Bounded-Skew Clock and Steiner Routing

JASON CONG, ANDREW B. KAHNG, CHENG-KOK KOH, and C.-W. ALBERT TSAO
University of California, Los Angeles

We study the minimum-cost bounded-skew routing tree problem under the pathlength (linear) and Elmore delay models. This problem captures several engineering tradeoffs in the design of routing topologies with controlled skew. Our bounded-skew routing algorithm, called the *BST/DME* algorithm, extends the DME algorithm for exact zero-skew trees via the concept of a *merging region*. For a *prescribed topology*, BST/DME constructs a bounded-skew tree (BST) in two phases: (i) a bottom-up phase to construct a binary tree of merging regions which represent the loci of possible embedding points of the internal nodes, and (ii) a top-down phase to determine the exact locations of the internal nodes. We present two approaches to construct the merging regions: (i) the *Boundary Merging and Embedding* (BME) method which utilizes merging points that are restricted to the *boundaries* of merging regions, and (ii) the *Interior Merging and Embedding* (IME) algorithm which employs a sampling strategy and a dynamic programming-based selection technique to consider merging points that are *interior* to, as well as on the boundary of, the merging regions. When the topology is not prescribed, we propose a new *Greedy-BST/DME* algorithm which combines the merging region computation with topology generation. The Greedy-BST/DME algorithm very closely matches the best known heuristics for the zero-skew case, and for the unbounded-skew case (i.e, the Steiner minimal tree problem). Experimental results show that our BST algorithms can produce a set of routing solutions with smooth skew and wirelength trade-off.

1. INTRODUCTION

In layout synthesis of high-performance systems, it has become increasingly important to control signal delay, e.g., for clock skew minimization or the timing-driven routing of large global nets. At the same time, a routing solution should have low wiring area to reduce the die size and the capacitive effect on both performance and power dissipation. Thus, the "zero-skew" clock tree and performance-driven routing literatures have seen rapid growth over the past several years; see [Kahng and Robins 1994] for a detailed review. Recent works have accomplished exact zero skew under the Elmore delay model [Tsay 1993; Chao et al. 1992b; Edahiro 1993a]. The *Deferred-Merge Embedding* (DME) algorithm by [Chao et al. 1992b; Edahiro 1992] can be either applied to a given clock topology or combined with a clock topology generation algorithm to achieve zero skew with a smaller wirelength [Edahiro 1993a]. Recent works have also given new methods for single-layer (planar) clock routing [Zhu and Dai 1996; Kahng and Tsao 1996]. Over the past two years, a number of authors have applied wiresizing optimizations and/or buffer optimizations to minimize phase (insertion) delay [Edahiro 1993b; Zhu et al. 1993; Pullela et al. 1996] (see also the many works in the clock buffer placement and sizing literature, as surveyed in [Friedman 1995]), skew sensitivity to process variation [Pullela et al. 1996; Chung and Cheng 1994; Lin and Wong 1994; Xi and Dai 1995], and/or power dissipation [Pullela et al. 1996; Vittal and Marek-Sadowska 1995]. A comprehensive survey of these optimization techniques can be found in [Cong et al. 1996a].

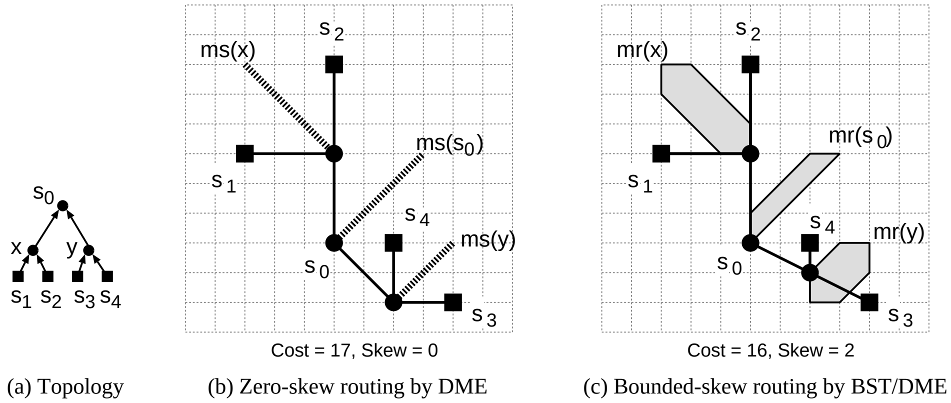


Fig. 1. Comparison of DME zero-skew routing in (b) and BST/DME bounded-skew routing in (c) for the prescribed topology G in (a). BST/DME lowers the routing cost by allowing non-zero skew bound. Note that in (b) the merging segments are depicted by dashed lines, and in (c) the merging regions are depicted by shaded polygons.

"Exact zero skew" is typically obtained at the expense of increased wiring area and higher power dissipation. In practice, circuits still operate correctly within some non-zero skew bound, and so the actual design requirement is for a *bounded-skew routing tree* (BST) [Kahng and Robins 1994]. In addition, works such as those in [Zhu et al. 1993; Pullela et al. 1996] use initial non-zero skew routing solutions which are then wiresized to satisfy a given skew bound; construction of an initial minimum-cost BST is a key underlying optimization. Thus, in this paper we study the BST problem under both the *pathlength* (linear) and *Elmore* delay models [Elmore 1948]. We propose the new BST/DME algorithm which, similar to the DME construction of a zero-skew tree, computes a routing tree for a prescribed topology using two bottom-up and top-down phases. The enabling concept is a *merging region*, which generalizes the merging segment concept of [Boese and Kahng 1992; Chao et al. 1992a; Edahiro 1991] for zero-skew clock trees. Fig. 1 highlights the difference between the DME algorithm for zero-skew routing and our proposed BST/DME algorithm for bounded-skew routing. In contrast to constructing merging segments in the zero-skew DME algorithm, the bottom-up process of our BST algorithms constructs a tree of merging regions, each containing possible locations of the corresponding internal node in the given topology. The top-down process then determines the exact locations of all internal nodes. Similar to the Greedy-DME algorithm by [Edahiro 1992; Edahiro 1993a; Edahiro 1994], we combine merging region computation with clock topology generation to achieve bounded-skew with a smaller wirelength. We refer to the new algorithm as the *Greedy-BST/DME* algorithm. A key distinction is that our Greedy-BST/DME algorithm allows merging at non-root nodes whereas Greedy-DME always merges two subtrees at their roots.

In this paper, we propose two approaches to construct the merging regions: (i) the *Boundary Merging and Embedding* (BME) method and (ii) the *Interior Merging and Embedding* (IME) method. The BME method uses merging points that are restricted to the *boundaries* of merging regions; each internal node has a merging region which is constructed from two segments on the boundaries of child merging region. We refer to a segment used for construction of the parent merging region as a *joining segment*. The IME method employs a sampling strategy and dynamic programming to consider merging points that are *interior* to, as well as on boundary segments of, the merging regions. The key difference is that the IME method uses a set of joining segments (possibly interior to the merging regions) from each child merging region, instead of only one joining segment. Merging two regions generates a set of merging regions for a parent node from the two child sets of joining segments. A key step in the IME method uses a dynamic programming-based selection technique to choose a *set* of "best" merging regions among the generated merging regions for the parent node.

We show several interesting properties of the merging region under bounded-skew routing. For the pathlength delay model, we prove that the merging region is bounded by *well-behaved segments* which are Manhattan arcs ($\pm 45°$ lines) and rectilinear line segments (horizontal or vertical line segments). The skew value along a boundary Manhattan arc is constant, and the skew values along boundary rectilinear line segments are linearly decreasing, then constant, then linearly increasing. We also prove that the merging region is a convex polygon with at most 8 boundary segments; hence, a merging region can be constructed in constant time. Under Elmore delay, the well-behaved property generalizes such that the merging region is bounded by segments (with arbitrary slopes) on which the skew values are piecewise-linear decreasing, then constant, then piecewise-linear increasing. The merging region is a convex polygon with at most $2n + 4$ boundary segments, where n is the number of leaf nodes in the tree topology; a given merging region can be constructed in $O(n)$ time. One minor caveat is that the "merging region" that we construct is not a complete generalization of the DME merging segment: when so-called detour wiring is needed or when sibling merging regions overlap, our construction may not return a merging region containing all the minimum-cost merging points.

In practice, bounding pathlength skew does not provide reliable control of actual delay skew. Fig. 2(a) plots HSPICE delay skew against pathlength delay skew for routing trees generated by our BST/DME algorithm under pathlength delay on MCNC benchmark r3 [Tsay 1993]. Not only is the correlation poor, but the pathlength-based BST solutions simply cannot meet a tight skew bound (of $100ps$ or less). On the other hand, Fig. 2(b) demonstrates the accuracy and fidelity of Elmore delay skew to actual skew; cf. [Boese et al. 1995]. Nevertheless, the BST problem under the pathlength delay model is theoretically interesting. Moreover, the pathlength delay formulation provides a better platform to present our BST algorithms due to the simplicity of the well-behaved property and the regularity of the merging regions under pathlength delay. Thus, we present the basic approaches of our work (both BME and IME) under the pathlength delay formulation but present only the experimental results obtained by the Elmore-based BST solutions.

The rest of the paper is organized as follows: In Section 2, we give several basic definitions and formulate the bounded-skew routing tree problem. We also review previous DME-based methods for zero-skew routing. In Section 3, we develop our results on the merging region construction and present the Boundary Merging and

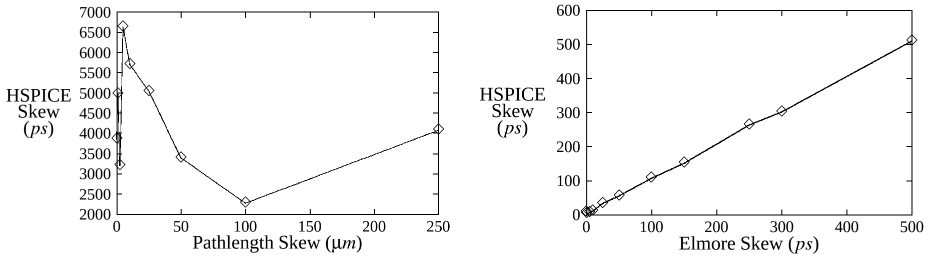


Fig. 2. Plots of (a) pathlength skew and (b) Elmore delay skew versus actual (HSPICE simulation) delay skew for routing solutions obtained by our BST/DME algorithm under pathlength delay and Elmore delay for benchmark r3.

Embedding method under the pathlength and Elmore delay models. In Section 4, we present the Interior Merging and Embedding method which can also be applied to both pathlength and Elmore delay models. The BST/DME algorithms presented in Sections 3 and 4 assume a prescribed clock tree topology. In Section 5, we present a Greedy-BST/DME algorithm which computes merging region and constructs the clock tree topology concurrently in the bottom-up phase. The proposed topology construction algorithm is a generalization of the Greedy-DME algorithm by [Edahiro 1992]. Section 6 summarizes experimental results and Section 7 concludes with directions for future work. All proofs of lemmas and theorems are presented in the Appendix. Early partial results from this work were presented in [Huang et al. 1995; Cong and Koh 1995; Cong et al. 1995a].

2. PRELIMINARIES

Assume that we are given a set $\mathcal{S} = \{s_1, s_2, \cdots, s_n\} \subset \mathcal{R}^2$ of clock sink locations in the Manhattan plane. A clock source location s_0 may also be given. A *routing topology* G is a rooted binary tree with n leaf nodes corresponding to the sinks in S. A clock tree $T_G(S)$ is an embedding of routing topology G, i.e. each internal node $v \in G$ is mapped to a location $l(v)$ in the Manhattan plane. (If G and/or S are understood, we may simply use $T(S)$ or T to denote the clock tree.) In the rooted topology, each node v is connected to its parent by edge e_v, and the cost of edge e_v is its wirelength, denoted by $|e_v|$. The cost of a routing tree T, denoted $cost(T)$, is the sum of its edge costs. If $t(u, v)$ denotes the signal delay between nodes u and v, then the *skew* of clock tree T is given by

$$skew(T) = \max_{s_i, s_j \in S} |t(s_0, s_i) - t(s_0, s_j)|.$$

We address the following problem:

Minimum-Cost Bounded Skew Routing Tree (BST) Problem: Given a set $S = \{s_1, ..., s_n\} \subset \mathcal{R}^2$ of sink locations and a skew bound B, find a routing topology G and a minimum-cost clock tree $T_G(S)$ that satisfies $skew(T_G(S)) \leq B$.

We will consider both this formulation and the BST variant where a fixed topology G has been specified. We do not include clock source location s_0 in our formulation since our methods can transparently accommodate any prescribed s_0.

Note that when the skew is unbounded, the problem becomes the classic rectilinear Steiner minimum tree problem.

Our work addresses the BST problem under the pathlength delay and the Elmore delay models. For any nodes $u, v \in G$ with u an ancestor of v, let $Path(u, v)$ denote the unique path from u to v in G. Under the pathlength delay model, the delay from u to v is the sum of edgelengths in the unique u-v path, i.e.,

$$t(u,v) = \sum_{e_w \in Path(u,v)} |e_w|.$$

The Elmore delay model is defined as follows: Let r and c denote the unit length wire resistance and capacitance, respectively. Then, the wire resistance and capacitance of edge e_v are $|e_v| \cdot r$ and $|e_v| \cdot c$, respectively. For any node v in G, we use T_v to denote the subtree of T that is rooted at v, and we use $Cap(v)$ to denote the total capacitance of T_v. Then, under the Elmore model the signal delay $t(u, v)$ is given by

$$t(u,v) = \sum_{e_w \in Path(u,v)} |e_w| \cdot r \cdot (\frac{|e_w| \cdot c}{2} + Cap(w)),$$

which can be computed in linear time recursively [Tsay 1993].

2.1 The DME Approach

The Deferred-Merge Embedding (DME) algorithm, proposed independently in [Boese and Kahng 1992; Chao et al. 1992a; Edahiro 1991], achieves exact zero skew given any delay model for which sink delays are monotone in the length of each edge of the clock tree (e.g., pathlength delay and Elmore delay). For pathlength delay, DME returns the optimal solution, i.e., a tree with minimum cost and minimum source-sink pathlength for any input sink set S and topology G.

Since the BME and IME methods we propose below are generalizations of DME, we now review the original DME method, following notations of [Chao et al. 1992b]. We use $d(s, t)$ to denote the Manhattan distance between points s and t; the distance between two pointsets P and Q is $d(P, Q) = min\{d(p, q) | p \in P, q \in Q\}$.

I. The DME Algorithm. Given a set of sinks S and a topology G, DME embeds internal nodes of G via: (i) a bottom-up phase that constructs a tree of *merging segments* which represent loci of possible placements of internal nodes in a zero-skew tree (ZST) T; and (ii) a top-down embedding phase that determines exact locations for the internal nodes in T (see Fig. 3).

In the **bottom-up phase**, each node $v \in G$ is associated with a merging segment, denoted $ms(v)$, which represents a set of possible placements of v in a minimum-cost ZST. The segment $ms(v)$ will always be a *Manhattan arc*, i.e., a segment with possibly zero length that has slope $+1$ or -1. Let a and b be the children of node v, and let TS_a and TS_b denote the subtrees of merging segments rooted at a and b. The construction of $ms(v)$ depends on $ms(a)$ and $ms(b)$, hence the bottom-up processing order. We seek placements of v which allow TS_a and TS_b to be merged with *minimum* added wire $|e_a| + |e_b|$ while preserving zero skew in T_v. The construction of $mr(v)$ is detailed in [Chao et al. 1992b]. *Detour wiring* occurs when $|e_a| + |e_b| > d(ms(a), ms(b))$.

Procedure Build_Tree_of_Segments (G,S)
Input: Topology G; set of sink locations S
Output: Tree of merging segments TS containing $ms(v)$ for each node v in G, and edge length $\|e_v\|$ for each $v \neq s_0$

for each node v in G (bottom-up order)
  if v is a sink node,
    $ms(v) \leftarrow \{l(v)\}$
  else
    calculate $ms(v)$ by DME construction rules [Chao et al. 1992b]

(a) Bottom-up Phase: Construction of the tree of merging segments TS.

Procedure Find_Exact_Placements(TS)
Input: Tree of segments TS containing $ms(v)$, and value of $\|e_v\|$ for each node v in G
Output: ZST $T(S)$

for each internal node v in G (top-down order)
  if v is the root
    Choose any $l(v) \in ms(v)$
  else
    Let p be the parent node of v
    Choose any $l(v) \in ms(v)$ s. t. $d(l(v), l(p)) \leq \|e_v\|$

(b) Top-down Phase: Construction of the ZST by embedding internal nodes of G within TS.

Fig. 3. The DME algorithm.

Given the tree of merging segments, the **top-down phase** embeds each internal node v of G as follows: (i) if v is the root node, then DME selects any point in $ms(v)$ to be $l(v)$; or (ii) if v is an internal node other than the root, DME chooses $l(v)$ to be any point on $ms(v)$ that is at distance $|e_v|$ or less from the embedding location of v's parent. Details of the embedding rules are also given in [Chao et al. 1992b].

Fig. 1(b) gives an example of the DME algorithm for a clock source s_0 and sinks s_1-s_4 with a topology shown in Fig. 1(a). Merging segments $ms(x)$, $ms(y)$, and $ms(s_0)$ are constructed in bottom-up order, then each node is embedded at a point on its merging segment that is closest to its parent. Fig. 1(b) gives the zero-skew clock tree with a total wirelength of 17 units after the top-down embedding.

II. Greedy-DME: DME with Topology Construction Note that DME requires an input topology. Several works [Boese and Kahng 1992; Chao et al. 1992a; Edahiro 1992] have thus studied topology constructions that lead to low-cost routing solutions when DME is applied; the most successful is the Greedy-DME method of [Edahiro 1992] which determines the topology of the merging tree in a greedy bottom-up fashion. Let $\mathcal{F}$ denote a set of merging segments which initially consists of all the sink locations, i.e., $\mathcal{F} = \{ms(s_i)\}$. Greedy-DME iteratively finds the pair of nearest neighbors in $\mathcal{F}$, that is, $ms(a)$ and $ms(b)$ such that $d(ms(a), ms(b))$ is minimum. A merging segment $ms(v)$ is computed for parent node v from a zero-skew merge of $ms(a)$ and $ms(b)$; $\mathcal{F}$ is updated by adding $ms(v)$ and deleting both

$ms(a)$ and $ms(b)$. Note that $\mathcal{F}$ contains only merging segments of subtree roots. After $n-1$ iterations, $\mathcal{F}$ contains the merging segment for the root of the topology.

In [Edahiro 1993a], $O(n \log n)$ time complexity was achieved by finding several nearest-neighbor pairs at once, i.e., the algorithm first constructs a "nearest-neighbor graph" H over $\mathcal{F}$. Node v in H corresponds to merging segment $ms(v)$ in $\mathcal{F}$, with nodes u and v in H connected by edge E_{uv} if $ms(u)$ is the nearest neighbor of $ms(v)$ or $ms(v)$ is the nearest neighbor of $ms(u)$. The weight of edge E_{uv}, denoted $|E_{uv}|$, is simply the distance between $ms(u)$ and $ms(v)$. Via zero-skew merges, $|\mathcal{F}|/f$ nearest-neighbor pairs (edges) are taken from the graph in non-decreasing order of edge weight, where f is a constant typically between 2 and 4. The solution is improved by a post-processing local search that adjusts the resulting topology (cf. "CL+I6" in [Edahiro 1993a]). Greedy-DME achieves 20% reduction in wiring cost compared with the methods of [Chao et al. 1992b].

3. THE BOUNDARY MERGING AND EMBEDDING METHOD

We now propose the BST/DME algorithm which, similar to the DME algorithm for zero-skew routing, computes a bounded-skew routing tree for a prescribed topology using two bottom-up and top-down phases, as shown in Fig. 4. The key difference is that instead of constructing merging segments as in the DME algorithm, for each node $v \in G$ with children a and b we construct a *merging region* of v, denoted $mr(v)$, which is the set of all locations where the child merging regions $mr(a)$ and $mr(b)$ can be merged with minimum wiring cost while still maintaining the skew bound B. To efficiently compute merging regions, we propose the *Boundary Merging and Embedding* (BME) method, which considers only the merging points lying on the nearest boundary segments of $mr(a)$ and $mr(b)$. We first present the BME method under pathlength delay, and then extend it to handle the Elmore delay model.

3.1 Notations and Definitions

In the following, a *rectilinear* line segment is a horizontal or vertical line segment. An *octilinear polygon* is a convex polygon with boundaries defined by only Manhattan arcs and rectilinear line segments. Such a polygon, along with its interior, defines an *octilinear region*, which has at most 8 boundary line segments. To construct a merging region with minimum merging cost, we define the following terms.

Shortest Distance Region/Segment. For any two convex polygonal regions P and Q with boundaries $\partial(P)$ and $\partial(Q)$, the *shortest distance region* between P and Q, denoted $SDR(P,Q)$, is the set of points which have minimum sum of Manhattan distances to the boundaries of P and Q, i.e., $SDR(P,Q) = \{p | d(p, \partial(P)) + d(p, \partial(Q)) = d(\partial(P), \partial(Q))\}$. We always construct a merging region within the corresponding shortest distance region. When $\partial(P) \bigcap \partial(Q) = \emptyset$, then $SDR(P,Q)$ must be a polygonal region. However, if $\partial(P) \bigcap \partial(Q) \neq \emptyset$ the region $SDR(P,Q)$ may consist of multiple line segments or points, each of which is also a convex polygonal region (e.g., see $SDR(C,D)$ in Fig. 5). In such a case, we take, for simplicity, the longest of these segments to be $SDR(P,Q)$.[1] The *shortest*

[1] We can treat each of these line segments or points as a (convex polygonal) shortest distance region, within which a merging region will be constructed. Then, there will be multiple merging

Procedure Build_Tree_of_Merging_Regions (G,S,B)
Input: Topology G, set of sink locations S, and skew bound B
Output: Tree of merging regions TR

for each node v in G (bottom-up order)
  if v is a sink node
    $mr(v) \leftarrow \{l(v)\}$
  else
    Calculate $mr(v)$ by BME construction rules

(a) Bottom-up phase: Construction of tree of merging regions TR.

Procedure Find_Exact_Placements(TR)
Input: Tree of merging regions TR
Output: BST $T(S)$ with skew $\leq B$; value of $\lvert e_v \rvert$, $\forall v \in G$

for each internal node v in G (top-down order)
  if v is the root
    Choose any $l(v) \in mr(v)$
  else
    Let p be the parent node of v
    Let Q be the merging region of v's sibling node
    if $\lvert e_v \rvert$ not determined by BME construction rules
      $\lvert e_v \rvert \leftarrow d(JS_Q(mr(v)), l(p))$
    Choose any $l(v) \in JS_Q(mr(v))$ closest to $l(p)$

(b) Top-down phase: Construction of the BST by embedding internal nodes of G within merging regions of TR.

Fig. 4. The BST/DME algorithm for the prescribed topology.

distance segments between P and Q are defined as $SDS_Q(P) = \partial(P) \bigcap SDR(P,Q)$, and $SDS_P(Q) = \partial(Q) \bigcap SDR(P,Q)$.

Joining Segment. Let node v be an internal node with children a and b, which have merging regions $P = mr(a)$ and $Q = mr(b)$, respectively. The segments of P and Q that are used to construct the merging region $mr(v)$ are the *joining segments*, denoted $JS_Q(P)$ and $JS_P(Q)$. Since the goal of the BME method is to construct min-cost merging regions from the boundaries of P and Q, BME uses shortest distance segments as the joining segments, i.e., $JS_Q(P) = SDS_Q(P)$ and $JS_P(Q) = SDS_P(Q)$. We distinguish the joining segment and shortest distance segment concepts because actually any pair of line segments which are interior to or on the boundaries of the merging regions P and Q can be used to construct merging region $mr(v)$. The IME method in Section 4 considers joining segments that are interior to the merging regions.

Merging Region. Given a routing topology G, the *merging region* of each internal node $v \in G$, denoted $mr(v)$, is defined recursively as follows:

—If v is a sink s_i, then $mr(s_i) =$ sink location $\{l(s_i)\}$.

regions for nodes in the given topology. We can apply the IME method in Section 4 to deal with this case.

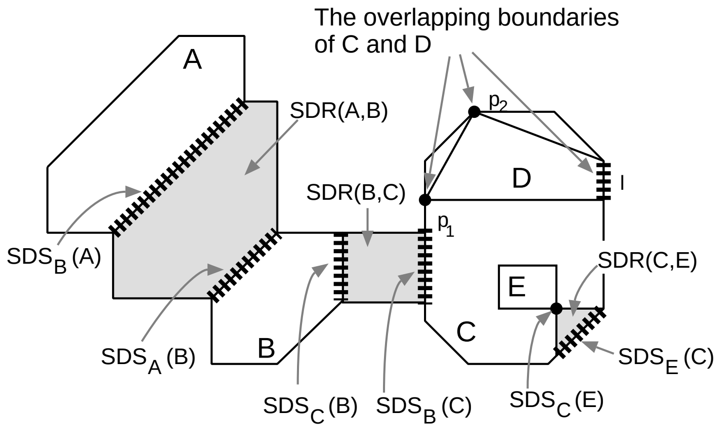


Fig. 5. Examples of Shortest Distance Segments (shown as thick dashed lines or solid dots), and Shortest Distance Regions (shown as dotted regions). Note that the boundaries of polygons C and D overlap at the two points p_1 and p_2 and at the line segment l. For simplicity, the BME method sets $SDR(C, D) = l$.

—If v is an internal node with children a and b, then let L_a and L_b be the joining segments of $mr(a)$ and $mr(b)$. The merging region $mr(v)$ is the set of possible locations of v such that
(i). $mr(v) \subseteq SDR(L_a, L_b)$,
(ii). the difference in (pathlength) delays from v to any two of sinks in T_v is within the skew bound, and
(iii). the merging cost $|e_a|+|e_b|$ is minimum, subject to the constraint that point $p \in L_a$ can merge with point $q \in L_b$ only if $d(p, q) = d(L_a, L_b)$.

Ideally, the min-cost merging region of a node v should be defined as the set of possible locations of v without condition (i) and without the constraint in (iii) However, under this ideal definition the merging region is very difficult to compute since it involves merging of points interior to merging regions, whose difficulty will be discussed in Section 4. Our definition enables an efficient construction of merging regions with a set of simple rules under both the pathlength delay and Elmore delay models (see Sections 3.2 and 3.4 below). Although a “merging region” constructed by the BME method does not necessarily contain all feasible merging points having minimum merging cost $|e_a| + |e_b|$, in practice, the merging region that we construct for each internal node of a “good topology” is precisely the “ideal” min-cost merging region.

Delay and Skew Functions. Let $max\_t(p)$ and $min\_t(p)$ denote the maximum and minimum delay values (max-delay and min-delay, for short) from point p to all leaf nodes in the subtree rooted at p. The skew of point p, denoted $skew(p)$, is $max\_t(p) - min\_t(p)$. (If all points of a pointset P have identical max-delay and min-delay, and hence identical skew, we similarly use the terms $max\_t(P)$, $min\_t(P)$ and $skew(P)$.) As p moves along any line segment the values of $max\_t(p)$ and $min\_t(p)$, together with $skew(p)$, respectively define the *delay* and *skew functions*

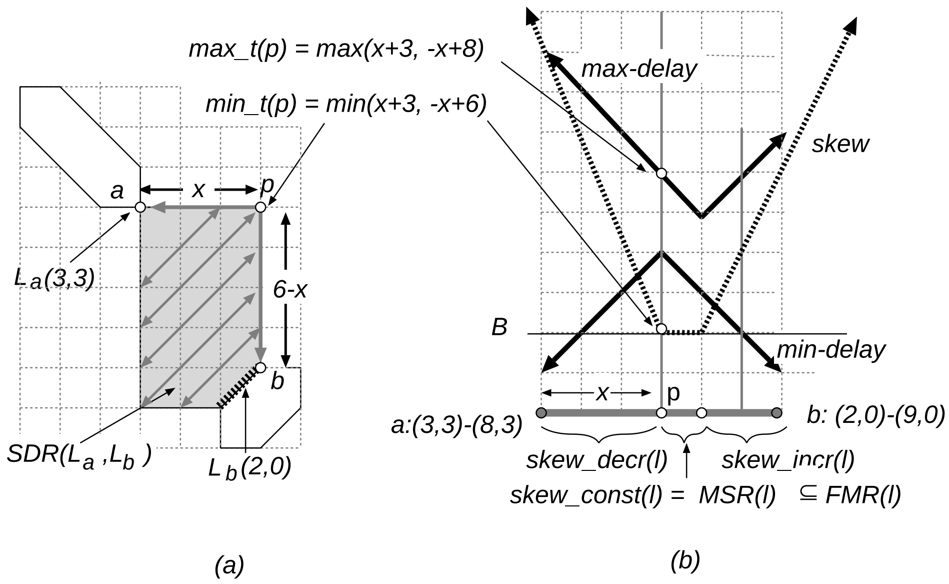


Fig. 6. (a) Merging $mr(a)$ with $mr(b)$ using joining segments L_a and L_b, respectively. The max-delay and min-delay of L_a and L_b are expressed as coordinate pairs. (b) Properties of pathlength delays and skew over the boundary segments $l = \overline{ap} + \overline{pb}$ of $SDR(L_a, L_b)$. The first and second coordinate pairs associated with points a and b represent (max-delay, min-delay) before and after merging, respectively.

over the segment. A point p with $skew(p) \leq B$ is a *feasible merging point*. For any region R, the *feasible merging region* $FMR(R)$ consists of all feasible merging points in R. Let $min\_skew(R)$ denote the minimum skew within a region R; then, the *minimum skew region* $MSR(R)$ is the set of points $p \in R$ with $skew(p) = min\_skew(R)$.

Well-Behaved Pathlength Delay Property. Fig. 6 illustrates the pathlength delays of points in shortest distance region $SDR(L_a, L_b)$ of Fig. 1(c), where joining segments L_a has max-delay and min-delay of 3 units and L_b has constant max-delay of 2 units and zero min-delay. To compute the merging region with minimum merging cost, each point $p \in SDR(L_a, L_b)$ is connected to L_a and L_b with the minimum wirelength $|e_a| = d(p, L_a) = x$ and $|e_b| = d(p, L_b) = d(L_a, L_b) - |e_a| = 6 - x$, respectively. So, the delay functions of p are

$$max\_t(p) = \max\{1 + |e_a|, \ 2 + |e_b|\} = \max\{x + 3, -x + 8\}, \qquad (1)$$

$$min\_t(p) = \min\{1 + |e_a|, \ 0 + |e_b|\} = \min\{x + 3, -x + 6\}, \qquad (2)$$

where $x = d(p, L_a)$ as shown in Fig. 6. We observe the following interesting delay properties for a line segment $l \subseteq SDR(L_a, L_b)$. First, for any Manhattan arc $l \subseteq SDR(L_a, L_b)$ which is parallel to L_a and L_b, all points of l have the same max-delay, min-delay, and skew. Second, from Eqns. (1) and (2) points p on any vertical or horizontal segment $l \subseteq SDR(L_a, L_b)$ have $skew(p) = \max\{2x - 3, -2x + 5, 2\}$, which is is a piecewise-linear function of x, as shown in Fig. 6(b). In light of these observations, we call these Manhattan arcs and rectilinear segments *well-behaved*.

Formally, a line segment $l = \overline{ab}$ is well-behaved if the max-delay and min-delay functions of point p on l are either (i) both constants or (ii) of the forms

$$max\_t(p) = \max\{x + \alpha, -x + \beta\}, \tag{3}$$
$$min\_t(p) = \min\{x + \alpha', -x + \beta'\}, \tag{4}$$

where $x = d(p, a)$ or $d(p, b)$. In other words, $max\_t(p)$ and $min\_t(p)$ are both piecewise linear (with slope $+1$ or -1) functions of the position of p on l, and $max\_t(p)$ ($min\_t(p)$) is a convex (concave) function whose value is minimum (maximum) at some point on l, and then increases (decreases) toward both endpoints of l. By the definition of skew,

$$skew(p) = \max\{2x + \tilde{\alpha}, -2x + \tilde{\beta},\ \gamma\},$$

where $x = d(p, a)$, $\tilde{\alpha} = \alpha - \beta'$, $\tilde{\beta} = \beta - \alpha'$, and $\gamma = \max\{\alpha - \alpha', \beta - \beta'\}$ are all constants, i.e., $skew(p)$ defined over a well-behaved rectilinear line segment l is a piecewise linear convex function with up to three linear pieces. Define a *turning point* on l as a point where the slope of a piecewise linear function defined over l changes. Then from Fig. 6(b) we can see that the max-delay and min-delay turning points each determine a skew turning point. Furthermore, the skew turning points divide l into three contiguous intervals (one or two of which may be empty), $skew\_decr(l)$, $skew\_const(l)$ and $skew\_incr(l)$, which correspond to the portions of l where the skew changes with slopes is -2, 0 and $+2$, respectively. Note that $skew\_const(l)$ is between two skew turning points, and may degenerate to a single point if the two skew turning points coincide. Obviously, $FMR(l)$ and $MSR(l)$ are each a single contiguous portion of l and can be computed in constant time.

Well-Behaved Region. Finally, a *well-behaved* region R is an octilinear region bounded by (i) well-behaved rectilinear segments and (ii) Manhattan arcs with constant max-delay and min-delay values.

3.2 Construction of the Merging Region

The following *BME construction rules* BM1-BM5 assume that for each internal node v with children a and b, merging regions $P = mr(a)$ and $Q = mr(b)$ are well-behaved. Since both P and Q are octilinear polygons, the joining segments $JS_Q(P) = SDS_Q(P)$ and $JS_P(Q) = SDS_P(Q)$ must be either (i) a pair of parallel Manhattan arcs (with constant min-delays and max-delays), or (ii) a pair of parallel (well-behaved) rectilinear line segments (see Fig. 7). To simplify notation in what follows, we let $L_a = JS_Q(P)$, and $L_b = JS_P(Q)$. Note that before merging P and Q we are given two delay functions defined over L_a and L_b, which will change after P and Q are merged. To avoid any confusion, we refer to the original delay functions defined for point p as $max\_\bar{t}(p)$ and $min\_\bar{t}(p)$, and refer to the new delay functions defined over l after merging as $max\_t(p)$ and $min\_t(p)$. Rules BM2-BM5 will use only the new delay functions defined over the segment $l \in SDR(L_a, L_b)$ when computing $FMR(l)$, $MSR(l)$, and skew turning points on l.

BM1. Compute joining segments $L_a = SDS_Q(P)$ and $L_b = SDS_P(Q)$.

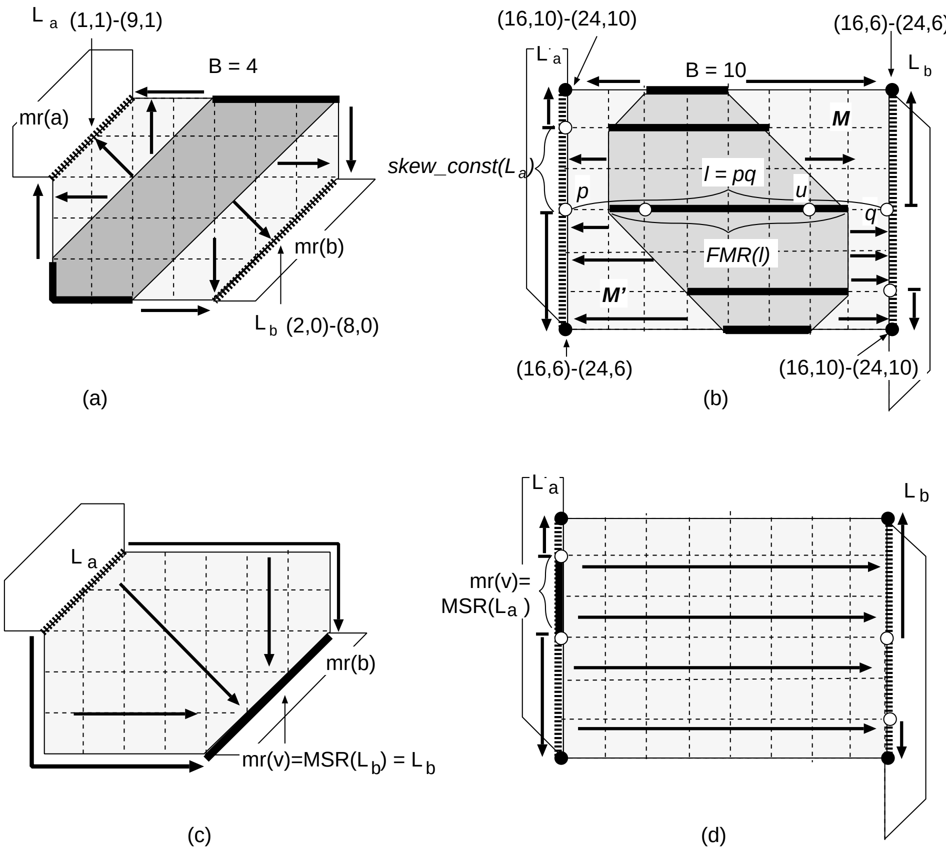


Fig. 7. Construction of merging region $mr(v)$, shown as shaded, given the merging regions for v's children a and b. The skew turning points of joining segments L_a and L_b (thick dotted lines) are shown as hollow points in (b) and (d). The first and second coordinate pairs associated with points on L_a and L_b represent (max-delay, min-delay) before and after merging, respectively. In (a) and (b), $mr(v) = FMR(R) \neq \emptyset$ divides R into two regions M and M' (dotted areas) where delays and skew values change monotonically as we traverse from the boundaries of $FMR(R)$ horizontally to L_a or L_b. In (c) and (d), $FMR(R) = \emptyset$, so the delays and skew values change monotonically in the horizontal direction in the whole region R. In this case, we set $mr(v) = MSR(L_b)$. Arrows indicate the directions of increasing skew/max-delay and decreasing min-delay. Note that in (a) and (c), delays and skew values also change monotonically along any shortest path from L_a to L_b.

BM2. Based on the given delay functions $max\_\bar{t}(p)$ and $min\_\bar{t}(p)$ defined for points p on L_a and L_b (before merging), for each boundary segment l of $SDR(L_a, L_b)$ (including L_a and L_b), compute the new delay functions, then compute the new skew functions, FMR's and skew turning points for segment l.

BM3. If L_a and L_b are parallel vertical (horizontal) segments, compute $FMR(l)$ for each horizontal (vertical) line segment $l = \overline{pq}$ such that $p \in L_a$, $q \in L_b$, and either p or q is a skew turning point of L_a or L_b (see Fig. 7(b)).

BM4. Let F be the set of FMRs computed via rules BM2 and BM3. If $F \neq \emptyset$, then $mr(v)$ is equal to the smallest convex polygonal region containing F.

BM5. If $F = \emptyset$, then $mr(v) = MSR(L_a)$ if $skew(MSR(L_a)) \leq skew(MSR(L_b))$, and $mr(v) = MSR(L_b)$ otherwise (see Fig. 7(c) and (d)). Since $skew(mr(v)) > B$, detour wiring is needed to meet skew bound constraint as follows. Let $x = d(L_a, L_b) + skew(mr(v)) - B$. If $skew(MSR(L_a)) \leq skew(MSR(L_b))$, $|e_a| = 0$ and $|e_b| = x$. Otherwise, $|e_b| = 0$ and $|e_a| = x$. In either case, the new delay functions at points $p \in mr(v)$ are computed as $min\_t(p) = max\_\bar{t}(p) - B$ and $max\_t(p) = max\_\bar{t}(p)$ (so that $skew(mr(v)) = B$).

Since the BME construction rules only compute $FMR(l)$ and $MSR(l)$ for a finite number of well-behaved segments l, and since each $FMR(l)$ or $MSR(l)$ can be computed in constant time, we have the following observation.

FACT 1. *It requires constant time to compute a merging region by using the BME construction rules.* □

The construction of merging regions is illustrated in Fig. 7. Apart from the merging region construction rules, there are two main differences between BST/DME (Fig. 4) and DME(Fig. 3) . First, when two merging regions $mr(a)$, and $mr(b)$ cannot be merged with minimum merging cost $d(mr(a), mr(b))$, then the edge lengths $|e_a|$ and $|e_b|$ will be determined by construction rule BM5 in the bottom-up phase of BST/DME. Otherwise, the edge lengths will be determined in the top-down phase. Second, each node v can be embedded only at the location in joining segment $JS_Q(mr(v))$ that is closest to the location of its parent p, even if $|e_v| > d(JS_Q(mr(v)), l(p))$, where Q is the merging region of v's sibling node.

3.3 Correctness of BME Construction Rules

Let $v \in G$ have children a and b with merging regions $P = mr(a)$ and $Q = mr(b)$ respectively. Again, for convenience we let $L_a = JS_Q(P)$ and $L_b = JS_P(Q)$, and use R to indicate $SDR(L_a, L_b)$. If P and Q are both well-behaved, then we can prove the following properties of $R = SDR(L_a, L_b)$ after merging L_a and L_b.

FACT 2. *If joining segments L_a and L_b are parallel Manhattan arcs with constant max-delay and min-delay, then any Manhattan arc parallel to L_a (and L_b) has constant max-delay and min-delay (and thus constant skew).* □

LEMMA 1. *Any rectilinear line segment $l \subseteq SDR(L_a, L_b)$ after merging L_a and L_b is well-behaved.* □

Fact 2 follows directly from the discussion of well-behaved pathlength delay property. Similarly, Lemma 1 follows directly if we restrict L_a and L_b to be Manhattan arcs with constant max-delays and min-delays. Proofs of Lemma 1 (for the case when L_a and L_b are rectilinear line segments) and subsequent Lemmas are presented in Appendix A.1. Fact 2 and Lemma 1 are used to prove the correctness of BME construction rules for the case where the joining segments L_a and L_b are parallel Manhattan arcs, while Lemmas 2-5 below are used for the case where the joining segments L_a and L_b are parallel rectilinear segments. Without loss of generality, we assume that all the rectilinear joining segments L_a and L_b are vertical.

LEMMA 2. *Suppose L_a and L_b are vertical, and $l = \overline{pq} \subseteq SDR(L_a, L_b)$ is a horizontal line segment connecting $p \in L_a$ and $q \in L_b$ (Fig. 7(b)). If $skew\_const(l) \neq \emptyset$, then $skew\_const(l) \subseteq FMR(l)$.* □

LEMMA 3. *Suppose L_a and L_b are vertical. Let $R = SDR(L_a, L_b)$. (i) If $FMR(R) \neq \emptyset$, then the max-delay, min-delay, and skew values increase at constant rates $+1$, -1, and $+2$ respectively as we traverse from the boundaries of $FMR(R)$ horizontally to L_a or L_b (see Fig. 7(b)). (ii) If $FMR(R) = \emptyset$, then $MSR(R) = MSR(L_a)$ if $skew(MSR(L_a)) < skew(MSR(L_b))$, and $MSR(R) = MSR(L_b)$ otherwise (see Fig. 7(d)).* □

LEMMA 4. *If $FMR(R) = \emptyset$, then $mr(v)$ constructed by the BME construction rules (i) is $MSR(R)$, and (ii) is a well-behaved octilinear region (segment) with merging cost $|e_a| + |e_b| = d(L_a, L_b) + min\_skew(R) - B$, which is minimum subject to the constraint that $p \in L_a$ can merge with $q \in L_b$ only if $d(p,q) = d(L_a, L_b)$.* □

LEMMA 5. *If $FMR(R) \neq \emptyset$, then $mr(v)$ constructed by the BME construction rules (i) is equal to $FMR(R)$, and (ii) is a well-behaved octilinear region with minimum merging cost $= d(L_a, L_b)$.* □

The merging region of each sink node v is the sink location $l(v)$, and is thus well-behaved. Therefore, from the above Facts and Lemmas we have the following theorem.

THEOREM 1. *Under the pathlength delay model, for any node $v \in G$ the merging region $mr(v)$ computed by the BME construction rules (i) is consistent with the merging region definition, (ii) is a well-behaved octilinear region, and (iii) is computed in constant time.* □

3.4 Extension to Elmore Delay Model

We now extend the BME construction rules from pathlength to Elmore delay. Now, consider the points in $SDR(L_a, L_b)$ in Fig. 6(a) under Elmore delay. First, it is easy to see that under Elmore delay any segment $l \in SDR(L_a, L_b)$ that is parallel to L_a and L_b still has a constant max-delay, min-delay, and skew. Second, if l is a shortest path between L_a and L_b, then the skew over l is still a piecewise-linear function (but with different slopes), as shown in Fig. 8. To see this, consider again the boundary segments $l = \overline{ap} + \overline{pb}$ in Fig. 6(a). Again, to avoid confusion, the original max-delay and min-delay of points a and b are denoted as $max\_\bar{t}(a)$, $min\_\bar{t}(a)$, $max\_\bar{t}(b)$, and $min\_\bar{t}(b)$, respectively. The max-delay and min-delay values from point p via a to sinks in subtree T_a can be written as functions of $x = d(p, a)$, i.e., as

$$t_1(x) = Kx^2 + \alpha_1 x + max\_\bar{t}(a)$$

$$t_2(x) = Kx^2 + \alpha_1 x + min\_\bar{t}(a)$$

in Fig. 8, where $K = rc/2$ and $\alpha_1 = r \cdot Cap(a)$ (again r and c are resistance and capacitance per unit wirelength, and $Cap(a)$ is the the total capacitance of the subtree T_a rooted at node a). Similarly, we can write the max-delay and min-delay from point p via b to sinks in T_b as functions

$$t_3(x) = Kx^2 + \alpha_2 x + \beta_1$$

$$t_4(x) = Kx^2 + \alpha_2 x + \beta_2$$

also shown in Fig. 8. Here, $h = d(a,b)$ and $\alpha_2 = -r(hc + Cap(b))$, $\beta_1 = K \cdot h^2 + r \cdot h \cdot Cap(b) + max\_\bar{t}(b)$, and $\beta_2 = K \cdot h^2 + r \cdot h \cdot Cap(b) + min\_\bar{t}(b)$. We thus have

$$max\_t(p) = \max\{t_1(x), t_3(x)\} = \max\{\alpha_1 x + max\_\bar{t}(a), \alpha_2 x + \beta_1\} + K \cdot x^2$$

$$min\_t(p) = \min\{t_2(x), t_4(x)\} = \min\{\alpha_1 x + min\_\bar{t}(a), \alpha_2 x + \beta_2\} + K \cdot x^2.$$

Since $max\_t(p)$ and $min\_t(p)$ are each the sum of a piecewise-linear function of x and the same quadratic term $K \cdot x^2$, $skew(p)$ over segment $\overline{ab}$ is also a piecewise-linear function of x which divides $\overline{ab}$ into (at most) three contiguous intervals: $skew\_decr(l)$, $skew\_const(l)$, and $skew\_incr(l)$, where the skew changing rate are respectively $\alpha_2 - \alpha_1 = -r \cdot (Cap(a) + Cap(b) + h \cdot c)$, 0, and $\alpha_1 - \alpha_2 = r \cdot (Cap(a) + Cap(b) + h \cdot c)$.

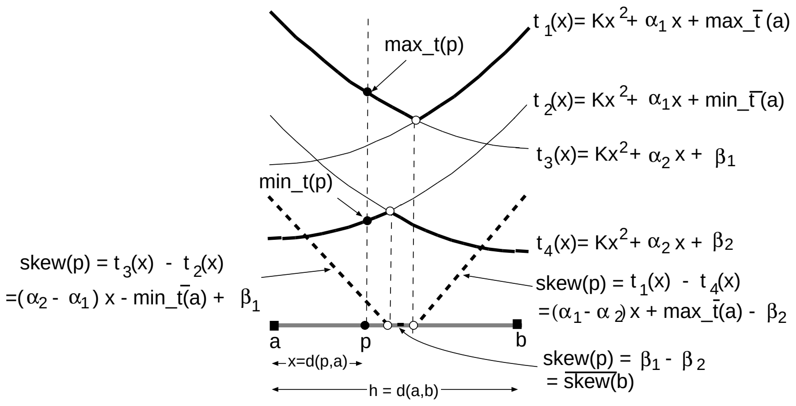


Fig. 8. Properties of Elmore delays and skew over boundary segments $l = \overline{ap} + \overline{pb}$, which is shown in Fig. 6.

Well-Behaved Elmore Delay Property. We now formalize this delay/skew property as follows. Given functions $f_1(x) = \max_{i=1,\cdots,n_1}\{\alpha_i \cdot x + \beta_i\}$ and $f_2(x) = \min_{i=1,\cdots,n_2}\{\alpha'_i \cdot x + \beta'_i\}$, a line segment $l = \overline{ab}$ is *well-behaved* if the max-delay and min-delay functions of point p on l are of the forms

$$max\_t(p) = f_1(x) + K \cdot x^2, \quad (5)$$

$$min\_t(p) = f_2(x) + K \cdot x^2, \quad (6)$$

where again $x = d(a,p)$. We say that $f_1(x)$ ($f_2(x)$) is an n_1-piecewise-linear *convex* (n_2-piecewise-linear *concave*) function since $f_1(x)$ ($f_2(x)$) has n_1 (n_2) linear regions with slopes strictly increasing (decreasing) from one endpoint of l to the other one, i.e., from a to b or vice versa (Fig. 9). Lemma 6 in Appendix A.2 shows that $skew(p)$ defined over l will be an n-piecewise-linear convex function, i.e.,

$$skew(p) = \max_{i=1,\cdots,n}\{\tilde{\alpha}_i \cdot x + \tilde{\beta}_i\}$$

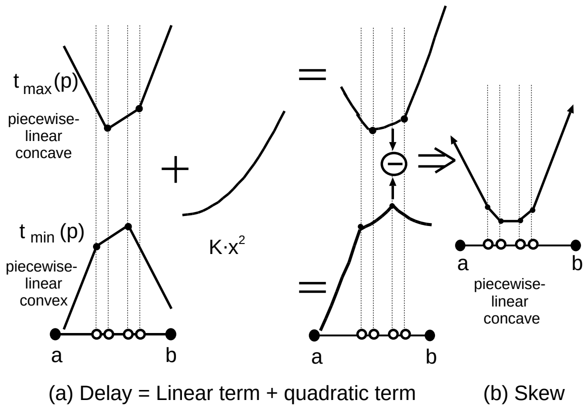


Fig. 9. (a) Delay and (b) skew functions for a well-behaved line segment. Skew turning points are indicated by the hollow circles on $\overline{ab}$.

such that (i) $n \leq n_1 + n_2 - 1$, and (ii) each skew turning point of l corresponds to a max-delay or min-delay turning point. Similar to the situation under the pathlength delay model, for any well-behaved line segment l, $FMR(l)$ and $MSR(l)$ are each single contiguous portions of l which can be computed in time linear in the number of skew turning points of l.

Well-Behaved Region Under Elmore delay, a convex polygonal region R is well-behaved if its boundary and interior segments are all well-behaved. Note that any boundary segment of a well-behaved region R which is a Manhattan arc does not necessarily have constant delays and skew. Also note that the delays of each boundary segment of R may have different quadratic terms.

3.5 Construction of the Merging Regions under Elmore Delay

The rules for constructing merging regions under Elmore delay are similar to those presented in Section 3. Assume that for each internal node v with children a and b, merging regions $P = mr(a)$ and $Q = mr(b)$ are well-behaved. Therefore, the shortest distance segments $L_a = SDS_Q(P)$ and $L_b = SDS_P(Q)$ must be well-behaved segments. However, $SDR(L_a, L_b)$ (and thus $mr(v)$) will not be well-behaved if either

Case I. L_a and L_b are parallel Manhattan arcs with non-constant delay functions, or

Case II. The delay functions on L_a and L_b have different quadratic terms.

In Case I, the delay value defined for a point in $SDR(L_a,L_b)$ will not be unique since the delays of each point in $SDR(L_a, L_b)$ can be defined by many pairs of points (p, q) with $p \in L_a$ and $q \in L_b$, as long as $d(p, q) = d(L_a, L_b)$. For Case II, Lemma 7 in Appendix A.2 shows that the line segments within $SDR(L_a,L_b)$ will not be well-behaved. To guarantee well-behaved regions under Elmore delay, the computation of the joining segments $JS_Q(P)$ and $JS_P(Q)$ (rule BM1) is modified as follows.

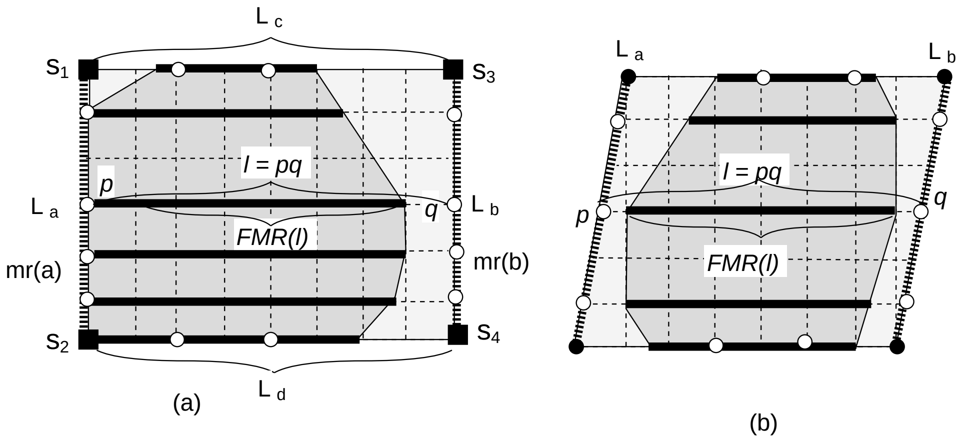


Fig. 10. Example of merging region construction when the joining segments of children merging regions are parallel segments which (a) are vertical or (b) have slopes < -1 or $> +1$. In (a) is an example with $n = 4$ sinks $s_1, \cdots, s_4$, where there can be up to n skew turning points on each of L_a and L_b. Note that each turning point corresponds to a possible vertex of $mr(v)$.

BM1. Compute $JS_Q(P) = SDS_Q(P)$ and $JS_P(Q) = SDS_P(Q)$. If either Case I or Case II holds, then $JS_Q(P)$ and $JS_P(Q)$ are chosen to be (arbitrary) single points on $SDS_Q(P)$ and $SDS_P(Q)$, respectively.

Since the joining segments can be parallel segments with any slope, (i.e., other than Manhattan arcs and rectilinear line segments; see Fig. 10), rule BM3 is modified as follows.

BM3. If L_a and L_b are parallel segments with slope < -1 or $> +1$ (between $+1$ and -1), compute $FMR(l)$ for each horizontal (vertical) line segment $l = \overline{pq}$ such that $p \in L_a$, $q \in L_b$, and either p or q is a skew turning point of L_a or L_b (see Fig. 10).

Finally, the calculation of detour wiring (rule BM5) must be modified under Elmore delay as follows:

BM5. If $F = \emptyset$, then the computation of $mr(v)$ is the same as before, but the computation of minimum detour wiring is modified as follows. Let $l_1 = MSR(L_a)$ and $l_2 = MSR(L_b)$. If $skew(l_1) \leq skew(l_2)$, then we set $|e_a| = 0$ and $|e_b| = x$, where x satisfies the equation $rx(cx/2 + Cap(b)) + min\_\bar{t}(p_2) = max\_\bar{t}(p_1) - B$, with (i) $p_1 \in l_1$, (ii) $p_2 \in l_2$, (iii) $\overline{p_1 p_2}$ is horizontal (vertical) if L_a and L_b are vertical (horizontal), and (iv) again $min\_\bar{t}$ and $max\_\bar{t}$ are the original delays of l_1 and l_2 before $mr(a)$ and $mr(b)$ are merged. Similarly, if $skew(l_1) > skew(l_2)$, we set $|e_a| = x$ and $|e_b| = 0$, where x satisfies the equation $rx(cx/2 + Cap(a)) + min\_\bar{t}(p_1) = max\_\bar{t}(p_2) - B$. The new delays of points $p \in mr(v)$ are computed as $min\_t(p) = max\_\bar{t}(p) - B$ and $max\_t(p) = max\_\bar{t}(p)$.

Similar to the proof of Theorem 1, we can show that the merging region $mr(v)$ computed by the revised BME construction rules (i) satisfies the merging region definition, and (ii) is a well-behaved region. Lemma 9 in Appendix A.2 shows that there can be up to n skew turning points on each of L_a or L_b, where n is the number of leaf nodes in the subtree T_v rooted at v. Each skew turning point corresponds to a possible vertex of $mr(v)$, from which it easily follows that (iii) $mr(v)$ has at most $2n+4$ sides and can be computed in $O(n)$ time. We thus have the following theorem, whose detailed proof is given in Appendix A.2.

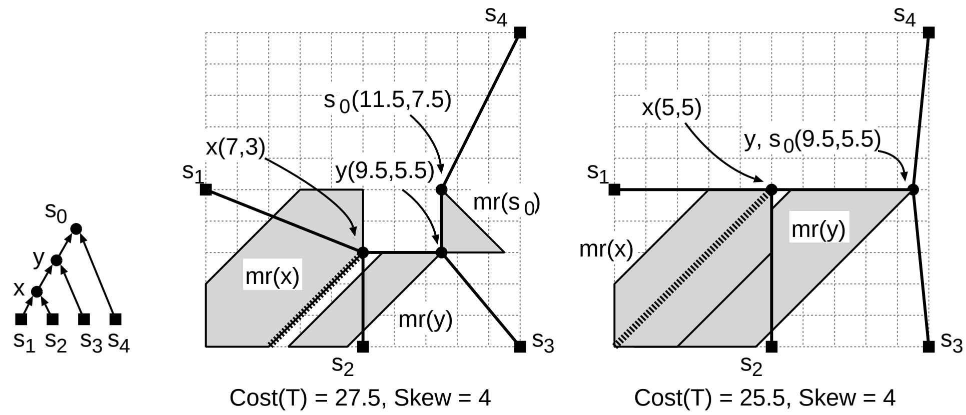
THEOREM 2. *For any node $v \in G$ the merging region $mr(v)$ computed by the BME construction rules under Elmore delay (i) is consistent with the merging region definition, (ii) is a well-behaved region with at most $2n+4$ sides, and (iii) can be computed in $O(n)$ time where n is the number of leaf nodes in T_v.* □

Note that in Fig. 10(a), the number of skew turning points on the boundaries of the merging region of v's ancestor nodes will continuously increase only if (i) $mr(v)$ has a boundary segment l such that $l \subseteq L_a$ or $l \subseteq L_b$, and (ii) l is used as the joining segment in consecutive merging steps. However, such conditions will hardly ever hold for a few consecutive merging steps. Since each skew turning points on the joining segments corresponds to at most one vertex of the resulting merging region, in practice each merging region still has a constant number of boundary segments and can be computed in constant time. Indeed, in all our experiments, no merging region has ever had more than 9 sides. It is unlikely that two parallel joining segments have exactly the "same" slopes unless they are rectilinear segments or Manhattan arcs. So, in practice the case in Fig. 10(b) rarely happens.

3.6 Optimality of BST/DME

Note that when merging regions P and Q overlap (e.g., consider the pair of polygons C and D (or E) in Fig. 5), or when either $L_a = JS_Q(P) \neq SDS_Q(P)$ or $L_b = JS_P(Q) \neq SDS_P(Q)$, then $SDR(L_a, L_b)$ is not equal to the set of points which has minimum sum of distances to P and Q. Thus, $mr(v)$ will not contain all points that have minimum merging cost. Also, when $FMR(SDR(L_a, L_b)) = \emptyset$ (i.e., detour wiring is needed), then the merging cost of $mr(v)$ is not necessarily the minimum. An example is given in our technical report [Cong et al. 1995b]. Therefore, for node $v \in G$ with children a and b which have merging regions $P = mr(a)$ and $Q = mr(b)$ respectively, if (i) P and Q do not overlap, (ii) $JS_Q(P) = SDS_Q(P)$ and $JS_P(Q) = SDS_P(Q)$, and (iii) no detour wiring is needed, then $mr(v)$ will contain all points having minimum merging cost. In all our experiments, the above condition holds for most nodes in a good routing topology (see the discussions in Section 6). In particular, in the zero-skew case (ii) is always true, and the other two conditions holds for most nodes in a good routing topology. So the performance of BST/DME for zero-skew bound will be very close to that of DME for the given topology.

Even if all the merging regions are equal to the full set of minimum-cost merging points, our method is still not optimal for the given topology. A four-sink counter-example is given in Fig. 11, and finding an optimal BST solution for a prescribed topology is still open. Nevertheless, when combined with the topology generation method described in Section 5, BST/DME not only closely matches the best known heuristics for both the zero-skew and infinite-skew limiting cases, but also provides a smooth skew-cost tradeoff over all intermediate values of B.



(a) Topology (b) Boundary Merging and Embedding (c) Interior Merging and Embedding

Fig. 11. An example of routing 4 sinks (filled squares) with a skew bound of 4 units. Each internal node (filled circle) is embedded in its merging region (shaded region). Each pair of coordinates associated with a point or a segment represents its max-delay and min-delay. For a fixed topology in (a), (b) the routing cost is 27.5 units if only boundary segments are considered for merging, and (c) we can reduce merging cost by merging interior points (dashed line in $mr(x)$).

4. THE INTERIOR MERGING AND EMBEDDING METHOD

As outlined in the previous section, the construction of a merging region is based on the nearest *boundary* segments of its children's merging regions: no interior point of the child merging regions is used to construct the parent merging region. However, such an approach produces a sub-optimal merging cost when detour occurs. Furthermore, even if no detour is required, it is not always advisable to use only boundary segments for merging. We observe that for any merging region constructed by BME, a point along the boundary is likely to have skew very close to B. In other words, by merging nearest boundary segments, BME method tends to fully utilize the available skew resource at the bottom level of the routing tree as in Fig. 11(b) and this may result in a smaller merging region at a higher level. Instead, we can conserve the skew resource by merging interior points, which may result in a larger merging region at a parent node and possibly reduce the total merging cost (Fig. 11(c)). Note that merging region $mr(y)$ in Fig. 11(c) overlaps with $mr(x)$, has a larger area, and is closer to s_4 when compared to $mr(y)$ in Fig. 11(b).

Given the above considerations, merging of interior points of the merging regions has strong potential to reduce total wirelength. However, merging interior points may cause ambiguity in the delay functions for a point p in the new merging region: p may correspond to the merging of infinitely many pairs of interior points from its child merging regions, and may therefore have different max-delay and min-delay values. Since max-delay and min-delay information is required to construct merging regions at a higher level, this ambiguity (which is avoided when only nearest boundary segments are considered as in the BME method) causes difficulty in the merging process. To overcome ambiguity and yet exploit the interiors of merging regions, we propose the *Interior Merging and Embedding* (IME) algorithm which

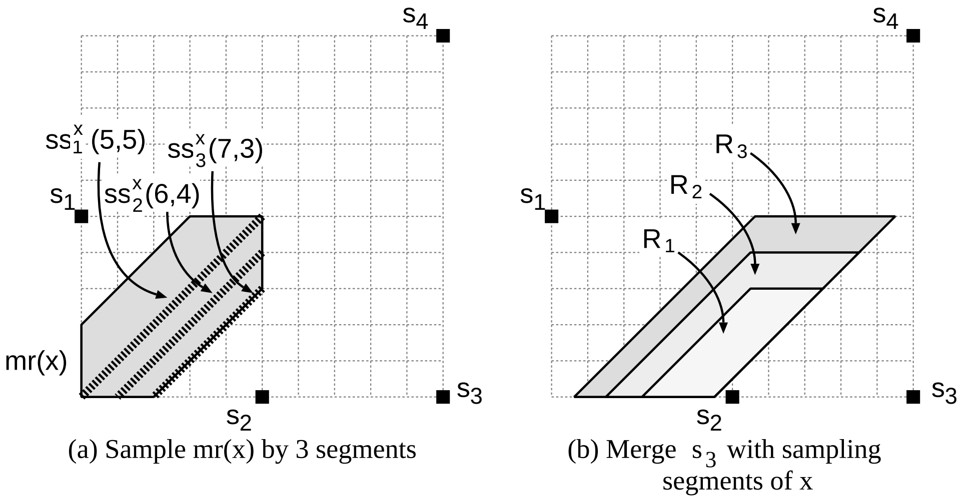


Fig. 12. Interior merging using Manhattan arcs. For the example given in Fig. 11, (a) the merging region $mr(x)$ is sampled by three Manhattan arcs $\{ss_1^x, ss_2^x, ss_3^x\}$. The pair coordinates for each sampling segment denote the max-delay and min-delay of the sampling segment. (b) Merging these sampling segments with sink s_3 produces three merging regions where R_i is produced by merging s_3 with ss_i^x.

employs a sampling strategy and a dynamic programming-based selection technique to consider merging points which are interior to, rather than on the boundary of, the merging region.

4.1 Overview of IME Method

As the BME method, the IME method is concerned with constructing of merging regions. In other words, the template for the BST/DME algorithm given in Fig. 4(a) still applies except that we replace the BME construction rules by the IME construction rules (to be given below) in the statement "Calculate $mr(v)$ by BME construction rules." The key difference between the two methods is that in IME, each node v in the topology G is associated with a set $\mathcal{R}_v = \{R_1^v, R_2^v, \cdots, R_k^v\}$ of merging regions (if v is understood, we simply use $\mathcal{R}$ and R_i) whereas a node in BME is associated with only one merging region. The *interior points* in two sets of merging regions can be used to construct merging regions of their parent node, via the use of *sampling*. Each merging region R_i is sampled by a set of well-behaved line segments (or *sampling segments*, denoted ss) in R_i. For example, under the pathlength delay model, we use only Manhattan arcs with constant min-delays and max-delays, and rectilinear line segments as sampling segments since these segments are well-behaved (Fact 2 and Lemma 1). On the other hand, any line segment in a well-behaved region under Elmore delay is well-behaved (Lemma 7) and can be used as a sampling segment. We denote the set of sampling segments (or *sampling set*) of R_i by $SS(R_i)$. The sampling set of $\mathcal{R}$, denoted $SS(\mathcal{R})$, is $\bigcup_{i=1}^k SS(R_i)$.

Consider the merging of two nodes a and b in G. Let $\mathcal{R}_a$ and $\mathcal{R}_b$ be the set of merging regions associated with a and b, respectively. The parent, say v, of a and b in G has as many as $|SS(\mathcal{R}_a)| \times |SS(\mathcal{R}_b)|$ possible merging regions due to the merging of each sampling segment ss_a in $SS(\mathcal{R}_a)$ with each sampling segment ss_b in $SS(\mathcal{R}_b)$. Given two sampling segments ss_a and ss_b, we apply the BME construction rules BM1–BM5 as outlined before to construct a merging region. Note that all points on ss_a and ss_b are feasible embedding point for nodes a and b, respectively. Therefore, we can treat ss_a and ss_b as a "merging region" of a and b, respectively, and apply the BME construction rules to construct a merging region of v since all sampling segments chosen are well-behaved. Consider the same example given in Fig. 11, we sample $mr(x)$ by three sampling segments $\{ss_1^x, ss_2^x, ss_3^x\}$ as shown in Fig. 12. We then merge each sampling segment with sink s_3 and construct three merging regions $\{R_1, R_2, R_3\}$ for internal node y where R_i is the result of merging ss_i^x with s_3.

This example also illustrates the difficulty of merging interior points as mentioned in the beginning of this section. Region R_1 is contained within R_2 and R_3. If we pick any point in R_1, then this point is also contained in R_2 and R_3 and therefore have different max-delay and min-delay functions and hence, different skew functions. If we consider all interior points of $mr(x)$ for merging with s_3, then any point in R_1 has infinitely many different delay and skew functions. Generally speaking, given a merging region, there is an infinite number of choices of sampling segments. Furthermore, even if we select only a constant number of sampling segments for each region, the size of the overall number of merging regions may grow exponentially during our bottom-up construction of merging regions. In particular, even if each region is sampled by no more than s sampling segments, the number of merging regions at the root of the routing topology is $O(s^n)$, where n is the total number of sinks. To achieve an efficient implementation, we limit the number of merging regions of an internal node by a constant, say k. Each region is in turn sampled by exactly s sampling segments when the region is being merged with other regions of the sibling node. When we merge two sampling segment set, each with $\leq k \cdot s$ sampling segments, k^2s^2 merging regions are generated for the parent node. A key step in the IME method lies in selecting the "best" k merging regions for the new parent node. A simple *greedy* selection strategy is to choose k merging regions with the smallest total capacitances [Cong et al. 1996a]. We also present a dynamic programming-based selection method to compute the k "best" possible merging regions for the new node.

Before we present the dynamic programming-based selection method, we describe the IME Construction Rules for the IME method. In the following, we assume that the sets of merging regions $\mathcal{R}_a$ and $\mathcal{R}_b$ of children a and b, respectively, are given, and we want to compute the merging regions $\mathcal{R}_v$ for the parent internal v. The IME construction rules are as follows:

IM1. Compute the sampling segment set of node a, denoted $SS(R_a)$, by sampling each R_i^a in $\mathcal{R}_a$ using s well-behaved line segments in R_i^a. Similarly, compute $SS(R_b)$ of node b.

IM2. For each sampling segment ss_a from $SS(\mathcal{R}_a)$ and each sampling segment ss_b from $SS(\mathcal{R}_b)$, apply BME construction rules BM1–BM5 to construct a new merging region $mr(v)$ of v from ss_a and ss_b and put $mr(v)$ in $\mathcal{R}_v$.

IM3. Select from $\mathcal{R}_v$ k merging regions using either the greedy selection technique or the dynamic programming-based selection technique (to be described in details in the next subsection.)

Note that in rule IM2, we treat ss_a and ss_b as "merging regions" of a and b, respectively. To apply the BME construction rules, we first find joining segments of ss_a and ss_b and then perform merging of $JS(ss_a)$ with $JS(ss_b)$. The only procedure of the IME construction rules left to be explained is the dynamic programming-based selection technique to be described in the next subsections.

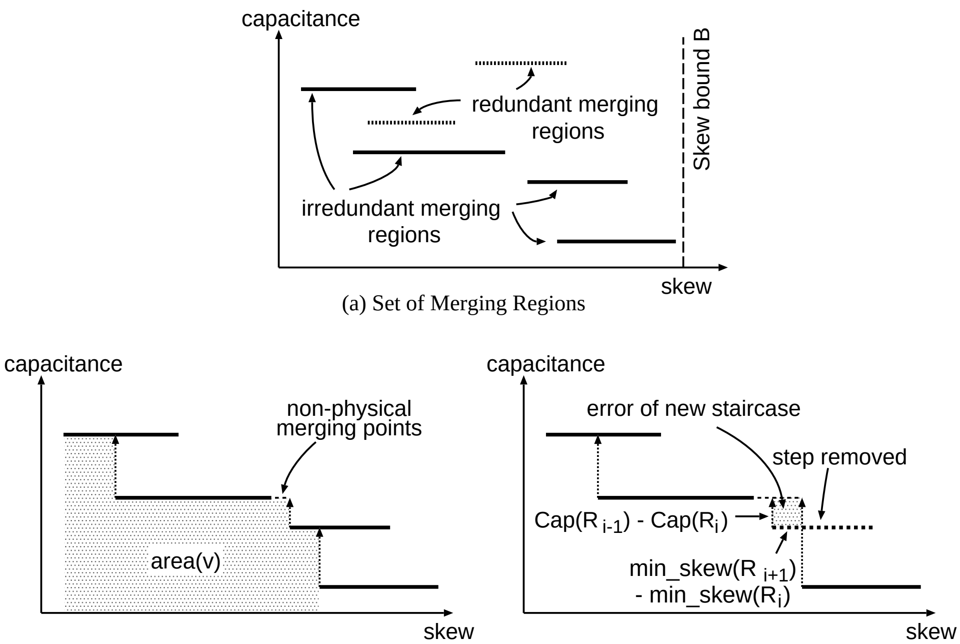
4.2 Dynamic Programming-Based Selection Technique

In what follows, we require the following terminology. A merging region R is associated with three values: (i) $Cap(R)$, the total capacitance rooted at region R,[2] (ii) $min\_skew(R)$, and (iii) $max\_skew(R)$, the maximum skew possible within the merging region. Recall that merging regions in IME are still constructed by the BME construction rules. Consider merging of children a and b of node v. We construct a merging region of v by merging two sampling segments, say L_a and L_b, of $mr(a)$ and $mr(b)$, respectively with a merging cost of $|e_a|+|e_b|$. From the result in the previous section, we note that the resultant merging region of v, denoted $mr(v)$, has a capacitance of $Cap(mr(v)) = Cap(mr(a)) + Cap(mr(b)) + (|e_a| + |e_b|) \cdot c$, which is constant for all points in $mr(v)$. Also note that $max\_skew(R)$ is kept within the skew bound B by the BME construction rules. If we plot a graph with the horizontal axis representing the skew and the vertical axis representing the capacitance, then each merging region R_i of node v is a horizontal line segment with y-coordinate $Cap(R_i)$ and x-coordinates $min\_skew(R_i)$ and $max\_skew(R_i)$ for the left and right endpoints, respectively. Points within the merging region R_i is mapped many-to-one to the horizontal line segment representing R_i.

Consider a node v in G associated with a set of more than k merging regions after merging its two children. Then, a merging region R of v is *"redundant"* if and only if there exists another merging region R' of v such that $min\_skew(R') < min\_skew(R)$ and $Cap(R') < Cap(R)$ (See Fig. 13(a)). Let $IMR(v) = \{R_1, R_2, \cdots, R_m\}$ denote the set of irredundant merging regions of v with R_i's arranged in descending order of $Cap(R_i)$, then, for all i with $1 \leq i < m$, $min\_skew(R_i) < min\_skew(R_{i+1})$.

The set of irredundant merging regions forms a *staircase* with $m-1$ steps as shown in Fig. 13. By creating a *step* from a height of $Cap(R_i)$ to $Cap(R_{i-1})$ at a x-coordinate of $min\_skew(R_i)$ and then from $min\_skew(R_i)$ to $min\_skew(R_{i-1})$ at a y-coordinate of $Cap(R_{i-1})$ for all i with $1 < 1 \leq m$, we have a $m-1$ step staircase starting at $min\_skew(R_m)$ as shown in Fig. 13(c). Note that the pair of coordinates $(min\_skew(R_i), Cap(R_{i-1}))$ may fall outside the range of the horizontal line segment representing R_{i-1}. In other words, some horizontal line segments in

[2]We use $Cap(v)$ to denote the total capacitance rooted at node v in the previous section. The slight change in the notation is due to the fact in IME, we have multiple merging regions for a node.



(a) Set of Merging Regions

(b) Irredundant regions form a staircase

(c) Staircase with a step removed

Fig. 13. (a) Set of merging regions. (b) Set of irredundant merging regions form a staircase. (c) Removing an intermediate step results in a new staircase with an error depicted by the shaded region.

the staircase might not correspond to physical points in the m merging regions of v. For example, the dashed-line horizontal line segment in Fig. 13(b) corresponds to non-physical merging points.

The area of the staircase of a set of merging regions of node v, denoted $area(v)$, is defined to be the area under the staircase between the skews $min\_skew(R_1)$ and $min\_skew(R_k)$:

$$area(v) \; = \; \sum_{i=1}^{m-1} \{min\_skew(R_{i+1}) - min\_skew(R_i)\} \times Cap(R_i)$$

If we *remove* one of the intermediate steps, say R_i $(1 < i < m)$, we obtain a $(m-2)$-step staircase which approximates the original staircase with an error of $(min\_skew(R_{i+1}) - min\_skew(R_i)) \times (Cap(R_{i-1}) - Cap(R_i))$ (Fig. 13(d)). Therefore, in order to retain a good spectrum of no more than k merging regions at each step, we propose to solve the following problem:

The Optimal (m, k)**-Sampling Problem:** Given a set of m irredundant merging regions, $IMR = \{R_1, \cdots, R_m\}$, find a subset of k $(2 \leq k \leq m)$ merging regions such that after removing each of the $m - k$ intermediate merging regions, the total error of the resulting staircase $IMR' = \{R_{\pi(1)} = R_1, R_{\pi(2)}, \cdots, R_{\pi(k-1)}, R_{\pi(k)} =$

$R_m\} \subset IMR$ is minimum (or equivalently, $area(IMR') - area(IMR)$ is minimal), where $\pi : \{1 \cdots k\} \rightarrow \{1 \cdots m\}$ is a strictly monotonically increasing function.

Note that both R_1 and R_m are retained in IMR' where R_1 is the merging region with the smallest min-skew and R_m is the merging region with the smallest total capacitance. The motivation for retaining R_1 and R_m in IMR' is that when considered for merging at the next level, R_1 tends to produce larger parent merging regions whereas R_m tends to produce lower-cost parent merging regions.

4.3 Optimal Solution to the (m, k)-Sampling Problem

We developed an optimal algorithm to the (m, k)-sampling problem based on the dynamic programming approach similar to the algorithm used in [Wang and Wong 1992] for floorplan construction. Effectively, we compute an optimal (m', k')-sampling solution $S_i[m', k']$ for each $2 \leq m' \leq m$, $2 \leq k' \leq k$ and $1 \leq i \leq m - m' + 1$ by choosing the best k'-sampling from m' merging regions $\{R_i, R_{i+1}, \cdots, R_{i+m'-1}\}$ under the condition that R_i and $R_{i+m'-1}$ are in the k'-sampling. Let $err_i[m', k']$ be the minimum error for the optimal (m', k')-sampling solution $S_i[m', k']$. We can show that

THEOREM 3. *For each $2 \leq m' \leq m$, $2 \leq k' \leq k \leq m$ and $1 \leq i \leq m - m' + 1$, the minimum error $err_i[m', k']$ for the optimal (m', k')-sampling solution is:*

$$err_i[m', k'] = \begin{cases} 0 & \text{if } m' = k' \\ \{(min\_skew(R_{i+m'-1}) - min\_skew(R_{i+m'-2})) \times & \text{if } m' > k' \\ \quad (Cap(R_i) - Cap(R_{i+m'-2}))\} + err_i[m' - 1, k'] & \text{and } k' = 2 \\ \min_{i < i' \leq m' - k' + i + 1} \{err_i[i' - i + 1, 2] + & \text{if } m' > k' \\ \qquad err_{i'}[m' - i' + i, k' - 1]\} & \text{and } k' > 2 \end{cases} \tag{7}$$

A straightforward implementation of the above computation gives an $O(k \cdot m^3)$-time optimal (m, k)-sampling algorithm. After careful pruning of the solution space, we can achieve a better time complexity. First, note that for case (i) and (ii) of Eqn. (7), the solution is straightforward and we can compute all $S_i[m', k']$ and $err_i[m', k']$ where $1 \leq i \leq m - m' + 1$, and m' and k' satisfy the conditions stated in cases (i) and (ii) of Eqn. (7) in $O(m^2)$-time.

In the following, we assume $m > k > 2$. We are interested in obtaining the optimal solution $S_1[m, k]$. To determine the index of the region after R_1 in the optimal solution, we compute $err_1[m, k]$ with $err_1[i, 2]$ and $err_i[m - i + 1, k - 1]$ for $1 < i \leq m - k + 2$. Assuming that $k - 1 > 2$, we again apply case (iii) of Eqn. (7) to solve for $err_i[m - i + 1, k - 1]$ using $err_i[i' - i + 1, 2]$ and $err_{i'}[m - i' + 1, k - 2]$ for $i < i' \leq m - k + 2$. Continuing this recursion, we observe a pattern shared by the errors $err_1[m, k]$, $err_i[m - i + 1, k - 1]$, $err_{i'}[m - i' + 1, k - 2]$, and so on. If we denote these errors generically by $err_i[m', k']$, then we observe that $m' = m - i + 1$ for all cases. Therefore, we do not have to compute $err_i[m', k']$ for every valid combination of i, m' and k', each in its respective range ($2 \leq m' \leq m$, $2 \leq k' \leq k$ and $1 \leq i \leq m - m' + 1$). Instead, *only* errors $err_i[m', k']$ with $m' = m - i + 1$ are required.

```
Procedure Optimal_(m,k)-Sampling_Algorithm(IMR)
Input: Irredundant merging regions IMR = {R_1, ..., R_m}
Output: Best "k" merging regions IMR' = {R_1 = R_π(1), R_π(2), ..., R_m = R_π(k)}
        where π : {1 ... k} → {1 ... m} is strictly monotonically increasing.

for each k', i s.t. 2 ≤ k' ≤ k, 1 ≤ i ≤ m − k' + 1
  err_i[k', k'] ← 0;
  next_i[k', k'] ← i + 1;

for each m', i s.t. 2 ≤ m' ≤ m, 1 ≤ i ≤ m − m' + 1
  err_i[m', 2] ← (min_skew(R_{i+m'−1}) − min_skew(R_{i+m'−2}))×
                 (Cap(R_i) − Cap(R_{i+m'−2})) + err_i[m' − 1, 2];
  next_i[m', 2] ← i + m' − 1;

for each k' s.t. 2 < k' ≤ k
  for each i s.t. k − k' + 1 ≤ i ≤ m − k' + 1
    i' ← i + 1;
    err_i[m − i + 1, k'] ← err_i[i' − i + 1, 2] + err'_i[m − i' + 1, k' − 1];
    next_i[m − i + 1, k'] ← i';
    for each i' s.t. i + 1 < i' ≤ m − k' + 1
      if err_i[i' − i + 1, 2] + err_{i'}[m − i' + 1, k' − 1] < err_i[m − i + 1, k'],
        err_i[m − i + 1, k'] ← err_i[i' − i + 1, 2] + err_{i'}[m − i' + i, k' − 1];
        next_i[m − i + 1, k'] ← i';

i ← 1;
m' ← m;
S_1[m, k] ← {R_i};
for k' ← k downto 2 do
  i ← next_i[m', k'];
  m' ← m' − i + 1;
  S_1[m, k] ← S_1[m, k] ∪ {R_i};
```

Fig. 14. The optimal (m, k)-Sampling Algorithm. Note that $m \geq k$.

The *Optimal (m, k)-Sampling Algorithm* is given in Fig. 14. In the algorithm, the matrix $next_i[m', k']$ records the index of the merging region immediately after R_i in the optimal subset (of size k') of the set $\{R_i, R_{i+1}, \cdots, R_{i+m'-1}\}$. We first initialize $err_i[m', k']$ and $next_i[m', k']$ for the first two cases of Eqn. (7). Then, for each k' and i such that $2 < k' \leq k$ and $k - k' + 1 \leq i \leq m - k' + 1$, we apply case (iii) of Eqn. (7) to compute the minimum error $err_i[m - i + 1, k']$. Therefore, we have the following result:

THEOREM 4. *The time complexity of the optimal (m, k)-Sampling Algorithm is $O(k \cdot m^2)$.* □

We can observe that the most expensive operation in the merging process is due to the optimal (m, k)-sampling algorithm which is polynomial in terms of $m \leq k^2 \cdot s^2$. Since m is a constant, the merging process can be performed in constant time and the time complexities of BME and IME are still in the same order. In our experiments, the number of irredundant regions m is much lower than $k^2 \cdot s^2$. So the run-time ratios of IME over BME can be only dominated by the number of total regions generated during merging, i.e., $k^2 \cdot s^2$.

To summarize, if we were to apply the dynamic programming-based selection technique in the IME construction, we replace the IME construction rule IM3 by the following rules:

IM3.a. Sort merging regions of v in $\mathcal{R}_v$ in descending order of their capacitance and scan the sorted $\mathcal{R}_v$ to remove "redundant" merging regions.

IM3.b. Apply Optimal (m, k)-Sampling Algorithm on the set of irredundant merging regions of v if necessary.

5. TOPOLOGY GENERATION FOR BOUNDED-SKEW ROUTING

We now consider the variant BST formulation where the topology is not fixed and can be determined dynamically. Our topology generation method for bounded-skew routing is an extension of [Edahiro 1992; Edahiro 1993a] that exploits flexibility stemming from allowed skew during the topology construction. With this new topology generation method, not only can our BST/DME algorithm provide a smooth cost-skew tradeoff, but also very closely match the performance of the best-known heuristics for both the zero-skew [Edahiro 1993a; Edahiro 1994] and infinite-skew limiting cases [Borah et al. 1994; Kahng and Robins 1992]. Note the latter case corresponds to the Steiner minimal tree problem.

Recall that in DME, two merging subtrees are always merged at their roots so as to maintain zero skew. However, the shortest connection between two trees may not be between their roots. Indeed, subtrees may be merged at non-root nodes as long as the resulting skew is $\leq B$. This flexibility allows reduced merging cost and is the key merit of the Greedy-BST/DME approach.

Consider the example in Fig. 15, where the eight sinks are equally spaced on a horizontal line. When B is near zero, the minimum tree cost can be obtained by merging subtrees T_a and T_b at their roots a and b. However, this topology is bad when B is large, even if the costs of the two subtrees can be minimum. When the skew bound is large, ideally one should adjust the subtree topology so that the roots of subtrees become closer while the subtree costs remain the same or increase slightly. Suppose the skew bound B is large enough such that the least cost BST connecting the eight sinks is the straight line from sink 1 to sink 8. Although subtrees T_a and T_b in Fig. 15 can be optimally embedded, T_r obtained by merging T_a and T_b is not optimal since the smallest distance between T_a and T_b is the distance between sinks 4 and 5. However, if we reposition a on edge e_4 and b on edge e_5 to obtain new topologies T'_a and T'_b, respectively, then subtree roots a and b become now closer, while the minimum routing costs of T'_a and T'_b is the same as that of T_a and T_b. We can obtain the least cost BST T_r by merging T'_a and T'_b.

Fig. 16 illustrates in more details how we adjusts the tree topology T_r. First, the root r is moved down to some tree edge, say $e_u = \overline{uu'}$, so that the root becomes the parent of nodes u and u'. Then the tree topology is adjusted accordingly by adding edges $\overline{e_b}$, e_u, and $e_{u'}$, deleting edges e_a, e_b, and e_u, and redirecting all the edges on the unique path from node u' to node a in the original tree T_r. Note that when we shift the root of the tree in this way, only a few tree edges will be removed or added so that the basic structure of the subtrees remains the same. In

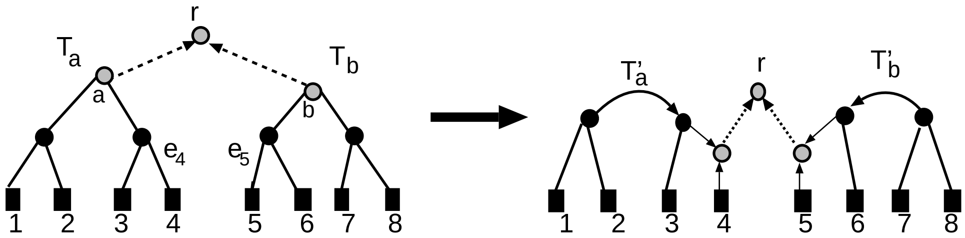


Fig. 15. An example showing that given skew bound $B \gg 0$, changing the subtree topology before merging will reduce the merging cost.

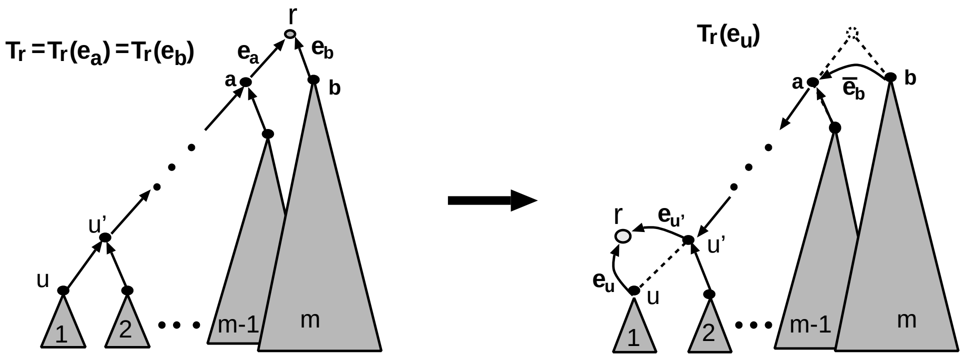


Fig. 16. Repositioning the root in changing the topology.

practice, the costs of the two subtrees will have little increase when the topologies are changed this way. We refer to the new tree topology as $T_r(e_u)$. Under this definition, $T_r = T_r(e_a) = T_r(e_b)$ where a and b are children of r. Let $|T_r|$ denote the number of nodes in T_r. Then, there are $|T_r| - 1$ edges in T_r and therefore, $|T_r| - 2$ alternative locations of r (since $T_r = T_r(e_a) = T_r(e_b)$) if we allow r to be placed at any level. In our Greedy-BST/DME algorithm, we consider alternative locations of tree roots when two subtrees are merged. To exploit this flexibility due to non-zero skew bound, we extend the idea of nearest neighbor graph of the Greedy-DME algorithm by considering the repositioning of tree roots.

The Greedy-BST/DME algorithm follows the Greedy-DME structure, as shown in Fig. 17. One key difference between Greedy-BST/DME and Greedy-DME is in the construction of the nearest-neighbor graph H. Recall that in Greedy-DME, each node v in H corresponds to merging segment $ms(v)$ of a root node v of a subtree; and each edge E_{uv} of H represents the merging of two subtrees, T_u and T_v, which are rooted at u and v. The weight of edge E_{uv}, denoted $|E_{uv}|$, represents the merging cost of T_u and T_v. In our Greedy-BST/DME algorithm, we still have a forest of subtrees, $\mathcal{F}$. Each node v in H corresponds to a node in $\mathcal{F}$ (which is not necessarily the tree root as in the Greedy-DME algorithm). Therefore, the number of nodes in H is equal to the number of nodes in $\mathcal{F}$, which is between the number of sinks n and $2n$.

Algorithm Greedy-BST/DME(S, B)

Input: Set of sinks S; skew bound B; parameter f

Output: BST $\mathbf{T(S)}$ with skew $\leq B$

$n \leftarrow |S|$ /* starting with n subtrees */
for all sinks $v \in S$
  $cost'(v) \leftarrow 0$
  $mr'(v) \leftarrow mr(v) \leftarrow \{l(v)\}$
while $(n > 1)$
  Construct nearest-neighbor graph H in $O(n^2)$ time
  $A \leftarrow$ sorted edges of H in non-decreasing order of
    edge weight in $O(n \lg n)$ time
  for $i = 1$ to $\min\{max\{1, n/f\}, n-1\}$ do
    Take edge E_{uv} with smallest weight from A
    Delete all edges incident to u or v from A
    Let T_r (T_s) be the subtree containing node u (v)
    if $u \neq r$ and u not a child of node r
      $T_r(e_u) \leftarrow$ relocate root r to e_u
      and adjust tree topology (see Fig. 16)
    if $v \neq s$ and u not a child of node s
      $T_s(e_v) \leftarrow$ relocate root s to e_v
      and adjust tree topology (see Fig. 16)
    $T \leftarrow$ merge $T_r(e_u)$ and $T_s(e_v)$ in $O(|T|)$ time
    Update $mr'(w)$, $cost'(w)$ $\forall w \in T$ in $O(|T|)$ time
    $n \leftarrow n - 1$ /* one less subtree */
$\mathbf{T(S)} \leftarrow$ Find_Exact_Placements(T) (see Fig. 4(b))

Fig. 17. The Greedy-BST/DME Algorithm.

Consider two nodes u and v in H. Let T_r be the subtree in $\mathcal{F}$ containing u and T_s be the subtree containing v. Note that u could be the root r and v could be the root s. Then each edge E_{uv} of H represents the merging of two subtrees $T_r(e_u)$ and $T_s(e_v)$, and $|E_{uv}|$ represents the wirelength increase by merging $T_r(e_u)$ and $T_s(e_v)$. We define $|E_{uv}|$ as follows. If nodes u and v are in the same tree, then $|E_{uv}| = \infty$ (same trees cannot be merged). Otherwise, we first construct the new subtrees $T_r(e_u)$ and $T_s(e_v)$, and then merge $T_r(e_u)$ and $T_s(e_v)$ into a new tree, say T_t. Then $|E_{uv}| = cost(T_t) - cost(T_r) - cost(T_s)$.

By maintaining two more variables $mr'(w)$ and $cost'(w)$ for each node $w \in H$ we can still compute $|E_{uv}|$ in constant time. The definitions of $mr'(v)$ and $cost'(v)$ for node $v \in H$ are as follows. Again, let T_s be the subtree containing v and $T_s(e_v)$ be the resulting adjusted tree after the root is relocated to edge e_v. Then $mr'(v)$ is the merging region of tree root s of $T_s(e_v)$ and $cost'(v) = cost(T_s(e_v))$. Thus, for any edge E_{uv}, we can compute $|E_{uv}|$ in constant time if $mr'(u)$, $mr'(v)$, $cost'(u)$ and $cost'(v)$ are known (i.e., $|E_{uv}| = cost(T) - cost(T_r) - cost(T_s) = |e_r| + |e_s| + cost'(u) + cost'(v) - cost(T_r) - cost(T_s)$). After $T_r(e_u)$ and $T_s(e_v)$ are merged into a new tree T_t, we have to update $mr'(w)$ and $cost'(w)$ for each node

$w \in T$. By a depth-first traversal of the tree, $mr'(w)$ and $cost'(w)$ of all nodes $w \in T$ can be computed in $(|T|)$ time; the resulting time complexity is dominated by the construction of H.

Since the number of nodes in H is $O(n)$, straightforward computation of all edge weights takes $O(n^2)$ time. The nearest-neighbor graph will be constructed and used to merge the remaining subtrees $O(\log n)$ times, if the parameter f is a constant (cf. the discussion of Greedy-DME in Section 2). Thus, the time complexity of Greedy-BST/DME is $O(n^2 \log n)$. By using the bucket decomposition method of [Edahiro 1994], the nearest-neighbor graph can be constructed in linear time, so that the total time complexity becomes $O(n \log n)$.

Note that when the root is repositioned at a tree edge at the lower level, the tree will become very unbalanced. Thus, when the skew bound B is small, significant detour wiring will be required to maintain the given skew bound. Thus, in our implementation we compute the possible root positions by examining each tree edge in the top-down manner. When it is found that the detour wiring is required when the tree root is repositioned at edge e_v, then we ignore all the edges in the subtree T_v. Thus, when $B = 0$, Greedy-BST/DME almost merges each pair of two subtrees at their roots, and has the same linear time complexity as Greedy-DME.

6. EXPERIMENTAL RESULTS

We have implemented the BST/DME and Greedy-BST/DME algorithms in ANSI C on Sun SPARC-20 machines. Recall that Greedy-BST/DME considers merging region construction with topology generation, whereas BST/DME considers only merging region construction for a given topology. To avoid cumbersome notation, we would simply use BME (or IME) to refer to the BST/DME algorithm, as well as the Greedy-BST/DME algorithm, that employ the BME (or IME) method to construct merging region. The context in which BME or IME is used should differentiate between BST/DME and Greedy-BST/DME clearly. Moreover, we use IME-GS to refer to the IME method with greedy selection of merging regions, and IME-DPS to refer the version employing dynamic programming-based selection technique. If no distinction is required, we simply refer to both IME-GS and IME-DPS as IME.

The benchmark test cases r1–r5 [Tsay 1993] were used to evaluate our Greedy-BST/DME algorithms for skew bounds in the range of 0–10ns. Table I compares the various bounded-skew routing costs obtained by our algorithms with (i) the best reported zero-skew clock routing costs of the best known algorithm (CL+I6 from [Edahiro 1993a]) and (ii) the Steiner tree routing costs of one of the best known heuristics, the BOI Steiner algorithm [Borah et al. 1994]. Also included in Table I are the CPU times for BME, IME-GS, and IME-DPS.

In this experiment, the IME algorithms keep at most $k = 5$ merging regions for each internal node, and slices each merging region to $s = 7$ Manhattan arcs for merging with other nodes. For instances where IME-DPS performs better than IME-GS, the average wirelength is 5.8% less. On the other hand, when IME-GS performs better, the average wirelength is 3% less. In terms of overall wirelength, IME-DPS is slightly better than IME-GS by an arithmetic mean of 1.5% less wirelength.[3] Moreover, we note that the run-times of IME-DPS and IME-GS are comparable although the dynamic programming selection method has a higher complexity than the greedy selection method. This is due to the fact that the bottleneck in the Greedy-BST/DME algorithm is the computation of nearest-neighbor graph (See Sections 2.1 and 5), but not merging region construction. In general, the run-time of IME due to the nearest-neighbor graph computation is no more than k^2 times that of BME. Therefore, in subsequent experiments, we will consider only IME-DPS solutions.

We also observe that both BME and IME have comparable results, with IME solutions having marginally less wirelength than that of BME solutions by an arithmetic mean of 1%. It appears that IME produces better results for larger circuits r3–5, but at the expense of longer run-times. In particular, IME performs very well for circuit r3, achieving an average of 6% wirelength reduction compared to BME solutions. On the other hand, BME in general performs better for smaller circuits. More importantly, we see a decrease in total wirelength for solutions constructed by both BME and IME as the skew bound increases. On average, a 42% wirelength reduction is observed when varying the skew bound from 0 to ∞.

When the skew bound $B = 0$, our routing costs are on average 5.4% higher than that of [Edahiro 1993a]. We believe that this is due to the fact that the CL+I6 algorithm performs local optimization using exhaustive search and calculates an optimum sequence, which we did not implement in our algorithm. The other reason is that [Edahiro 1993a] used the best result from eight different values of the f parameter ranging from 2 to 4 (recall the discussion of the Greedy-DME algorithm in Section 2.1), while we use only $f = 1$. When $B = \infty$, the Steiner trees constructed by BME average only 1.47% higher cost than those constructed by the BOI algorithm [Borah et al. 1994]. Note that BME and IME produce identical routing solutions when $B = 0$. If IME also considers rectilinear sampling segments, then BME and IME produce identical routing solutions when $B = \infty$.

To make a fairer comparison between the performance of BME and IME, we ran both (BST/DME) algorithms on the topologies generated by the Greedy-BST/DME algorithms in the previous experiment. In this case, IME uses *both* Manhattan arcs and rectilinear segments to sample each regions. The results are shown in Tables II and III. Since the run-times in both tables are similar, we show only the run-times of the algorithms in Table II. Table II shows a better IME performance than that in Table I where only Manhattan arcs are considered as sampling segments. The IME solutions in Table II have an average of 2% less wirelength than the IME solutions in Table I. Also, both Tables II and III show that IME outperforms BME in most cases, but with longer CPU times. Note that, however, the gain by IME is again marginal.

[3] To obtain the arithmetic mean, we first normalized the IME-DPS wirelengths with respect to the corresponding IME-GS wirelengths. The arithmetic mean is the average of the normalized wirelengths. We also computed the geometric mean of the normalized wirelengths, and found that the arithmetic mean and geometric mean obtained in this manner match very closely. Therefore, we report the arithmetic mean, i.e., the average, only in this and subsequent comparisons.

Table I. The cost-skew tradeoff and run-times of the BME and IME algorithms for benchmark circuits r1-5 [Tsay 1993]. We mark the cases where IME-GS and IME-DPS outperform BME by † and *, respectively.

Skew Bound (ps)	Algorithm	Wirelengths (CPU time: hr:min:sec)				
		r1	r2	r3	r4	r5
0	CL+I6	1253347	2483754	3193801	6499660	9723726
	BME or IME	1307145	2628882	3332253	6804479	10493166
		(00:00:11)	(00:00:18)	(00:00:27)	(00:01:11)	(00:01:59)
1	BME	1229755	2501043	3062376	6797056	9881525
		(00:00:42)	(00:02:07)	(00:02:29)	(00:05:57)	(00:10:09)
	IME-GS	1350863	†2417042	†3026204	†6750572	9958339
		(00:02:48)	(00:08:15)	(00:13:13)	(00:31:37)	(01:01:12)
	IME-DPS	1274819	2526961	*3060284	*6751435	9896655
		(00:03:30)	(00:09:33)	(00:14:54)	(00:40:20)	(01:08:39)
10	BME	1084381	2191861	2803597	5966729	8092600
		(00:01:14)	(00:02:53)	(00:04:37)	(00:10:53)	(00:18:26)
	IME-GS	1087215	2388383	2996170	†5549435	8434325
		(00:06:18)	(00:17:35)	(00:26:02)	(01:14:36)	(03:23:30)
	IME-DPS	1112512	2353202	*2727299	*5350241	*8065110
		(00:06:06)	(00:17:33)	(00:28:44)	(01:18:07)	(02:45:30)
100	BME	941141	2208774	2676650	4769482	7668381
		(00:01:42)	(00:04:13)	(00:05:56)	(00:14:19)	(00:22:36)
	IME-GS	947481	†2118057	†2597786	4889710	†7640582
		(00:14:00)	(00:40:45)	(00:48:44)	(03:06:50)	(06:21:56)
	IME-DPS	*930426	*1978378	*2366874	4953715	*7003996
		(00:13:18)	(00:37:30)	(01:01:34)	(03:09:16)	(06:00:21)
1000	BME	793498	1779280	2361706	4729012	6242931
		(00:01:59)	(00:05:05)	(00:09:03)	(00:16:32)	(00:31:09)
	IME-GS	874749	†1693660	†2116823	4733074	7129680
		(00:25:48)	(01:19:07)	(01:56:45)	(07:05:35)	(12:27:36)
	IME-DPS	861561	1995665	*2097784	4740962	6280007
		(00:25:24)	(01:13:03)	(02:06:22)	(06:41:31)	(13:42:13)
10000	BME	780100	1668872	2102182	4060592	6024276
		(00:02:02)	(00:05:36)	(00:08:46)	(00:19:12)	(00:28:09)
	IME-GS	788359	†1547491	†1995801	†3971843	6610821
		(00:33:39)	(01:59:18)	(02:48:48)	(11:51:13)	(23:29:55)
	IME-DPS	790285	*1574153	*1998007	4326234	*5912472
		(00:32:30)	(02:07:21)	(03:12:15)	(10:30:53)	(24:39:45)
∞	BME or IME	780100	1528084	1929421	3836130	5638069
		(00:02:03)	(00:06:06)	(00:08:49)	(00:24:43)	(00:39:17)
	BOI	769260	1498760	1902560	3781390	5571100
		(00:00:05)	(00:00:25)	(00:00:53)	(00:04:31)	(00:06:03)

Based on the results in Tables I–III, we also try to compare the quality of the topologies generated by Greedy-BST/DME with BME and Greedy-BST/DME with IME, respectively. Comparing the solutions obtained by BME embedding in Tables I (topologies generated by Greedy-BST/DME with BME) and II (topologies generated by Greedy-BST/DME with IME), we found that IME-generated topologies have an average of 2% less wirelength. Similarly, comparing the solutions obtained by IME embedding in Tables III (BME-generated topologies) and II (IME-generated topologies), IME-generated topologies have an average of 2.3% less wirelength after embedding. It seems to be the case that IME-generated topologies are better than BME-generated topologies.

Table II. The performance comparison between BME and IME using the topologies generated by Greedy-BST/DME with IME. We mark the cases where IME outperforms BME by *.

Skew Bound (ps)	Algorithm	Wirelengths (CPU time: min:sec)				
		r1	r2	r3	r4	r5
1	BME	1273637 (00:01)	2517746 (00:03)	3053304 (00:05)	6723972 (00:13)	9859802 (00:24)
	IME	*1270938 (00:27)	*2516781 (00:54)	*3052404 (01:19)	*6717013 (02:59)	*9848820 (04:55)
10	BME	1099627 (00:01)	2329605 (00:03)	2697450 (00:05)	5294651 (00:13)	7984073 (00:25)
	IME	*1098997 (00:23)	*2326915 (01:03)	*2695746 (01:17)	*5290219 (05:58)	*7976755 (05:06)
100	BME	913598 (00:01)	1949013 (00:03)	2326170 (00:05)	4864974 (00:12)	6879762 (00:23)
	IME	*913501 (00:25)	*1948385 (00:55)	*2325800 (01:19)	*4864659 (02:58)	6880406 (05:07)
1000	BME	904989 (00:01)	1966189 (00:03)	2056433 (00:05)	4657888 (00:11)	6159874 (00:24)
	IME	*844664 (00:27)	*1961365 (00:58)	*2056315 (01:22)	4657929 (07:12)	6161011 (05:38)
10000	BME	775249 (00:01)	1540561 (00:03)	1958586 (00:04)	4245122 (00:12)	5793815 (00:23)
	IME	*775249 (00:34)	1540944 (01:12)	1958728 (01:32)	*4234894 (04:00)	5794538 (06:13)

Table III. The performance comparison between BME and IME using the same topologies generated by Greedy-BST/DME with BME. The total wirelengths of BME for different skew bounds are given in Table I. The table shows only the wirelengths of IME routing solutions. We mark the cases where IME outperforms BME by *.

Skew Bound (ps)	Wirelengths for IME				
	r1	r2	r3	r4	r5
1	*1222431	*2394473	3426513	*6404626	*9454704
10	1085001	*2154839	*2789091	*5401802	8126549
100	*925630	*2197963	*2515446	4848367	*7364356
1000	795342	1837930	2503806	4951131	7132643
10000	*780100	1669490	2102617	*4019368	6136801

When BME is run on the topologies given in Tables II and III, we found that for these cases (i) there is at most one internal nodes which child nodes have overlapped merging regions, (ii) there are less than 2% internal nodes which need detour wiring when their children are merged, (iii) all joining segments are equal to the shortest distance segments.[4] Therefore, the merging regions constructed by BME are equal to the min-cost merging regions in most cases.

A more detailed experiment using Greedy-BST/DME on all benchmark circuits was conducted to investigate the tradeoff between total wirelength and skew, and the tradeoff between power dissipation and skew for realistic skew bounds in the range of $0 - 150ps$. We used HSPICE simulations to measure the power dissipation for benchmarks r1-3 at 50MHz, and r4-5 at 5MHz (due to the rise/fall time constraints). Due to space limitation, we only show the result of IME for the benchmark circuits r3 in Fig. 18.

When the skew bound is relaxed from zero to $150ps$, we achieved a average power reduction of up to 18.4%. We also achieved 26.6% average wirelength reduction when compared to the best reported zero-skew solutions (by the CL+I6 algorithm in [Edahiro 1993a]).

To further justify the superiority of Greedy-BST/DME over Greedy-DME in terms of topology generation, we also run BME to embed the topologies generated by Greedy-DME for the benchmark circuit r1 under different skew bounds. As we can see from Fig. 19, only up to 16% wirelength reduction is achieved by the method of topology generation by Greedy-DME followed by BME embedding. On the other hand, Greedy-BST/DME with BME achieves up to 40% wirelength reduction while the run-times only increases at most 4 times than Greedy-DME. (Greedy-BST/DME with IME has similar wirelength as Greedy-BST/DME with BME, but its run-time is longer). Another comparison between topologies by Greedy-BST/DME and Greedy-DME can be seen from [Huang et al. 1995; Cong and Koh 1995]. Finally, we give the layout of BME routing solutions generated by Greedy-BST/DME for benchmark r1 with skew bound $B = 1ps$, $10ps$, and ∞ in Fig. 20.

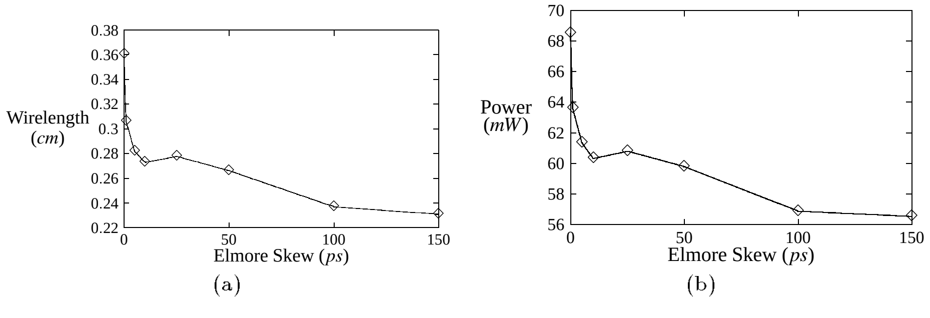


Fig. 18. Tradeoff between (a) total wirelength and skew bound, and (b) power dissipation and skew bound for benchmark r3.

[4]In these experiments, all pairs of joining segments are either rectilinear segments with the same quadratic terms $\frac{rc}{2} \cdot x$ or Manhattan arcs with constant delays. So Cases I and II in Section 3.5 never happen.

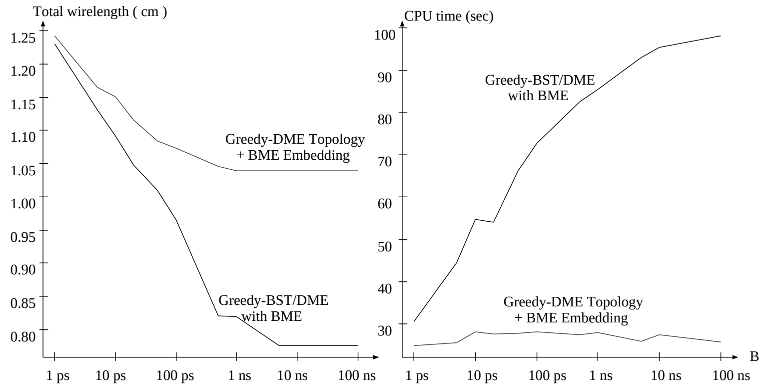
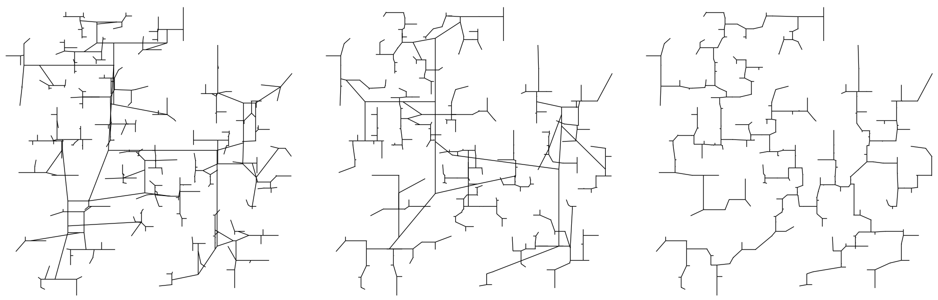


Fig. 19. Comparisons of (a) total wirelength and (b) run-times between topologies produced by Greedy-BST/DME and Greedy-DME for various skew bounds from $1ps$ to $100ns$



B=1ps
Cost=1229755

B=10 ps
Cost=1112512

B=∞
Cost=780100

Fig. 20. Layout of BME routing solutions for benchmark r1 with skew bound $B = 1$, 10, and ∞ ps.

7. CONCLUSION AND FUTURE WORK

In conclusion, we have presented new bounded-skew routing tree approaches under the pathlength and Elmore delay models. We prove several key properties of the merging regions under the two delay models. We propose two approaches to construct merging regions for internal nodes of the topology tree. Our first approach, called BME, utilizes merging points that are restricted to the boundaries of merging regions. A second approach, called IME, employs a sampling strategy and dynamic programming to consider merging points that are interior to the merging regions. We also propose a new algorithm that dynamically construct the routing tree topology as we compute the merging regions.

Our current implementation of IME only uses simple sampling segment like Manhattan arcs and rectilinear segments. We are studying if well-behaved sampling segments with other orientations will improve the results. We are also studying better sampling strategies for speed-up of the IME method. One way to speed up IME is to sample regions according to a variable-size sampling set; we can use fewer sampling segments for smaller merging regions. Another possible speed-up technique is to avoid the generation of redundant regions, instead of eliminating redundant regions *after* they are generated. In our experiments, we use k=5 and s =7, so that merging two nodes could produce more than $k^2s^2 > 1000$ merging regions. However, the number of *irredundant* regions is never larger than 50.

Our final goal is to extend our BST/DME method to (i) incorporate our recent work on optimal sizing of interconnects and drivers/buffers [Cong and Koh 1994; Cong and Leung 1995; Cong et al. 1996b], and (ii) consider practical clock routing issues such as the various layer parasitics, macro cell blockages, and the hierarchy of clock buffer design [Kahng and Tsao 1997a; Kahng and Tsao 1997b].

APPENDIX

A. PROOFS OF LEMMAS AND THEOREMS

A.1 Proofs of Lemmas for Theorem 1

In the following, we give the proofs of lemmas that are used to prove Theorem 1. Lemma 1, along with Fact 2 (in Section 3.3), are used to prove the correctness of BME construction rules for the case where the joining segments L_a and L_b are parallel Manhattan arcs, while Lemmas 2-5 are used for the case where the joining segments L_a and L_b are parallel rectilinear segments.

Let a and b be the children of node $v \in G$, and we assume that they are associated with well-behaved merging regions $P = mr(a)$ and $Q = mr(b)$, respectively. Also we let $L_a = JS_Q(P)$, $L_b = JS_P(Q)$, and $R = SDR(L_a, L_b)$. Without loss of generality, if L_a and L_b are rectilinear, we assume that they are vertical in the following proofs.

Lemma 1. *Any rectilinear line segment $l \subseteq SDR(L_a, L_b)$ after merging L_a and L_b is well-behaved.*

Proof: First, recall that we refer to the original delay functions defined for points $u \in l$ as $max\_\bar{t}(u)$ and $min\_\bar{t}(u)$, and refer to the new delay functions defined over l after merging as $max\_t(u)$ and $min\_t(u)$.

Consider the case where the joining segments L_a and L_b are parallel Manhattan arcs with constant max-delays and min-delays, as in Fig. 21(a). Let the max-delays of L_a and L_b be $max\_\bar{t}(L_a)$ and $max\_\bar{t}(L_b)$, respectively. It is obvious that the minimum pathlength from point $u \in l$ to L_a is $|e_a| = d(u, L_a) = d(u,p) + d(p, L_a) = x + d(p, L_a)$ and that the minimum pathlength from u to L_b is $|e_b| = d - |e_a| = d - x - d(p, L_a) = -x + d(p, L_b)$, where $x = d(u, p)$ and $d = d(L_a, L_b)$. Therefore, the max-delay function of point $u \in l$ is

$$\begin{aligned} max\_t(u) &= \max\{max\_\bar{t}(L_a) + |e_a|, \quad max\_\bar{t}(L_b) + |e_b|\} \\ &= \max\{x + max\_\bar{t}(L_a) + d(p, L_a), -x + max\_\bar{t}(L_b) + d(p, L_b)\}. \quad (8) \end{aligned}$$

Therefore, the max-delay function of l conforms to the form given in Eqn. (3). We say that l is well-behaved with respect to (w.r.t.) the max-delay. By replacing max_t, $max\_\bar{t}$, and max in Eqn. (8) by min_t, $min\_\bar{t}$, and min, respectively, one can easily verify that l is well-behaved w.r.t. the min-delay, too. Therefore, l is well-behaved. In the rest of the proof, we will only prove that l is well-behaved w.r.t. the max-delay. The proof for the well-behaved property w.r.t. the min-delay can be derived in a similar fashion.

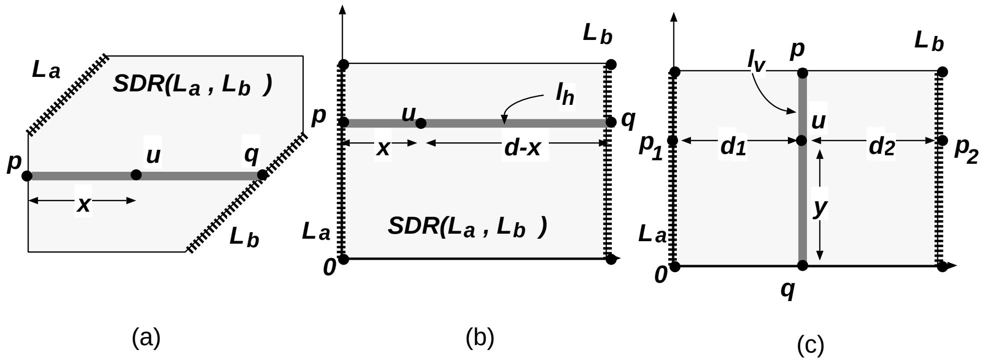


Fig. 21. Given vertical well-behaved segment L_a and L_b, any vertical or horizontal segment in $SDR(L_a, L_b)$ is also well-behaved.

Next, consider the case where L_a and L_b are vertical joining segments, as shown in Fig. 21(b). If $l_h = \overline{pq} \subseteq SDR(L_a, L_b)$ is a horizontal segment with $p \in L_a$ and $q \in L_b$, then for any point $u \in l_h$, we have

$$max\_t(u) \;=\; \max\{x + max\_\bar{t}(p), d - x + max\_\bar{t}(q)\} \qquad (9)$$

where $x = d(p, u)$. So l_h is well-behaved w.r.t. the max-delay.

Finally, we prove that any vertical line segment $l_v = \overline{pq} \subseteq SDR(L_a, L_b)$ is well-behaved, as shown in Fig. 21(c). Since L_a and L_b are well-behaved, we can assume that for any point $p_1 = (0, y) \in L_a$,

$$max\_\bar{t}(p_1) = \max\{y + \alpha_1, -y + \beta_1\}.$$

Similarly, we can assume that for any point $p_2 = (d, y) \in L_b$,

$$max\_\bar{t}(p_2) = \max\{y + \alpha_2, -y + \beta_2\}.$$

Let $d_1 = d(l_v, L_a)$ and $d_2 = d(l_v, L_b)$. Then for point $u = (d_1, y) \in l_v$ and points $p_1 \in L_a$ and $p_2 \in L_b$ which have the same y-coordinates as u,

$$\begin{aligned} max\_t(u) &= \max\{max\_\bar{t}(p_1) + d_1, max\_\bar{t}(p_2) + d_2\} \\ &= \max\{y + \alpha_1 + d_1, -y + \beta_1 + d_1, y + \alpha_2 + d_2, -y + \beta_2 + d_2\} \\ &= \max\{y + \max\{\alpha_1 + d_1, \alpha_2 + d_2\}, -y + \max\{\beta_1 + d_1, \beta_2 + d_2\}\} \end{aligned}$$

Note that $y = d(u, q)$. Since $\max\{\alpha_1 + d_1, \alpha_2 + d_2\}$ and $\max\{\beta_1 + d_1, \beta_2 + d_2\}$ are constants, l_v is well-behaved w.r.t. the max-delay. □

Lemma 2. *Suppose L_a and L_b are vertical, and $l = \overline{pq} \subseteq SDR(L_a, L_b)$ is a horizontal line segment connecting $p \in L_a$ and $q \in L_b$ (Fig. 7(b)). If $skew\_const(l) \neq \emptyset$, then $skew\_const(l) \subseteq FMR(l)$.*

Proof: Let $\overline{skew}(p)$ and $\overline{skew}(q)$ denote the original skew of points p and q. Since $p \in L_a \subseteq mr(a)$ and $q \in L_b \subseteq mr(b)$, so p and q are feasible merging points before $mr(a)$ and $mr(b)$ are merged, i.e., $\overline{skew}(p) \leq B$ and $\overline{skew}(q) \leq B$. Let $d = d(p,q)$ and $l_c = skew\_const(l)$. Either one of endpoints of l_c, say u, will be either the max-delay or min-delay turning point. Suppose u is the max-delay turning point, then by Eqn. (9) in the proof of Lemma 1,

$$max\_t(u) = x + max\_\bar{t}(p) = d - x + max\_\bar{t}(q),$$

where $x = d(p,u)$. Similarly, if u is the min-delay turning point, then

$$min\_t(u) = x + min\_\bar{t}(p) = d - x + min\_\bar{t}(q)$$

In either case, we can easily prove that

$$\begin{aligned} skew(u) &= max\_t(u) - min\_t(u) \\ &\leq \max\{max\_\bar{t}(p) - min\_\bar{t}(p), max\_\bar{t}(q) - min\_\bar{t}(q)\} \\ &= \max\{\overline{skew}(p), \overline{skew}(q)\} \leq B \end{aligned}$$

Since $skew(l_c) = skew(u)$, we have $l_c \subseteq FMR(l)$. Note that this lemma holds under any monotone delay model. □

LEMMA 3. *Suppose L_a and L_b are vertical. Let $R = SDR(L_a, L_b)$. (i) If $FMR(R) \neq \emptyset$, then the max-delay, min-delay, and skew values change monotonically at constant rates $+1$, -1, and $+2$ respectively as we traverse from the boundaries of $FMR(R)$ horizontally to L_a or L_b (see Fig. 7(b)). (ii) If $FMR(R) = \emptyset$, then $MSR(R) = MSR(L_a)$ if $skew(MSR(L_a)) < skew(MSR(L_b))$, and $MSR(R) = MSR(L_b)$ otherwise (see Fig. 7(d)).*

Proof: Let l be a horizontal line segment in R. By Lemma 1, l is well-behaved. If $FMR(R) \neq \emptyset$, then by Lemma 2 $skew\_const(l) \subseteq FMR(l) \subseteq FMR(R)$. That is, there is no skew turning point on l in the region $R - FMR(R)$. Therefore, the max-delay, min-delay, and skew values increase at constant rates $+1$, -1, and $+2$ respectively along l from the boundaries of $FMR(R)$ toward L_a or L_b.

On the other hand, if $FMR(R) = \emptyset$, we must have $skew\_const(l) = \emptyset$ (i.e., no skew turning points on l). Thus, either $l = skew\_incr(l)$ or $l = skew\_decr(l)$, That is, the skew values change monotonically along l. Therefore, $MSR(R) = MSR(L_a)$ if skew values increase along l, and $MSR(R) = MSR(L_b)$ otherwise. In other words, $MSR(R) = MSR(L_a)$ if $skew(MSR(L_a)) < skew(MSR(L_b))$, and $MSR(R) = MSR(L_b)$ otherwise. □

LEMMA 4. *If $FMR(R) = \emptyset$, then $mr(v)$ constructed by the BME construction rules (i) is $MSR(R)$, and (ii) is a well-behaved octilinear region (segment) with merging cost $|e_a| + |e_b| = d(L_a, L_b) + min\_skew(R) - B$, which is minimum subject to the constraint that $p \in L_a$ can merge with $q \in L_b$ only if $d(p,q) = d(L_a, L_b)$.*

Proof: Without losing generality, let us consider the case where L_a and L_b are vertical, as shown in Fig. 7(d). Let u be a point on a horizontal line segment $l = \overline{pq}$ with $p \in L_a$ and $q \in L_b$, and let $|e_a|$, $|e_b|$ be the lengths of the edges from u to p and q. Then,

$$max\_t(u) = \max\{max\_\bar{t}(p) + |e_a|, max\_\bar{t}(q) + |e_b|\}$$

$$min\_t(u) = \min\{min\_\bar{t}(p) + |e_a|, min\_\bar{t}(q) + |e_b|\}.$$

Now, consider the case where both e_a and e_b have no detour wiring. Then, $|e_a| = d(p,u)$, and $|e_b| = d(p,q) - |e_a|$. Let $x = d(p,u)$. We claim that only one of the following sets of inequalities holds:

$$max\_\bar{t}(p)+x > max\_\bar{t}(q)+d(p,q)-x, \quad min\_\bar{t}(p)+x > min\_\bar{t}(q)+d(p,q)-x, \quad (10)$$

$$max\_\bar{t}(p)+x < max\_\bar{t}(q)+d(p,q)-x, \quad min\_\bar{t}(p)+x < min\_\bar{t}(q)+d(p,q)-x. \quad (11)$$

Otherwise, one can easily verify that $skew(u) = \overline{skew}(p)$ or $\overline{skew}(q) \leq B$, contradicting the assumption that $FMR(R) = \emptyset$. For the case where Eqn. (10) holds, $skew(u) = max\_\bar{t}(p) - min\_\bar{t}(q) - d(p,q) + 2x > B$. To make u feasible with minimum merging cost $|e_a| + |e_b|$, we add detour wiring of $max\_\bar{t}(p) - min\_\bar{t}(q) - d(p,q) + 2x - B$ to e_b such that $|e_b| = max\_\bar{t}(p) - min\_\bar{t}(q) + x - B$. After adding the detour wiring, the skew of u is exactly B, and the total merging cost is $|e_a| + |e_b| = max\_\bar{t}(p) - min\_\bar{t}(q) + 2x - B$. Therefore, to minimize the merging cost of p and q, one should make $x = 0$, i.e., $u = p$, $|e_a| = 0$, and $|e_b| = max\_\bar{t}(p) - min\_\bar{t}(q) - B = skew(p) + d(L_a, L_b) - B$. Hence, to minimize the merging cost of L_a and L_b, $p \in L_a$ should be chosen such that $skew(p) = min\_skew(R)$, i.e., $p \in MSR(R)$. Since $|e_a| = 0$, $u = p \in MSR(R)$ and thus $mr(v) = MSR(R) = MSR(L_a)$ (Lemma 3). Similarly, we can prove the lemma for the case where Eqn. (11) holds, or equivalently, $MSR(R) = MSR(L_b)$. Therefore, $mr(v)$ constructed by the BME construction rules is a line segment (or point) $MSR(R)$ with skew = B, and the merging cost $|e_a| + |e_b|$ is minimum. □

Lemma 5. *If $FMR(R) \neq \emptyset$, then $mr(v)$ constructed by the BME construction rules (i) is equal to $FMR(R)$, and (ii) is a well-behaved octilinear region with minimum merging cost = $d(L_a, L_b)$.*

Proof: If L_a and L_b are parallel Manhattan arcs (with constant delay values), as shown in Fig. 7(a) , then by Fact 2 and Lemma 1, it is easy to see that
(i) $FMR(R)$ is a well-behaved convex region with at most 6 boundary segments, and
(ii) $mr(v)$ constructed by the BME construction rules is equal to $FMR(R)$.

We next consider the case where, without loss of generality, L_a and L_b are parallel vertical segments as in Fig. 22. Based on (1) $skew(l) = B$ where l is a boundary segment of $FMR(R)$ and l is not a boundary segment of R; and (2) the delay and skew values change at constant rates (± 1 and ± 2) along L_a, L_b, and the horizontal lines between $FMR(R)$ and L_a (or L_b) (Lemmas 3 and Lemma 1); we prove informally in the following that (I) $FMR(R)$ must be an octilinear polygon with at most 8 sides, and (II) the vertices of $FMR(R)$ lie either on the boundary segments of R or opposite the skew turning points of L_a or L_b.

Now, Consider $l_c = skew\_const(L_b)$ in Fig. 22. If $skew(l_c) \leq B$, $FMR(R)$ has a rectilinear boundary segment on L_b (Fig. 22(a)). Otherwise, by moving l_c to the left by $\frac{skew(l_c)-B}{2}$ units, we obtain a boundary segment of $FMR(R)$ which is parallel to $skew\_const(L_b)$ (Fig. 22(b)). Since $FMR(R) \subseteq SDR(L_a, L_b)$, by Lemma 1, any rectilinear boundary segment of $FMR(R)$ is well-behaved.

Now, consider another three points in region M in Fig. 22.

—$q_1 = (x, y) \in L_b$ which has $skew(q_1) \geq B$ and is above q. Note that q and q_1 may coincide (see Fig. 22(b)).

—$q_2 = (x - \delta\ ,\ y + \delta)$, where $\delta \geq 0$.

—$q_3 = (x, y + \delta)$.

Then, by Lemma 1, we can compute $max\_t(q_3) = max\_t(q_1) + \delta$, and by Lemma 3, $max\_t(q_2) = max\_t(q_3) - \delta = max\_t(q_1)$. Therefore, all the points on Manhattan arc $\overline{q_1 q_2}$ have a constant max-delay. Similarly, we can show that $\overline{q_1 q_2}$ has a constant min-delay, and therefore constant skew ($= skew(q_1)$.) Therefore, we can easily see that (I) and (II) are correct.

Thus, $FMR(R)$ is a well-behaved region and the merging region $mr(v)$ constructed by the BME construction rules is actually $FMR(R)$. Clearly, each point in $FMR(R)$ has minimum merging cost $= d(L_a, L_b)$. So $mr(v) = FMR(R)$ has minimum merging cost. □

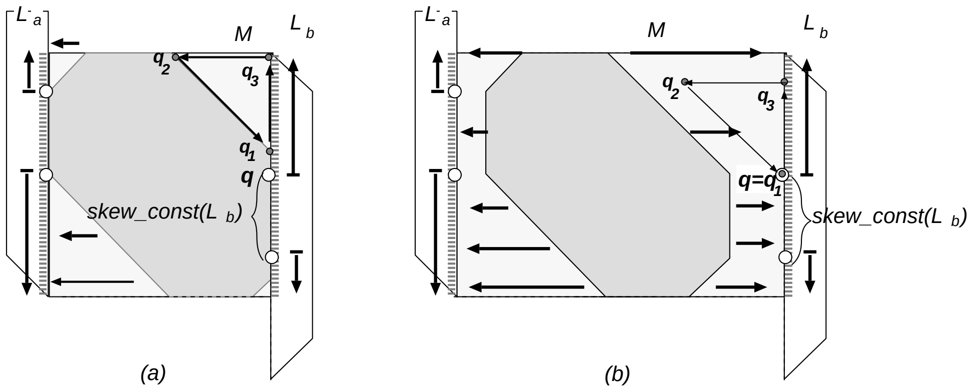


Fig. 22. Proof of the correctness of the BME construction rules for the case where joining segments L_a and L_b are vertical, and $FMR(R) \neq \emptyset$, where $R = SDR(L_a, L_b)$. Arrows indicate the directions of increasing skew/max-delay and decreasing min-delay.

A.2 Proofs of Theorem 2

Proof of Theorem 2 will be based on the following Lemmas, some of which are simply the extension of their counterparts under pathlength delay. First, we have the following fact from the discussion of the example in Fig. 8.

FACT 3. *Let line segment $l \subseteq SDR(L_a, L_b)$ be (i) either horizontal or vertical when L_a and L_b are parallel Manhattan arcs with constant delays or (ii) horizontal (vertical) when L_a and L_b are parallel segments with slopes > 1 or < -1 (between 1 and -1). Then line segment l can be divided into at most three consecutive linear regions (from left to right), denoted $skew\_decr(l)$, $skew\_const(l)$, and $skew\_incr(l)$, in which the skew changing rates are respectively $-r(Cap(a) + Cap(b) + hc)$, 0, and $+r(Cap(a) + Cap(b) + hc)$, where $h = d(L_a, L_b)$ and again $Cap(a)$ and $Cap(b)$ are the total capacitance of the subtrees rooted at nodes a and b.* □

LEMMA 6. *Let $f_1(x)$ and $f_2(x)$ be m_1-piecewise-linear and m_2-piecewise-linear functions, respectively. (i) If both $f_1(x)$ and $f_2(x)$ are convex, then $\max\{f_1(x), f_2(x)\}$ is an n-piecewise-linear convex function, where $n \leq m_1 + m_2 - 1$. (ii) If both $f_1(x)$ and $f_2(x)$ are concave, then $\min\{f_1(x), f_2(x)\}$ is an n-piecewise-linear concave function, where $n \leq m_1 + m_2 - 1$. (iii) If $f_1(x)$ is convex and $f_2(x)$ is concave, then $f_1(x) - f_2(x)$ is an n-piecewise-linear convex function, where $n \leq m_1 + m_2 - 1$.*

Proof: It is easy to see that (i) and (ii) hold. In case (iii) $f_1(x)$ and $f_2(x)$ have $m_1 - 1$ and $m_2 - 1$ turning points, respectively, so $f_1(x) - f_2(x)$ will have at most $m_1 + m_2 - 2$ turning points. Since the slopes of $f_1(x)$ increase and the slopes of $f_2(x)$ decrease as x increases, the slopes of $f_1(x) - f_2(x)$ will be an increasing function of x. Thus $f_1(x) - f_2(x)$ is an n-piecewise-linear convex function, where $n \leq m_1 + m_2 - 1$. □

LEMMA 7. *Let L_a and L_b be two parallel well-behaved joining segments which are not Manhattan arcs with constant delays. Also let $R = SDR(L_a, L_b)$. Then any line segment $l \in R$ is well-behaved if the given delay functions defined over L_a and L_b (before merging) have the same quadratic term.*

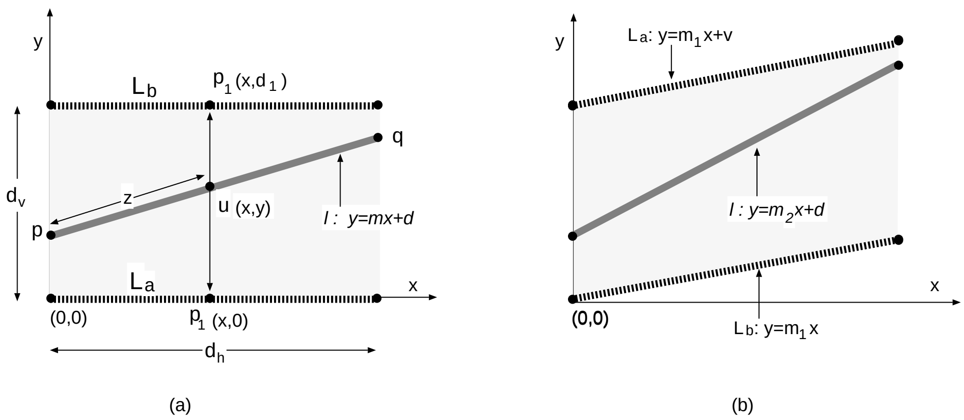


Fig. 23. Any line segment $l \subseteq SDR(L_a, L_b)$ (the dotted region) is well-behaved if L_a and L_b are well-behaved segments and the delay functions defined over L_a and L_b have the same quadratic term.

Proof: Assume that L_a and L_b are two horizontal line segments, as shown in Fig. 23(a). The same arguments of the well-behaved property of line segment $\overline{ab}$ in Fig. 8 show that any vertical line segment $l \in R$ is well-behaved and that the delay functions defined over l will have the quadratic term $K \cdot y^2$. So, in the following we will only prove that any non-vertical line segment $l = \overline{pq} \in R$ is well-behaved. As shown in Fig. 23(a), line segment l is described by the equation $y = mx + d$ where $m \neq \infty$, $0 \leq x \leq d_h$, and d is a constant. Also, for any $u = (x, y) \in l$ let $p_1 = (x, 0) \in L_a$ and $p_2 = (x, d_v) \in L_b$, where $d_v = d(L_a, L_b)$. Then we define $A(x, y)$ to be the max-delay from u via p_1 to sinks in the subtree rooted at node a.

Similarly, let $B(x, y)$ be the max-delay from u via p_2 to sinks in the subtree rooted at node b.

Since L_a and L_b are well-behaved and the delay functions defined over L_a and L_b have the same quadratic term, we can assume that $max\_\bar{t}(p_1) = \max_{i=1,\cdots,s_1}\{m_i x + d_i\} + K \cdot x^2$ for point $p_1 = (x, 0) \in L_a$, and $max\_\bar{t}(p_2) = \max_{i=1,\cdots,s_2}\{n_i x + e_i\} + K \cdot x^2$ for point $p_2 = (x, d_v) \in L_b$. Let $\alpha_1 = r \cdot Cap(a)$, $\alpha_2 = -r(cd_v + Cap(b))$, and $\gamma = K \cdot (d_v)^2 + r \cdot d_v \cdot Cap(b)$. Then, for point $u = (x, y) \in l$, we have $max\_t(u)$ $= \max\{A(x, y),\, B(x, y)\}$ with

$$\begin{aligned} A(x,y) &= K \cdot y^2 + \alpha_1 y + max\_\bar{t}(p_1) \\ &= K \cdot (mx+d)^2 + \alpha_1(mx+d) + \max_{i=1,\cdots,s_1}\{m_i x + d_i\} + K \cdot x^2 \\ B(x,y) &= K \cdot y^2 + \alpha_2 y + \gamma + max\_\bar{t}(p_2) \\ &= K \cdot (mx+d)^2 + \alpha_2(mx+d) + \gamma + \max_{i=1,\cdots,s_2}\{n_i x + e_i\} + K \cdot x^2 \end{aligned}$$

Let $K' = (1+m^2)K$, $m' = 2mdK + \alpha_1 m$, $d' = K \cdot d^2 + \alpha_1 d$, $n' = 2mdK + \alpha_2 m$, and $e' = K \cdot d^2 + \alpha_2 d + \gamma$. Then, we can write $A(x, y)$ and $B(x, y)$ as

$$\begin{aligned} A(x,y) &= \max_{i=1,\cdots,s_1}\{(m'+m_i)x + (d'+d_i)\} + K' \cdot x^2 \\ B(x,y) &= \max_{i=1,\cdots,s_2}\{(n'+n_i)x + (e'+e_i)\} + K' \cdot x^2 \end{aligned}$$

Let $f_1(x) = \max_{i=1,\cdots,s_1}\{(m'+m_i)x + (d'+d_i)\}$ and $f_2(x) = \max_{i=1,\cdots,s_2}\{(n' + n_i)x + (e'+e_i\}$, which are s_1- and s_2-piecewise-linear convex functions, respectively. Let $f_3(x) = \max\{f_1(x), f_2(x)\}$, which by Lemma 6 is an s-piecewise-linear convex function of x with $s \le s_1 + s_2 - 1$. Therefore, $max\_t(u) = f_3(x) + K' \cdot x^2$. Let $z = d(u,p) = (1+m)x$. We then have $max\_t(u) = f_3(\frac{z}{(1+m)}) + K' \cdot (\frac{z}{(1+m)})^2 = f_3'(z) + \frac{1+m^2}{(1+m)^2} \cdot K \cdot z^2 = f_3'(z) + K'' \cdot z^2$, where $K'' = \frac{1+m^2}{(1+m)^2} \cdot K$ and $f_3'(z)$ is still a piecewise-linear convex function. Similarly, we can prove that $min\_t(u)$ consists of a piecewise-linear concave function of z and the same quadratic term $K'' \cdot z^2$. Therefore, l is a well-behaved segment.

The above arguments can be generalized to show that if L_a and L_b are well-behaved with slope $-1 < m_1 < 1$ (see Fig. 23(b)) and if the delay functions defined over L_a and L_b are functions of x with the same quadratic term $K \cdot x^2$, then any line segment $l \in R$ will be well-behaved. In this case, the skew function defined over l will have quadratic term (i) $\frac{rc}{2}y^2$ if l is vertical, where r and c are per unit resistance and capacitance, or (ii) $K \cdot x^2$ if l is parallel to L_a, or (iii) $\frac{1+(m_1-m_2)^2}{(1+m_1-m_2)^2} \cdot K \cdot x^2$ if l is not vertical and has slope $m_2 \neq m_1$.

By symmetry, we can have the same conclusion for the case where L_a and L_b have slopes $> +1$ or < -1. □

The following Lemmas 8- 9 and Theorem 2 refer to Fig. 10, where joining segments L_a and L_b have slopes > 1 and L_c and L_d are the other two boundary segments of region $R = SDR(L_a, L_b)$.

LEMMA 8. *Let $R = SDR(L_a, L_b)$. (i) If $FMR(R) = \emptyset$, then $MSR(R) = MSR(L_a)$ if $skew(MSR((L_a)) < skew(MSR((L_b))$, and $MSR(R) = MSR(L_b)$*

otherwise. (ii) If $FMR(R) \neq \emptyset$, then the skew changing rate is $+r(Cap(a) + Cap(b) + hc)$ as we traverse from the boundaries of $mr(v)$ horizontally to L_a or L_b.

Proof: Note that Lemma 2 holds under any monotone delay model, including Elmore and pathlength delay model. Then the lemma can be proved in a similar fashion as in Lemma 3, except that by Fact 3, the skew changing rate in regions $R - FMR(R)$ is $+r(Cap(a) + Cap(b) + hc)$ (instead of $+2$ under pathlength delay) as we traverse from the boundaries of $mr(v)$ horizontally to L_a or L_b. □

LEMMA 9. *Let l be a boundary segment of $mr(v)$ in Fig. 10(a). Then there are at most n skew turning points on L_a or L_b, where n is the number of leaf nodes in the subtree T_v rooted at v.*

Proof: We first prove that the linear term of the max-delay function of l is an s-piecewise-linear function with $s \leq n/2 + 1$.

If v is an internal node whose children are sinks, then there can be at most 1 delay turning point on l. So, $s \leq n/2 + 1 = 2$. Otherwise, we assume that for any internal node v with children a and b the linear term of the max-delay functions on any boundary segments of $mr(a)$ and $mr(b)$ are s_1-piecewise-linear and s_2-piecewise-linear functions, where $s_1 \leq n_1/2 + 1$, $s_2 \leq n_2/2 + 1$, and n_1 and n_2 are the numbers of leaf nodes in the subtree rooted at nodes a and b, respectively. Since $mr(v) \in SDR(L_a, L_b)$, by Lemma 6, the linear term of the max-delay function defined over any boundary segment of $mr(v)$ will be an s-piecewise-linear function with $s \leq s_1 + s_2 - 1 \leq n_1/2 + 1 + n_2/2 + 1 - 1 = n/2 + 1$, where $n = n_1 + n_2$ is the number of leaf nodes in the subtree T_v rooted at v.

Similarly, we can prove that the linear term of the min-delay function on any boundary segment of $mr(v)$ will be an s-piecewise-linear function with $s \leq n/2+1$.

Thus, Lemma 6 implies that $skew(p)$ defined over L_a and L_b after merging nodes a and b will be an s-piecewise-linear function, where $s \leq n + 1$. So there can be up to n skew turning points on each of L_a or L_b. □

THEOREM 2. *For a node $v \in G$, the merging region $mr(v)$ computed by the BME construction rules under Elmore delay (i) is consistent with the definition of merging region, (ii) is a well-behaved region with at most $2n + 4$ sides, (iii) can be computed in $O(n)$ time, where n is the number of leaf nodes in T_v.*

Proof: With Fact 3 and Lemmas 6-9, we can obtain a proof similar to that of Theorem 1, except that there are two differences when the joining segments are not parallel Manhattan arcs with constant delays.

—Because the skew and delay changing rates defined over the line segments in $SDR(L_a, L_b)$ are not limited to $+1$, 0, and -1, the boundary segments of $mr(v)$ can have arbitrary slopes. Moreover, the boundary segment of $mr(v)$ which is a Manhattan arc does not necessarily have constant delays and skew.

—By Lemma 9, on L_a and L_b there will be totally at most $2n$ skew turning points, each of which corresponds to a possible vertex of the merging region. Also, there are at most 4 vertices on the other two boundary segments L_c and L_d (see Fig. 10). So $mr(v)$ will have at most $2n + 4$ vertices (sides). Since construction rule BM3 computes $FMR(l)$ for at most $2n$ horizontal line segments l passing through skew turning points, $mr(v)$ is computed in $O(n)$ time. □

A.3 Proofs of Theorem 3

THEOREM 3. *For each* $2 \leq m' \leq m$, $2 \leq k' \leq k \leq m$ *and* $1 \leq i \leq m - m' + 1$, *the minimum error* $err_i[m', k']$ *for the optimal* (m', k')*-sampling solution is:*

$$err_i[m', k'] = \begin{cases} 0 & \text{if } m' = k' \\ \{(min\_skew(R_{i+m'-1}) - min\_skew(R_{i+m'-2})) \times \\ \quad (Cap(R_i) - Cap(R_{i+m'-2}))\} + err_i[m'-1, k'] & \text{if } m' > k' \text{ and } k' = 2 \\ \min_{i < i' \leq m'-k'+i+1} \{err_i[i'-i+1, 2] + \\ \quad err_{i'}[m'-i'+i, k'-1]\} & \text{if } m' > k' \text{ and } k' > 2 \end{cases} \tag{7}$$

Proof: For each $1 \leq i \leq m - m' + 1$:

Case (i) If $m' = k'$, we can select all m' merging regions and therefore $err_i[m', k'] = 0$.

Case (ii) If $m' > k'$ and $k' = 2$, we are forced to retain the region R_i and $R_{i+m'-1}$. Therefore, the optimal error optimal error is the error of the optimal solution $S_i[m'-1, k' = 2]$ where only R_i and $R_{i+m'-2}$ are retained, plus the error incurred for removing $R_{i+m'-2}$ from $\{R_i, R_{i+m'-2}, R_{i+m'-1}\}$ (Fig. 24).

Case (iii) If $m' > k'$ and $k' > 2$, we have to choose another $k' - 2$ regions from $\{R_{i+1}, \cdots, R_{i+m'-2}\}$ besides the two mandatory regions R_i and $R_{i+m'-1}$. Suppose i' is the index of the region after i in the optimal solution $S_i[m', k']$, then the error of the new staircase between the skews $min\_skew(R_i)$ and $min\_skew(R_{i'})$ is given by $err_i[i'-i+1, 2]$ which is computed in Case (ii). Now, we have to select the optimal solution $S_{i'}[m'-i'+i, k'-1]$ from the regions $\{R_{i'}, \cdots, R_{i+m'-1}\}$ (Fig. 25). Note that $R_{i'}$ is retained in both sub-solutions $S_i[i'-i+1, 2]$ and $S_{i'}[m'-i'+i, k'-1]$, and $i < i' \leq m' - k' + i + 1$. Therefore, we iterate i' from $i+1$ to $m' - k' + i + 1$ and compute the optimal error $err_i[m', k']$ to be smallest among all the sums of $err_i[i'-i+1, 2]$ and $err_{i'}[m'-i'+i, k'-1]$. □

References

BOESE, K. D. AND KAHNG, A. B. 1992. Zero-skew clock routing trees with minimum wirelength. In *Proc. IEEE Int. ASIC Conf.* (Sept. 1992). pp. 1.1.1–1.1.5.

BOESE, K. D., KAHNG, A. B., MCCOY, B. A., AND ROBINS, G. 1995. Near-optimal critical sink routing tree constructions. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems 14*, 12 (Dec.), 1417–1436.

BORAH, M., OWENS, R. M., AND IRWIN, M. J. 1994. An edge-based heuristic for Steiner routing. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems 13*, 12 (Dec.), 1563–1568.

CHAO, T.-H., HSU, Y.-C. H., AND HO, J.-M. 1992. Zero skew clock net routing. In *Proc. Design Automation Conf* (1992). pp. 518–523.

CHAO, T.-H., HSU, Y.-C. H., HO, J.-M., BOESE, K. D., AND KAHNG, A. B. 1992. Zero skew clock routing with minimum wirelength. *IEEE Trans. on Circuits and Systems 39*, 11 (Nov.), 799–814.

CHUNG, J. AND CHENG, C.-K. 1994. Skew sensitivity minimization of buffered clock tree. In *Proc. Int. Conf. on Computer Aided Design* (1994). pp. 280–283.

CONG, J., HE, L., KOH, C.-K., AND MADDEN, P. H. 1996. Performance optimization of VLSI interconnect layout. *Integration, the VLSI Journal 21*, 1–94.

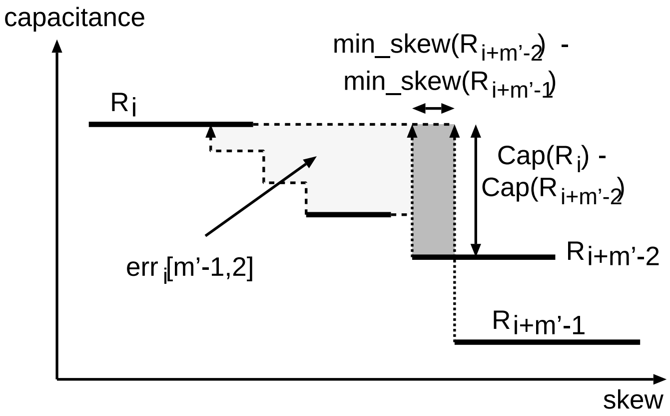


Fig. 24. The lightly shaded region is the area error $err_i[m'-1, k'=2]$ for the optimal solution $S_i[m'-1, k'=2]$, i.e., the error incurred when all intermediate regions between R_i and $R_{i+m'-2}$ are removed. The darker region is the additional error incurred when region $R_{i+m'-2}$ is also removed. It is the skew-capacitance product term in case (ii) of Eqn. (7).

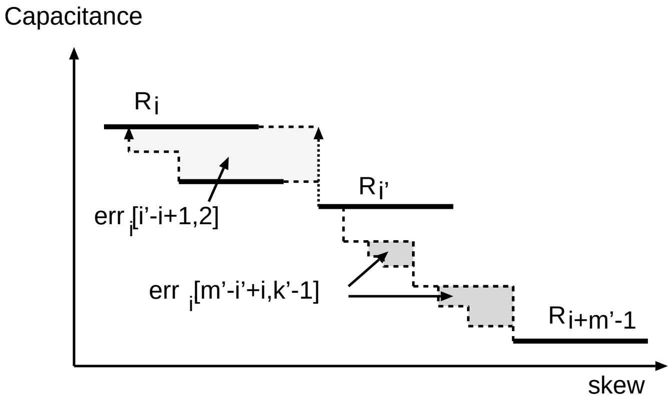


Fig. 25. The lightly shaded region is the area error $err_i[i'-i+1, 2]$ for the optimal solution $S_i[i'-i+1, 2]$, i.e., the error incurred when all intermediate regions between R_i and $R_{i'}$ are removed. The two darker regions account for the error $err_{i'}[m'-i'+i, k'-1]$ of the optimal solution $S_{i'}[m'-i'+i, k'-1]$ in case (iii) of Eqn. (7).

Cong, J., Kahng, A. B., Koh, C.-K., and Tsao, C.-W. A. 1995a. Bounded-skew clock and Steiner routing under Elmore delay. In *Proc. Int. Conf. on Computer Aided Design* (1995). pp. 66–71.

Cong, J., Kahng, A. B., Koh, C.-K., and Tsao, C.-W. A. 1995b. Bounded-skew clock and steiner routing under Elmore delay. Technical Report 950030 (Aug.), UCLA CS Dept.

Cong, J. and Koh, C.-K. 1994. Simultaneous driver and wire sizing for performance and power optimization. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems 2*, 4 (Dec.), 408–423.

Cong, J. and Koh, C.-K. 1995. Minimum-cost bounded-skew clock routing. In *Proc. IEEE Int. Symp. on Circuits and Systems* (April 1995). pp. 1.215–1.218.

CONG, J., KOH, C.-K., AND LEUNG, K.-S. 1996. Simultaneous buffer and wire sizing for performance and power optimization. In *Proc. Int. Symp. on Low Power Electronics and Design* (Aug. 1996). pp. 271–276.

CONG, J. AND LEUNG, K. S. 1995. Optimal wiresizing under the distributed Elmore delay model. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems 14*, 3 (March), 321–336.

EDAHIRO, M. 1991. Minimum skew and minimum path length routing in vlsi layout design. *NEC Research and Development 32*, 4 (Oct.), 569–575.

EDAHIRO, M. 1992. Minimum path-length equi-distant routing. In *Proc. IEEE Asia-Pacific Conf. on Circuits and Systems* (Dec. 1992). pp. 41–46.

EDAHIRO, M. 1993a. A clustering-based optimization algorithm in zero-skew routing. In *Proc. Design Automation Conf* (June 1993). pp. 612–616.

EDAHIRO, M. 1993b. Delay minimization for zero-skew routing. In *Proc. Int. Conf. on Computer Aided Design* (1993). pp. 563–566.

EDAHIRO, M. 1994. An efficient zero-skew routing algorithm. In *Proc. Design Automation Conf* (June 1994). pp. 375–380.

ELMORE, W. C. 1948. The transient response of damped linear networks with particular regard to wide-band amplifiers. *Journal of Applied Physics 19*, 1 (Jan.), 55–63.

FRIEDMAN, E. G., Ed. 1995. *Clock Distribution networks in VLSI Circuits and Systems: A Selected Reprint Volume*. IEEE Circuits and Systems Society.

HUANG, J. H., KAHNG, A. B., AND TSAO, C.-W. A. 1995. On the bounded-skew routing tree problem. In *Proc. Design Automation Conf* (June 1995). pp. 508–513.

KAHNG, A. B. AND ROBINS, G. 1992. A new class of iterative Steiner tree heuristics with good performance. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems 11*, 7 (July), 893–902.

KAHNG, A. B. AND ROBINS, G. 1994. *On Optimal Interconnections for VLSI*. Kluwer Academic Publishers.

KAHNG, A. B. AND TSAO, C.-W. A. 1996. Planar-DME: A single-layer zero-skew clock tree router. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems 15*, 1 (Jan.), 8–19.

KAHNG, A. B. AND TSAO, C.-W. A. 1997a. More practical bounded-skew clock routing. In *Proc. Design Automation Conf* (1997). pp. 594–599.

KAHNG, A. B. AND TSAO, C.-W. A. 1997b. Practical bounded-skew clock routing. *Journal of VLSI Signal Processing (Special issue on High Performance Clock Distribution Networks)*. Special issue on High Performance Clock Distribution Networks, to appear.

LIN, S. AND WONG, C. K. 1994. Process-variation-tolerant clock skew minimization. In *Proc. Int. Conf. on Computer Aided Design* (1994). pp. 284–288.

PULLELA, S., MENEZES, N., AND PILEGGI, L. T. 1996. Post-processing of clock trees via wiresizing and buffering for robust design. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems 15*, 6 (June), 691–701.

TSAY, R.-S. 1993. An exact zero-skew clock routing algorithm. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems CAD-12*, 2 (Feb.), 242–249.

VITTAL, A. AND MAREK-SADOWSKA, M. 1995. Power optimal buffered clock tree design. In *Proc. Design Automation Conf* (June 1995). pp. 497–502.

WANG, T.-C. AND WONG, D. F. 1992. A graph theoretic technique to speed up floorplan area optimization. In *Proc. Design Automation Conf* (1992). pp. 62–68.

XI, J. G. AND DAI, W. W.-M. 1995. Buffer insertion and sizing under process variations for low power clock distribution. In *Proc. Design Automation Conf* (1995). pp. 491–496.

ZHU, Q. AND DAI, W. W.-M. 1996. Planar clock routing for high performance, chip and package co-design. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems 4*, 2 (June), 210–226.

ZHU, Q., DAI, W. W.-M., AND XI, J. G. 1993. Optimal sizing of high-speed clock networks based on distributed RC and lossy transmission line models. In *Proc. Int. Conf. on Computer Aided Design* (1993). pp. 628–633.

This work was partially supported by DARPA/ITO under Contract J-FBI-93-112; National Science Foundation Young Investigator Award MIP-9357582 and grants MIP-9257982 and MIP-922370; grants from Silicon Valley Research, Intel, and Cadence under the 1995 and 1996 California MICRO Programs; Tan Kah Kee Foundation Postgraduate Scholarship and 1995 Design Automation Conference Graduate Scholarship.
Authors' addresses: J. Cong, A. B. Kahng, and C.-K. Koh: Department of Computer Science, University of California, Los Angeles, CA 90095; C.-W. A. Tsao: currently at Cadence Design Systems, Inc. 555 River Oaks Parkway, Bldg. 2, MS 2B1 San Jose, CA 95134.