MCF-based-MRDPT} algorithm yields an optimal solution to the MRDPT and runs in \( O((n^3 \log n)^2) \) time, where \( n \) is the number of sinks.

**Proof:** A max-flow on \( G^B \) yields a rectilinear Steiner distance-preserving tree. Since \( f(u, v) = 1 \) when \( w(u, v) = 0 \) for \( u = O(j) \) and \( v = x(j, k) \) in \( G^B \) and \( x(j, k) \) corresponds to an arc in \( G^A \), the object function of MCF problem, i.e., \( \sum w(u, v) f(u, v) \), coincides with \( \sum w(u, v) \) which is the object function we wish to minimize in the MRDPT. Therefore, the problem can be solved using any min-cost flow algorithm in [1] while avoiding the condition, \( y(I(j), j) + y(O(j), j) = 1, \forall j \in H \) in \( G^B \). The time required to transform the \( G^A \) into \( G^B \) is \( O(n^2) \). Note that the number of arcs and nodes in \( G^B \) are (as shown in Figure 3): \( |V_1| = 2 \times |N \cup H| - 1, |V_2| = 3 \times |N \cup H|, |A^B| \leq 2 \times |N \cup H| \) (for arcs \( (s, u) \), \( u \in V_1 \)) + \( 6 \times |H| \) + \( 5 \times |N| \) (for arcs \( (u, v) \), \( u \in V_1 \) and \( v \in V_2 \)) + \( 3 \times |N \cup N| \) (for arcs \( (v, t), v \in V_2 \)). Thus \( O(|V^B|) = O(|A^B|) = O(n^2) \), where \( n \) is the number of sinks. Currently, the best strongly polynomial time algorithm for min-cost flow algorithm is due to Orlin [13]; it runs in \( O((m \log n)(m+n \log n)) \) time, where \( m \) is the number of edges and \( n \) is the number of nodes in a given graph \( G \).

4. Extension to Multiple-Source Minimum Rectilinear Distance-Preserving Forest with Sink Assignment

The algorithm proposed in the previous section can also be applied to \textit{sink assignment problem} stated as follows. Clock tree networks often contain many geometrically dispersed sinks (or clocked elements). These sinks need to be connected to a central clock source either by shortest connections or through a set of buffers \( B \). Each buffer that is connected to the source is capable of splitting signal flow streams to different sinks. Suppose that buffers are in place and that each buffer can handle at most \( K \) sinks. For each sink \( j \), let \( w(i, j) \) denotes the line the cost of constructing a line between buffer \( i \) to sink \( j \) or sink \( i \) to sink \( j \) or high-level buffer \( i \) to low-level buffer \( j \). The problem is to construct the min-cost network for
connecting the source to the sinks. The problem can be formulated with a min-cost flow problem as follows. Construct the graph \( G^A = (H \cup N, A) \), each node \( i \in B \in H \) denotes the buffer \( i \) and has an incoming arc from sink or buffer \( j \) with a capacity of \( K \) units (i.e., node capacity \( c_i = K \)). A min-cost flow solution on the transformed graph \( G^A \) (constructed in the previous section) with a particular node capacity \( c_i \) for each node (called \( G^C \)) will determine the min-cost network for connecting the terminals to the source subject to the node capacity. To incorporate the node capacity constraints into \( G^B \), we transform the node capacity into arc capacity by splitting the node into two nodes and assign the node capacity to the arc connecting the two nodes [1]. We refer the resulting tree as load-constrained MRDPT.

Next, suppose that a plane contains \( S \) sources. We cluster the sinks into \( g \) groups and let \( g_i \) denotes the number of sinks at group \( i \). The only restrictions on the groups is that they be clustered in a minimum cost and that there be a single distance measure \( d_{ij} \) that reasonably approximates the distance any sink at group \( i \) must connect if it is assigned to any type of node \( j \) which is either in \( B \) or \( S \). Each source or buffer \( i \) can load at most \( c_i \) sinks. The objective is to assign clusters to sources through buffers or directed paths in a manner that maintains the loading constraints and minimizes the total forest cost. We model this problem as a min-cost flow problem. Every source is connected to the super source using an arc with zero weight and push appropriate units of flow to the sources using the transformed graph \( G^C \). With the new graph called \( G^B \), we can apply the same algorithm that has been used for the load-constrained MRDPT.

As described above, The load-constrained multiple distinct source problem has an interesting application to more general version of the clock tree network optimization problem.

We refer to the generated tree as Multiple-Source Minimum Rectilinear Distance-Preserving Forest (MS-MRDPF). To incorporate the skew minimization on the clock network even considering lower power design, buffer insertion algorithm under process variations for lower power clock distribution in [23] can effectively use the MS-MRDPF as an initial topology.

5. Conclusion

We investigated a special class of Steiner minimal tree, called minimum rectilinear Steiner distance-preserving tree. We showed that the problem is tractable by transforming the problem into the min-cost flow problem. Due to the graph-theoretical proof of the optimality, experimentation is omitted for brevity. We also proposed an effective extension to the application of constructing a high-performance clock tree network.

For certain applications [16], one may wish to generate a minimum rectilinear Steiner tree imposing different wirelength constraints on different sources sink paths within a given signal net, since the critical timing is path-dependent rather than net-dependent. We can prohibit the assignment of a sink to source or buffer \( j \) if the distance (or delay) \( d_{ij} \) between these two locations exceeds some specific distance \( D_{ij} \). Thus an open problem is how to formulate the minimum rectilinear Steiner tree using the approach proposed in this paper and how to incorporate the wirelength or delay constraints into the formulation. This paper opens a new efficient approach to the well-known Steiner tree problem and its variants. Based on the algorithm developed in this paper, one of the future work is to develop and experiment a new and effective heuristic on well-known Minimum Rectilinear Steiner Tree construction.

REFERENCES


조 준 동
1980년 성균관대학교 전자공학과 졸업(학사)
1989년 Polytechnic University 전산학과(석사)
1993년 Northwestern University EECS DEpt.(박사)
1983년~1987년 삼성전자(주) CAD 연구원
1993년~1994년 삼성전자(주) CAD 수석연구원
1995년~현재 성균관대학교 전자공학과 조교수
관심분야: VLSI 설계 최적화, 알고리즘