Bounded-Skew Clock and Steiner Routing Under Elmore Delay*

Jason Cong, Andrew B. Kahng, Cheng-Kok Koh and C.-W. Albert Tsao
Department of Computer Science
University of California, Los Angeles, CA 90024

ABSTRACT: For engineering tradeoffs in "zero-skew" clock tree routing, for performance-driven Steiner routing, and even for power distribution topology design [31], a *bounded-skew routing tree* formulation applies. Recent works [10, 19] have studied this problem under the pathlength (linear) delay model, and generalized the DME clock routing construction [2, 5, 13] via the concept of a *merging region*. However, in practice bounding the pathlength skew does not allow reliable control of actual delay skew, e.g., with respect to the Elmore delay approximation. Thus, in this work we study the minimum-cost bounded-skew routing tree problem under the Elmore delay model. The following are our contributions.

- We prove several key properties of the *merging region under Elmore delay* that is used by our bounded-skew tree construction: (i) the merging region is bounded by *well-behaved segments* on which the skew values are piecewise-linearly increasing, then constant, then piecewise-linearly decreasing; (ii) the merging region is a convex polygon with at most $4n$ sides, where n is the number of leaves in the topology; and (iii) the merging region can be constructed in $O(n)$ time. Note that the "merging region" that we construct is not a complete generalization of the DME merging segment: when detour wiring is needed or when merging regions of subtree roots overlap, our construction may not return a region containing all the minimum-cost merging points.

- We develop a tradeoff approach which minimizes total wirelength for any given skew bound under Elmore delay; we call this the *Boundary Merging and Embedding* (BME) method. This method encompasses extensions of the Extended-DME (Ex-DME, for fixed topology) and Extended-GreedyDME (ExG-DME, for variable topology) methods in [19] from bounded pathlength skew to bounded Elmore delay skew.

- We note that the BME strategy, as well as those in previous works, will utilize merging points that are restricted to the *boundaries* of merging regions. This restriction prevents the skew bound from being fully exploited in the tree cost minimization. We thus develop a new *Interior Merging and Embedding* (IME) algorithm which employs a sampling strategy and dynamic programming to consider merging points that are interior to, rather than on the boundary of, the merging regions.

Experimental results show that our new bounded-skew routing algorithms allow accurate control of Elmore delay skew, and also demonstrate the practical utility of merging points that are strictly interior to the merging regions. Our methods essentially match the best known methods for the extreme cases of zero and infinite skew, and offer smooth cost-skew tradeoffs in between to enable good engineering solutions.

*This work was partially supported by ARPA/CSTO under Contract J-FBI-93-112; NSF MIP-9357582, MIP-9257982 and MIP-922370; a grant from Intel Corporation and a Tan Kah Kee Foundation Postgraduate Scholarship.

1 Introduction

In layout synthesis of high-performance systems, it has become increasingly important to control signal delays, e.g., for clock skew minimization or the timing-driven routing of large global nets. At the same time, routing solutions should have low wiring area to reduce die size and capacitive effects 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 [22] for a detailed review. Recent works have accomplished exact zero skew under the Elmore delay model [30, 6, 15], and have given new methods for single-layer (planar) clock routing [32, 23, 24]. Over the past two years, a number of authors have applied wiresizing optimizations and/or buffer optimizations to minimize phase delay [18, 28, 16, 26, 27, 33, 34], skew sensitivity to process variation [28, 8, 25], and/or power dissipation [28]. The work of [29] developed a clock router that accomplishes specified pin-to-pin delays.

"Exact zero skew" is typically obtained at the expense of increased wiring area and higher power dissipation. In practice, circuits still operate correctly within a given skew tolerance, and indeed exact zero skew is never an actual design requirement [22]. The works of Zhu and Dai [33, 34] and Pullela et al. [27] are notable in that they use initial *non-zero skew* routing solutions which are then wiresized to satisfy a given skew bound. Construction of a minimum-cost *bounded-skew routing tree* (BST) is a key underlying optimization.[1] Given these antecedents, two recent works [10, 19] have addressed the BST problem, and proposed clock and Steiner global routing algorithms that construct BSTs under the linear, i.e., pathlength, delay model. The enabling concept in [10, 19] is that of a *merging region*, which generalizes the merging segment concept of [2, 5, 13] for zero-skew clock trees.

Unfortunately, in practice we find that bounding the pathlength skew does not afford any reliable control of the actual delay skew. Figure 1(a) shows HSPICE delay skew against pathlength delay skew for routing trees generated by the ExG-DME algorithm [19] on the r1-5 benchmark clock sink placements. Not only is the correlation poor, In general, not only is the correlation poor, but the pathlength-based BST solutions of [10, 19] are simply unable to meet tight skew bounds (of $100ps$ or less). On the other hand, Figure 1(b) demonstrates the accuracy and fidelity of Elmore delay skew to actual skew; cf. [3]. While it is possible to "convert", e.g., a zero pathlength skew routing into a zero Elmore delay skew routing, via the use of "snaking" to balance wirelengths [30], such an approach will usually entail over 30% increase in total wirelength. With this in mind, we pursue new approaches to the minimum-cost BST problem under Elmore delay. Our three main contributions are:

- First, we prove that the merging region under Elmore delay is bounded by *well-behaved segments* on which the skew values are piecewise-linearly increasing, then constant, then piecewise-linearly decreasing. We also prove that the merging region is a convex polygon with at most $4n$ sides, where n is the number of leaves 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

[1] Vittal and Marek-Sadowska [31] showed that the minimum-cost BST problem also arises in power distribution topology design.

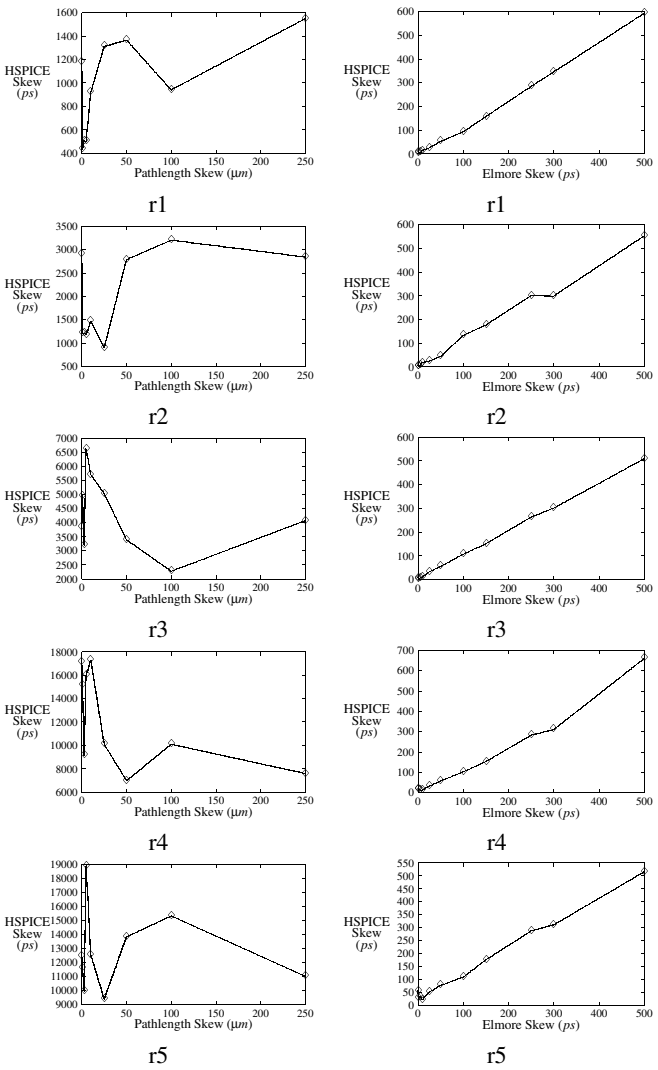


Figure 1: Plots of (a) Pathlength skew versus actual (HSPICE simulation) delay skew for BST solutions obtained by the ExG-DME algorithm [19], and (b) Elmore delay skew versus actual (HSPICE simulation) skew for BST solutions obtained by the IME algorithm for the r1-5 benchmark clock sink placements.

generalization of the DME merging segment: when detour wiring is needed or when merging regions of subtree roots overlap, our construction may not return a region containing all the minimum-cost merging points.

- Second, we develop a tradeoff approach which minimizes total wirelength for any given skew bound under Elmore delay. We call this the *Boundary Merging and Embedding* (BME) method, since it merges boundary points of merging regions in the same manner as in [10, 19]. Using our new construction of the merging region under Elmore delay, the BME method can extend Extended-DME (Ex-DME, for fixed topology) and Extended-GreedyDME (ExG-DME, for variable topology) [19] from bounded pathlength skew to bounded Elmore delay skew.

- Third, we note that the BME strategy, as well as those in previous works, will utilize merging points that are restricted

to the *boundaries* of merging regions. This restriction prevents the skew bound from being fully exploited in the tree cost minimization. We thus develop an improved *Interior Merging and Embedding* (IME) algorithm which employs a sampling strategy and dynamic programming to consider merging points that are interior to, rather than on the boundary of, the merging regions.

The rest of this 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 and bounded-skew routing. In Section 3, we develop our results on the merging region construction under Elmore delay; this construction can be used by BME to replace the merging region construction under pathlength delay. In Section 4, we present our IME algorithm, which uses sampling within the set of feasible merging points based on a dynamic programming approach. Section 5 summarizes experimental results and Section 6 concludes with directions for future work.

2 Preliminaries

Assume that we are given a set $S = \{s_1, s_2, \cdots, s_n\} \subset \mathcal{R}^2$ 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 leaves 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 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 $skew(T) = \max_{s_i,s_j \in S} |t(s_0,s_i) - t(s_0,s_j)|$. Two recent works [10, 19] were the first to address the following problem:

Minimum-Cost Bounded Skew Routing Tree (BST) Problem: Given a set $S = \{s_1, \ldots, s_n\} \subset \mathcal{R}^2$ 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.

In [10, 19], the BST problem was solved under the pathlength delay model, i.e., $t(s_0,s_i) =$ the sum of edgelengths in the unique s_0-s_i path. Given the poor correlation between pathlength delay/skew and actual delay/skew (recall Figure 1), our work addresses the BST problem under Elmore delay, which is defined as follows. For any edge e_v, let r_v and c_v respectively denote its lumped resistance and capacitance. For any node v in G, we use T_v to denote the subtree of T that is rooted at v, and we use C_v to denote the total capacitance of T_v. 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. Then, under the Elmore model the signal delay $t(u,v)$ is given by $t(u,v) = \sum_{e_w \in Path(u,v)} r_w \cdot (\frac{c_w}{2} + C_w)$.

2.1 The DME Approach

The Deferred-Merge Embedding (DME) algorithm, proposed independently in [13, 5, 2], 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 methods of [10, 19] as well as those we propose below are generalizations of DME, we now review the original DME method and the Greedy-DME variant of Edahiro [14], following notation of [2, 6]. 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.

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. 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.

II. The Greedy-DME Algorithm. Note that DME requires an input topology. Several works [2, 5, 14] 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 [14], which determines the topology of the merging tree in a greedy bottom-up fashion. Let K denote a set of merging segments which initially consists of all the sink locations, i.e., $K = \{ms(s_i)\}$. Greedy-DME iteratively finds the pair of nearest neighbors in K, that is, $ms(u)$ and $ms(v)$ such that $d(ms(u), ms(v))$ is minimum. A new parent merging segment $ms(v)$ is computed for node v from a zero-skew merge of $ms(u)$ and $ms(v)$; K is updated by adding $ms(v)$ and deleting both $ms(u)$ and $ms(v)$. After $n-1$ operations, K consists of the merging segment for the root of the topology.

In [15], $O(n \log n)$ time complexity was achieved by finding several nearest-neighbor pairs at once, i.e., the algorithm first constructs a "nearest-neighbor graph" which maintains the nearest neighbor of each merging segment in K. Via zero-skew merges, $|K|/k$ nearest-neighbor pairs are taken from the graph in non-decreasing order of distance, where k 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 [15]). Greedy-DME achieves 20% reduction in wiring cost compared with the methods of [5].

2.2 Previous DME-Based BST Heuristics

In generalizing the DME construction to bounded-skew routing, the enabling concept is that of the *merging region*. Intuitively, for node $v \in G$ with children a and b, the merging region of v corresponds to the set of all locations $l(v)$ at which subtrees T_a and T_b can be joined with minimum wiring cost while still maintaining the skew bound B. In the remainder of this section, we briefly outline some concepts behind the Ex-DME and ExG-DME constructions [10, 19]. We defer most of the ideas to Section 3, which in some sense re-develops and generalizes a number of these ideas within the Elmore delay context.

The works of [10, 19] showed that under the linear delay model, the merging region for any node is a convex polygon bounded by Manhattan arcs (i.e., segments with slope $+1$ or -1) and rectilinear segments (i.e., horizontal or vertical segments). Thus, under pathlength delay the merging region has at most eight sides; furthermore, it can be computed in constant time. For the case where a fixed topology G has been prescribed, the *Extended-DME* (Ex-DME) method essentially parallels the original DME construction and is optimal for the limiting cases of zero skew and infinite skew (i.e., the Steiner minimal tree problem with fixed topology G).

When the topology is variable, [10, 19] propose an *Extended-Greedy-DME* (ExG-DME) method which again uses the merging region construction, and generates a topology via iterated clustering operations analogous to those of Greedy-DME. The ExG-DME method not only provides a smooth skew-cost tradeoff but also closely matches the best known heuristics for zero-skew routing [15]. Notice that in maintaining zero skew, the original DME approach always merges two subtrees at their roots. However, when the skew bound is non-zero, the shortest connection between two subtrees is not necessarily at their roots – and there is no reason that subtrees cannot be merged at non-root nodes as long as the resulting skew is $\leq B$. The ExG-DME version of [19] exploits the flexibility stemming from allowed non-zero skew B by dynamically re-rooting subtrees as they are merged so that the roots of subtrees (after re-rooting) become closer without much change in the subtree costs.

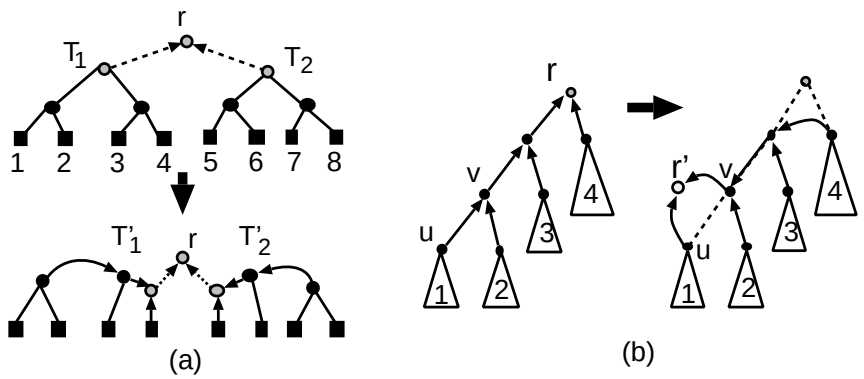


Figure 2: (a) An example showing that given skew bound $B \gg 0$, changing the subtree topology before merging will reduce the merging cost. (b) Repositioning the root in changing the topology.

In Figure 2a, 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_1 and T_2 at their roots as shown in the top example. However, this topology is highly suboptimal when B is large, even if the costs of the two subtrees are minimum. Ideally, for large B the subtree topologies should be modified so that the subtree roots are closer (and merging cost is reduced), as in the bottom example. The ExG-DME method of [19] adjusts a subtree topology by changing the position of the root as illustrated in Figure 2b; the root can be repositioned as the parent of nodes u and v, where u and v are the endpoints of any edge in the current tree. In practice, both the subtree cost and most of the topology remain intact. As a result, this version of ExG-DME also very closely matches the best known heuristics for unbounded-skew routing [4, 21]. The work of [19] also gives an *Extended-Planar-DME* method which again matches the best known heuristics [23] when the routing tree must be embeddable on a single layer. However, when evaluated according to Elmore delay skew, the pleasing "skew-cost tradeoff" properties of Ex-DME and ExG-DME vanish.

3 The Boundary Merging and Embedding Method

We now present our *Boundary Merging and Embedding* (BME) method, which encompasses the straightforward generalizations of Ex-DME and ExG-DME from pathlength to Elmore delay. The difference from [10, 19] lies in the computation of merging regions, i.e., we may execute the previous Ex-DME or ExG-DME strategies to achieve an Elmore delay skew bound simply by substituting the new merging region construction. We do not state a specific "BME algorithm template" since the merging region construction can be used by all variants (fixed-topology, variable-topology, single-layer, etc.).

3.1 Notation and Definitions

Let $t_{max}(p)$ and $t_{min}(p)$ denote the maximum and minimum delay values (max-delay and min-delay, for short) from point p to all leaves in the subtree rooted at p. The skew of point p, denoted $skew(p)$, is $t_{max}(p) - t_{min}(p)$. (If all points of a pointset P have identical max-delay and min-delay, we similarly use the terms $t_{max}(P)$, $t_{min}(P)$ and $skew(P)$.) As p moves along any line segment the values of $t_{max}(p)$ and $t_{min}(p)$, along with $skew(p)$, respectively define the *delay* and *skew functions* over the segment. If B is the given skew bound, then the *slack* of point p, denoted $slack(p)$, is defined to be $B - skew(p)$. A point p with $skew(p) \leq B$ (i.e., non-negative $slack(p)$) is a *feasible merging point*. For any region P, the *feasible merging region* $FMR(P)$ consists of all feasible merging points in P. The *minimum skew region* $MSR(P)$ is the set of points in P with minimum skew.

Well-Behaved Elmore Delay Property. Figure 3 illustrates that Elmore delay values over a horizontal line segment l connecting sinks a and b enjoy a convenient property: for $p \in l$, the delay functions $t_{max}(p)$ and $t_{min}(p)$ are each the sum of a piecewise-linear function and a unique quadratic term $K \cdot x^2$, where $x = d(p,a)$, and $K = \frac{rc}{2}$. To see this, let $h = d(a,b)$ and C_a and C_b be the loading capacitances at sinks a and b. Then we express the max-delay and min-delay from point p to sink

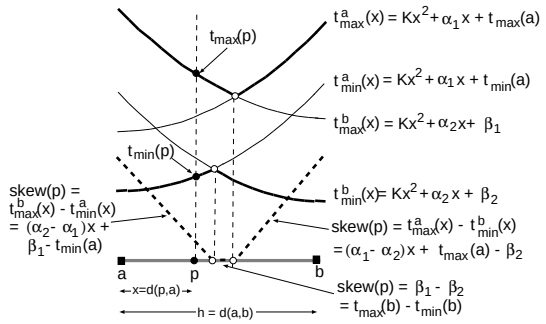


Figure 3: Properties of Elmore delays and skew over straight-line segment l connecting sinks a and b.

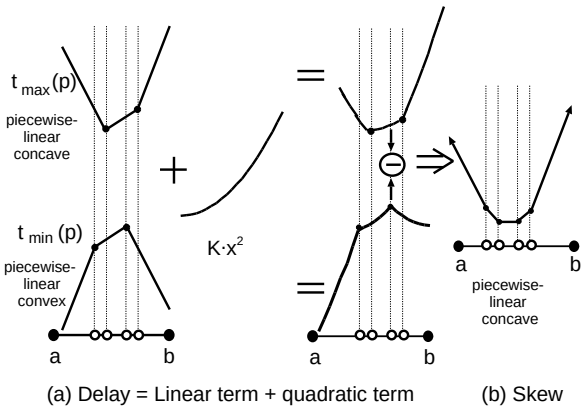


Figure 4: (a) Delay and (b) skew functions for a well-behaved line segment. Skew turning points are indicated by the hollow circles on $\overline{ab}$.

a as functions of $x = d(p,a)$; this corresponds to functions $t^b_{max}(x)$ and $t^a_{max}(x)$ in the Figure, with coefficients $K = rc/2$ and $\alpha_1 = rC_a$. Similarly, we denote the max-delay and min-delay from point p to sink b as $t^b_{max}(x)$ and $t^b_{max}(x)$ in the Figure, with $\alpha_2 = -r(hc + C_b)$, $\beta_1 = Kh^2 + rhC_b + t_{max}(b)$, and $\beta_2 = Kh^2 + rhC_b + t_{min}(b)$. We then have $t_{max}(p) = \max\{t^a_{max}(x), t^b_{max}(x)\}$ $= \max\{\alpha_1 x + t_{max}(a), \alpha_2 x + \beta_1\} + K \cdot x^2$, and $t_{min}(p) = \min\{t^a_{min}(x), t^b_{min}(x)\} = \min\{\alpha_1 x + t_{min}(a), \alpha_2 x + \beta_2\} + K \cdot x^2$. Since $t_{max}(p)$ and $t_{min}(p)$ are each the sum of a piecewise-linear function of x and the same quadratic term $K \cdot x^2$, we have that $skew(p)$ over segment $\overline{ab}$ is a piecewise-linear function of x (with up to 3 linear regions). If sinks a and b are not located on the same horizontal or vertical line segment, we can still prove similarly that this same delay/skew property exists over the boundary and interior segments (with any slopes) of the smallest rectangle containing a and b.

We may formalize this delay/skew property as follows. Let functions $f_1(x) = \max_{i=1,\cdots,n_1}\{\alpha_i \cdot x + \beta_i\}$ and $f_2(x) = \min_{i=1,\cdots,n_2}$ $\{\alpha'_i \cdot x + \beta'_i\}$. Then 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 $t_{max}(p) = f_1(x) + K \cdot x^2$ and $t_{min}(p) = f_2(x) + K \cdot x^2$, where again $x = d(a,p)$. We say that $f_1(x)$ ($f_2(x)$) is an n_1-piecewise-linear *concave* (n_2-piecewise-linear *convex*) 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 (Figure 4). A point $p \in l$ is called a *turning*

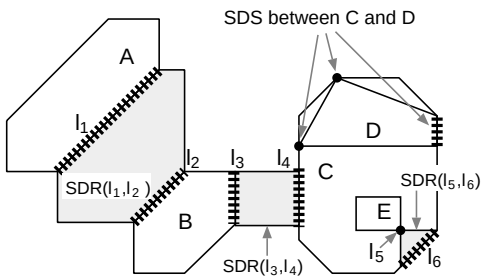


Figure 5: Examples of Shortest Distance Segments (shown as thick dashed lines or solid dots), and Shortest Distance Regions (shown as dotted regions). Note that there are three pairs of Shortest Distance Segments between the overlapping polygons C and D.

point if the slope of a piecewise-linear function defined over l changes at p. Lemma 1 in the Appendix shows that $skew(p)$ defined over the line segment l will be an n-piecewise-linear concave function, i.e., $skew(p) = \max_{i=1,\cdots,n} \{\tilde{\alpha}_i \cdot x + \tilde{\beta}_i\}$ such that (i) $n \leq n_1 + n_2 - 1$, and (ii) each skew turning point of l corresponds to one of the max-delay or min-delay turning points. A region P is a *well-behaved region* if it is a convex polygon whose boundary and interior segments are all well-behaved. Notice that given a well-behaved line segment l, computing $FMR(l)$ and $MSR(l)$ takes time linear in the number of skew turning points of l.

Shortest Distance Segments/Regions. To construct a merging region with minimum merging cost, we begin by defining two more terms. First, for any two convex polygonal regions P and Q with boundaries $\partial(P)$ and $\partial(Q)$, the *shortest distance segments* between P and Q, denoted $SDS(P)$ and $SDS(Q)$, are a pair of straight line segments on $\partial(P)$ and $\partial(Q)$ which are closest to each other. Note that when P and Q overlap, there could be many pairs of shortest distance segments between P and Q (see Figure 5). For simplicity, we will arbitrarily take one pair of them as $SDS(P)$ and $SDS(Q)$.[2] Given any two line segments l_1 and l_2, the *shortest distance region* between l_1 and l_2, denoted $SDR(l_1, l_2)$, is the set of points which have minimum sum of Manhattan distances to l_1 and l_2. We will construct each merging region within the corresponding shortest distance region.

Joining Segments. In the Appendix, we prove in Theorem 1 that each merging region that we construct under the Elmore model is well-behaved. Thus, for any merging regions P and Q, $l_1 = SDS(P)$ and $l_2 = SDS(Q)$ must be well-behaved segments. However, $SDR(l_1, l_2)$ will not be a well-behaved region if either

Case I. l_1 and l_2 are parallel Manhattan arcs with non-constant delay functions, or

Case II. The delay functions on l_1 and l_2 have different quadratic terms.

In our method, the delays of each point in $SDR(l_1, l_2)$ can be determined by any pair of points (p, q) with $p \in l_1$ and $q \in l_2$, so long as $d(p, q) = d(l_1, l_2)$. In Case I, the delay value defined for a point in $SDR(l_1,l_2)$ will not be unique, and line segments

[2] If we allow multiple pairs of shortest distance segments between two polygons, then there will be more than one merging region for each node in the given topology. The IME method in Section 4 will discuss how to deal with this case.

within $SDR(l_1,l_2)$ will not even be well-behaved in Case II. To guarantee well-behaved regions, we define the *joining segments* $JS(P) \in l_1$ and $JS(Q) \in l_2$ as follows. (The merging regions will be constructed within $SDR(JS(P), JS(Q))$.)

- If Case I or II holds, then $JS(P)$ and $JS(Q)$ are chosen to be (arbitrary) single points on l_1 and l_2, respectively.
- Otherwise, $JS(P) = l_1$ and $JS(Q) = l_2$.

Merging Regions. Finally, 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 $mr(s_i) = \{s_i\}$.
- 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 within $SDR(L_a, L_b)$ such that (i) the difference in Elmore delays from v to any pair of sinks in T_v is within the skew bound, and (ii) 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 $d(p,q) = d(L_a, L_b)$.

We can show that the merging region under this definition always exists. Because of the constraints that we have incorporated within the construction of merging regions, the "merging region" that we construct does not necessarily contain all feasible merging points having minimum merging cost $|e_a| + |e_b|$.

3.2 Construction of the Merging Regions

Given merging regions $mr(a)$ and $mr(b)$ of v's children, the following rules can be used to construct the new merging region $mr(v)$. The construction rules for merging regions are similar to those presented in [19], except that the joining segments can be parallel segments with any slopes, i.e., besides Manhattan arcs and rectilinear line segments (see Figure 6).

M0 Compute joining segments L_a and L_b on the boundaries of $mr(a)$ and $mr(b)$.

M1 Compute delay and skew functions and then $FMR(l)$ for the boundary segments l of $SDR(L_a, L_b)$.

M2 If L_a and L_b are parallel segments with slopes < -1 or $> +1$, compute $FMR(l)$ for each horizontal 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 Figure 6b). Similarly, if L_a and L_b are parallel segments with slopes between $+1$ and -1, compute $FMR(l)$ for each vertical line segment $l = \overline{pq}$ such $p \in L_a$, $q \in L_b$, and either p or q is a skew turning point of L_a or L_b.

M3 Let F be the set of FMRs computed via rules **M1** and **M2**. If $F \neq \emptyset$, then $mr(v)$ is equal to the smallest convex polygonal region containing F. Otherwise, $mr(v) = MSR(L_a)$ if $skew(MSR(L_a)) \leq skew(MSR(L_b))$, and $mr(v) = MSR(L_b)$

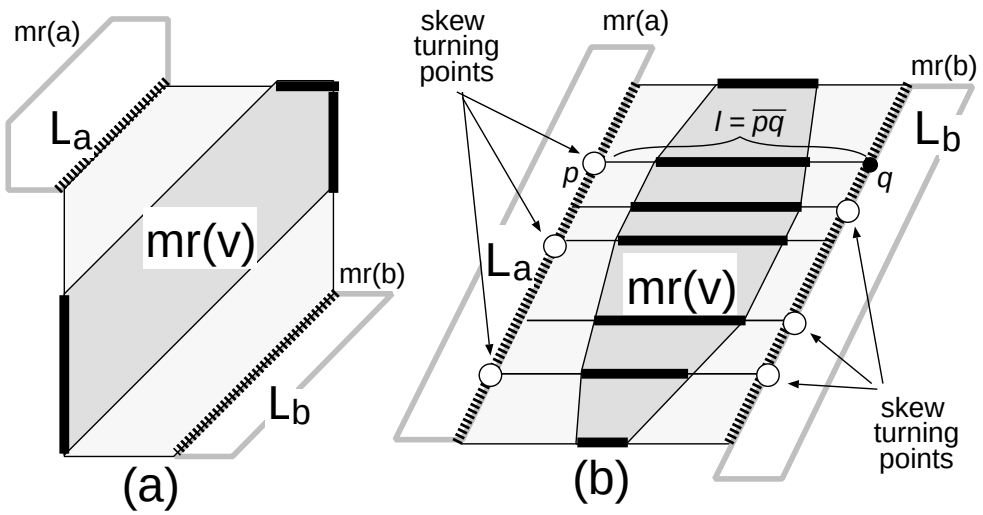


Figure 6: Construction of merging region $mr(v)$ given the merging regions for v's children a and b. The joining segments L_a and L_b, shown as thick dashed lines, are parallel Manhattan arcs (with constant max-delay and min-delay) in (a) and parallel line segments in (b). Note that merging region $mr(v)$ (heavily shaded) is constructed inside $SDR(L_a, L_b)$ (lightly shaded). Each thick solid line represents the $FMR(l)$ for a line segments l such that l is either the boundary segments of $SDR(L_a, L_b)$ or the horizontal line segments incident to the skew turning points (shown as hollow circles) on L_a or L_b.

otherwise.[3]

Similar to the proof of Theorem 1 in [19], we can show that each merging region $mr(v)$ for a node v in topology G is a well-behaved region. Lemma 5 in the Appendix shows that the delay functions on the boundary segments of $mr(v)$ will consist of m-piecewise-linear functions and a quadratic term, where $m \leq n$ (n is the number of leaves in the subtree T_v rooted at node v). Based on this lemma, we can show that $mr(v)$ has at most $4n$ sides. We thus have the following theorem. The detailed proof is given in the Appendix.

Theorem 1 *Under the Elmore delay model, each merging region $mr(v)$ for a node $v \in G$ is a well-behaved region with at most $4n$ sides, and can be computed by the above construction rules in $O(n)$ time, where n is the number of leaves in T_v.*[4] □

It is easy to see that for node $v \in G$ with children a and b, if (i) $mr(a)$ and $mr(b)$ do not overlap, (ii) $JS(mr(a)) = SDS(mr(a))$ and $JS(mr(b)) = SDS(mr(b))$, and (iii) no detour wiring is needed, then $mr(v)$ will contain all points having minimum merging cost. In practice, the above condition holds for most nodes in a good routing topology. Thus, for example, in the zero-skew case $mr(v)$ is equivalent to the merging segment constructed by the original DME method if no detour wiring is needed. Thus, we can expect that the performance of our method will be very close to that of DME.

[3] In the case where $F = \emptyset$, minimum merging cost $d(L_a, L_b)$ cannot meet the skew bound. Thus detour wiring is needed to satisfy the skew bound. To minimize the detour wiring, $mr(v)$ must be $MSR(SDR(L_a, L_b))$, which is shown to be either $MSR(L_a)$ or $MSR(L_b)$ in Lemma 4 in the Appendix.

[4] In all our experiments, no merging region has more than 9 sides. Thus, each merging region is in practice computed in constant time.

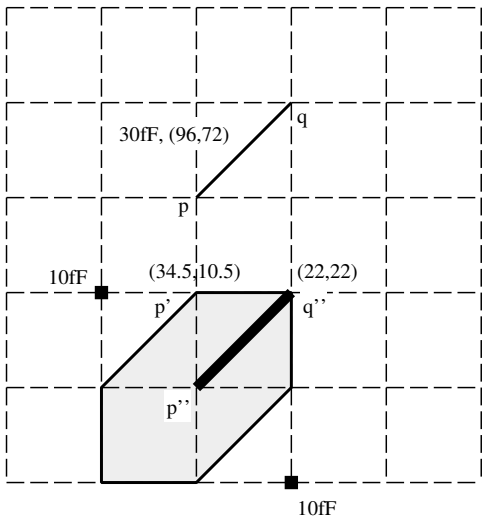


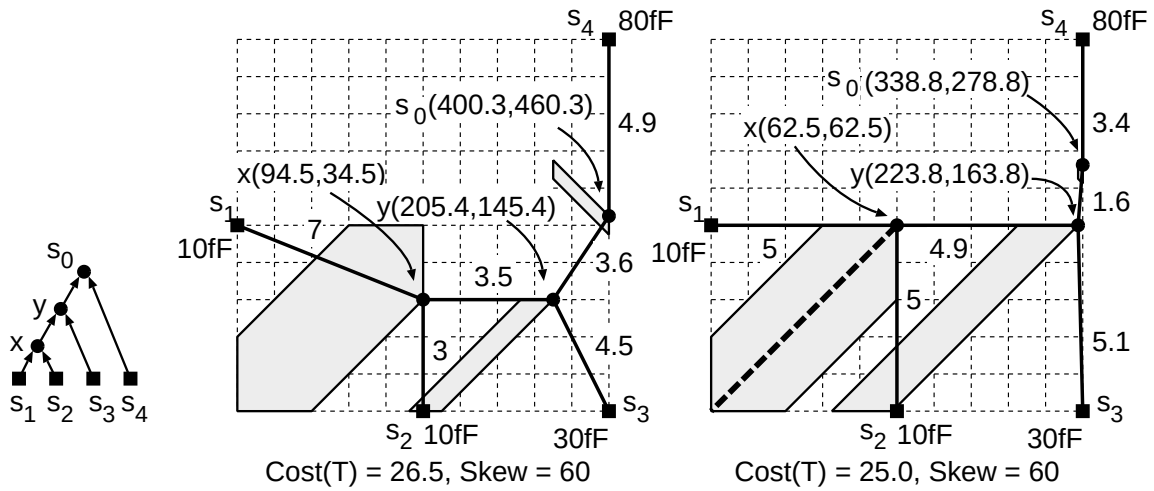
Figure 7: The Manhattan arc pq has a capacitance of $30fF$. It is to be merged with the merging regions defined by two sinks of capacitance $10fF$ each. Each pair of coordinates associated with a point p or segment $\overline{pq}$ represents its max-delay and min-delay, respectively. Merging of p and p' with a skew bound of 24 units requires a merging cost of ≈ 2.44 units and a detour is necessary. Merging of $\overline{pq}$ with $\overline{p''q''}$ under the same skew bound does not require detour and has a merging cost of 2 units, a 22% reduction of merging cost. We assume that an unit length wire has unit resistance and unit capacitance.

When the skew bound $B = \infty$, then the merging regions constructed under the Elmore delay model will be the same rectangles that are constructed under the pathlength delay model. Thus, the performance of ExG-DME under both the pathlength and Elmore delay models will be the same. [19] reported that the Steiner trees constructed by ExG-DME average only 0.21% higher cost than those constructed by one of the best known Steiner tree heuristics [4].

4 The Interior Merging and Embedding Method

As outlined in the previous section, the construction of a merging region is based on the *boundary* segments of its children's merging regions: no interior points of the child merging regions are used to construct the new merging regions. This is, of course, also the case for the pathlength-based BST algorithms in [10, 19]. However, such an approach produces a sub-optimal merging cost when detouring occurs during the merging. For example, in Figure 7, merging of points p and p' requires a detour and an additional merging cost of ≈ 0.44 unit when compared to the merging of segments $\overline{pq}$ with $\overline{p''q''}$. Therefore, it is possible to reduce the merging cost if we consider merging of interior points. Furthermore, even if no detour is required, it is not always advisable to use only boundary segments for merging. For example, instead of fully utilizing the available slack at the bottom level (which is the case of the BME method) as in Figure 8(b), we can conserve the slack by merging interior points and thus reduce the merging cost at a higher level (Figure 8(c)).

Given the above considerations, merging of interior points of the merging regions has strong potential to reduce total wire-



(a) Topology (b) Using only boundary segments for merging (c) Using samples for merging

Figure 8: An example of routing 4 sinks (filled squares) with a skew bound of 60 units. The merging regions are indicated by shaded regions. Each internal node (filled circle) is embedded in its merging 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 ≈ 26.5 units if only joining segments are considered for merging. (c) We can however reduce merging cost by merging interior points in order to save slack for higher level use. We assume that an unit length wire has unit wire resistance and capacitance.

length. However, merging interior points may cause ambiguity in the delay functions for a point p in the new merging region: this point may be merging point for 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 algorithm) 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 employs a sampling strategy and dynamic programming to consider merging points which are interior to, rather than on the boundary of, the merging region.

In the remainder of this section, we describe the use of samples from the merging regions to construct the next layer of merging regions. We focus our attention on the use of Manhattan arc to sample the merging regions, i.e. we represent each merging region by a set of Manhattan arcs, some of which are in the interior of the merging region. However, the idea can be extended easily to handle sampling segments which are horizontal and vertical. In fact, we can have a mixture of any type of sampling segments for a merging region as long as they are well-behaved.

4.1 Overview of IME Algorithm

The IME algorithm does bottom-up construction of the merging regions at each internal node, similar to the BME algorithm presented in the previous section. The key difference is that in the IME algorithm, each node v in the topology G is associated

with a set $\mathcal{R} = \{R_1, R_2, \ldots, R_k\}$ of merging regions. 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. We denote the *sampling set* of R_i by $SS(R_i)$. The sampling set of $\mathcal{R}$, denoted $SS(\mathcal{R})$, is $\bigcup_{i=1}^{k} SS(R_i)$. As mentioned, we focus our attention on the use of Manhattan arcs to sample the merging regions. The slope of the Manhattan arcs is chosen to $+1$ or -1 such that the points on the Manhattan arc in the merging region have constant max-delay and min-delay.

Consider the merging of two nodes u and v in G. Let $\mathcal{R}_u$ and $\mathcal{R}_v$ be the set of merging regions associated with u and v, respectively. The parent of u and v in G has as many as $|SS(\mathcal{R}_u)| \times |SS(\mathcal{R}_v)|$ possible merging regions due to the merging of each sampling segment in $SS(\mathcal{R}_u)$ with each sampling segment in $SS(\mathcal{R}_v)$. Since the sampling segments are all Manhattan arcs, the merging operation is straightforward, as discussed earlier.

Generally speaking, given a merging region, there are infinite number 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 other words, 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 at the sibling node. When we merge two sampling set, each with $\leq k \cdot s$ sampling segments, we use the dynamic programming method to compute the best possible sampling regions for the new node, again using no more than k regions. A key step in the IME algorithm lies in choosing the "best" k merging regions for the new node.

In what follows, we require the following terminology. A merging region R is associated with three values: (i) $Cap(R)$, the total capacitance in the subtree rooted at R, (ii) $min_skew(R)$, the minimum skew possible within the merging region, and (iii) $max_skew(R)$ the maximum skew possible within the merging region. In the IME algorithm, merging regions are constructed in such a way that (i) $Cap(R) = Cap(ss_a) + Cap(ss_b) + c_a + c_b$ is constant for all points in R, which is constructed by merging two sampling segments, say ss_a and ss_b, of its children a and b with merging cost $|e_a| + |e_b|$, respectively, and (ii) $max_skew(R)$ is kept within the skew bound B. 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.

Consider a node v in G with a set of more than k merging regions associated with it after merging its two children. A merging region R of v is said to be *"redundant"* if there exists another merging region R' of v such that $min_skew(R') < min_skew(R)$

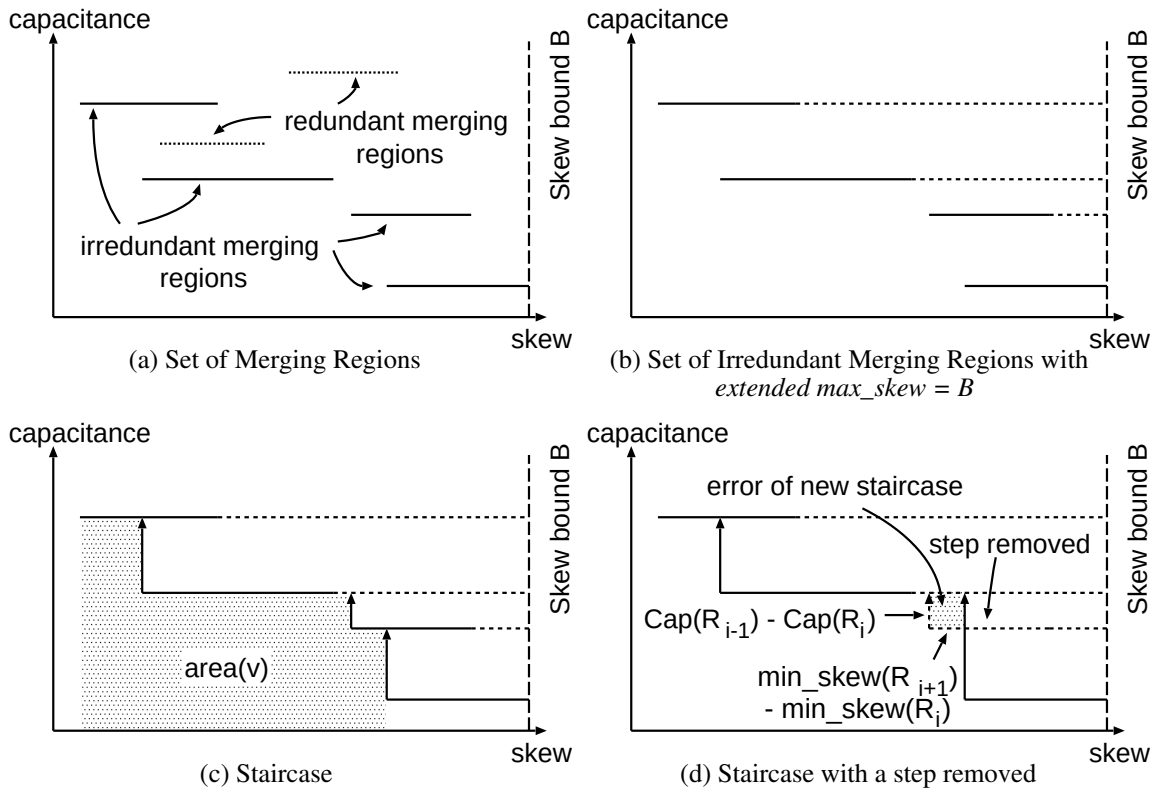


Figure 9: (a) Set of merging regions. (b) Set of irredundant merging regions with extended max-skew = B. (c) Forming a staircase from $min_skew(R_i)$ to merging region R_{i-1}. (d) Removing an intermediate step results in a new staircase with an error depicted by the shaded region.

and $Cap(R') < Cap(R)$ (See Figure 9(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 $min_skew(R_i) < min_skew(R_{i+1})$ for all i with $1 \leq i < m$.

The set of irredundant merging regions forms a *staircase* with $m-1$ steps as shown in Figure 9. First, we extend the line segment representing such irredundant merging region such that it touches the vertical line representing skew bound B (Figure 9(b)). Hence, each irredundant merging region has an *extended* max-skew equals to B, the skew bound. By creating a *step* from $min_skew(R_i)$ at a height of $Cap(R_i)$ to $Cap(R_{i-1})$ for all i with $1 < 1 \leq m$, we have a $m-1$ step staircase starting at $min_skew(R_m)$ as shown in Figure 9(c).

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 $min_skew(R_k)$:

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

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 error $(min_skew(R_{i+1}) - min_skew(R_i)) \times (Cap(R_{i-1}) - Cap(R_i))$ (Figure 9(d)). Therefore, in order to

retain a good spectrum of no more than k merging regions at each step, we solve the following problem:

The Optimal (m,k)-Sampling Problem: Given a set of m irredundant merging regions, $IMR = \{R_1, \ldots, R_m\}$, 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 $IMR' = \{R_{\pi(1)} = R_1, R_{\pi(2)}, \ldots, R_{\pi(k-1)}, R_{\pi(k)} = R_m\} \subset IMR$ 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'.

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

We developed an optimal algorithm to the (m,k)-sampling problem using the dynamic programming approach. 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 $\{R_i, R_{i+1}, \ldots, R_{i+m'-1}\}$ 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 2 *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:*

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

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 $\{R_i, R_{i+m'-2}, R_{i+m'-1}\}$ (Figure 10).

Case (iii) If $m' > k'$ and $k' > 2$, we have to choose another $k'-2$ regions from $\{R_{i+1}, \ldots, R_{i+m'-2}\}$ 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 $\{R_{i'}, \ldots, R_{i+m'-1}\}$ (Figure 11). 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 $err_{i'}[m'-i'+i,k'-1]$. □

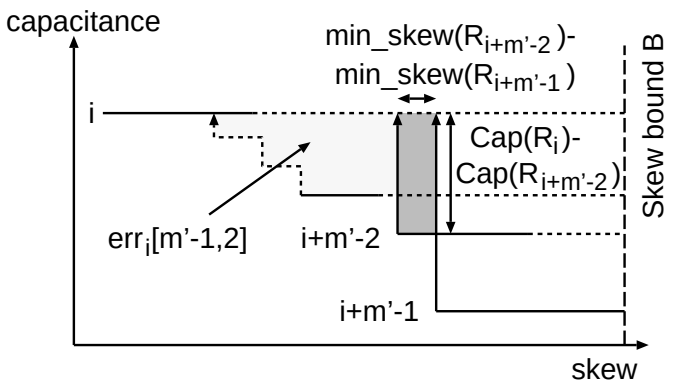


Figure 10: The lightly shaded region is the area error for the optimal solution $S_i[m'-1,k'=2]$ and the darker region is the skew-capacitance product term in case (ii) of Eqn. (2).

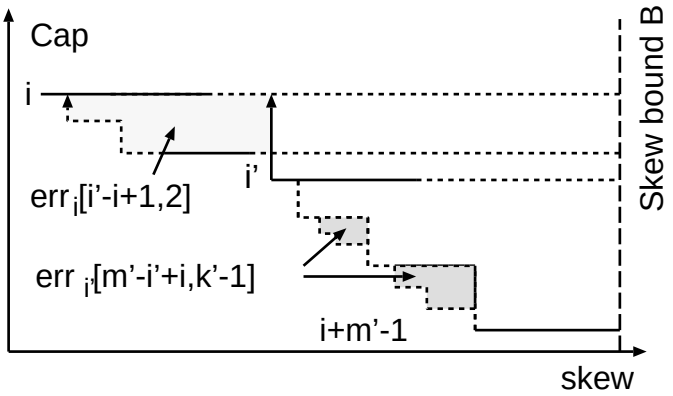


Figure 11: The lightly shaded region is the area error for the optimal solution $S_i[i'-i+1,2]$ and the two darker regions account for the error of the optimal solution $S_{i'}[m'-i'+i,k'-1]$ in case (iii) of Eqn. (2).

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. (2), the solution is straightforward and we can compute all $S_i[m',k']$ and $err_i[m',k']$ where $1 \le i \le m-m'+1$, and m' and k' satisfy the conditions stated in case (i) and (ii) of Eqn. (2) 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 $1 < i \le m-k+2$. Assuming that $k-1 > 2$, we again apply case (iii) of Eqn. (2) 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 $i < i' \le m-k+2$. 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

Optimal (m,k)-Sampling Algorithm

for each k', i s.t. $2 \le k' \le k$, $1 \le i \le m-k'+1$
  $err_i[k',k'] \leftarrow 0$;
  $next_i[k',k'] \leftarrow i+1$;

for each m', i s.t. $2 \le m' \le m$, $1 \le i \le m-m'+1$
  $err_i[m',2] \leftarrow (min-skew(R_{i+m'-1}) - min-skew(R_{i+m'-2})) \times$
    $(Cap(R_i) - Cap(R_{i+m'-2})) + err_i[m'-1,2]$;
  $next_i[m',2] \leftarrow i+m'-1$;

for each k' s.t. $2 < k' \le k$
  for each i s.t. $k-k'+1 \le i \le m-k'+1$
    $i' \leftarrow i+1$;
    $err_i[m-i+1,k'] \leftarrow err_i[i'-i+1,2] + err'_i[m-i'+1,k'-1]$;
    $next_i[m-i+1,k'] \leftarrow i'$;
    for each i' s.t. $i+1 < i' \le m-k'+1$
      if $err_i[i'-i+1,2] + err_{i'}[m-i'+1,k'-1] < err_i[m-i+1,k']$,
        $err_i[m-i+1,k'] \leftarrow err_i[i'-i+1,2] + err_{i'}[m-i'+i,k'-1]$;
        $next_i[m-i+1,k'] \leftarrow i'$;

$i \leftarrow 1$;
$m' \leftarrow m$;
$S_1[m,k] \leftarrow \{R_i\}$;
for $k' \leftarrow k$ **downto** 2 **do**
  $i \leftarrow next_i[m',k']$;
  $m' \leftarrow m'-i+1$;
  $S_1[m,k] \leftarrow S_1[m,k] \cup \{R_i\}$;

Figure 12: The optimal (m,k)-Sampling Algorithm. Note that $m \ge k$.

in its respective range ($2 \le m' \le m$, $2 \le k' \le k$ and $1 \le i \le m-m'+1$). Instead, only errors $err_i[m',k']$ with $m' = m-i+1$ are required.

The Optimal (m,k)-Sampling Algorithm is given in Figure 12. The matrix $next_i[m',k']$ in the algorithm records the index of the mergion region immediately after R_i in the optimal subset (of size k') of the set $\{R_i, R_{i+1}, \ldots, R_{i+m'-1}\}$. We first initialize $err_i[m',k']$ and $next_i[m',k']$ for the first two cases of Eqn. (2). Then, for each k' and i such that $2 < k' \le k$ and $k-k'+1 \le i \le m-k'+1$, we apply case (iii) of Eqn. (2) to compute the minimum error $err_i[m-i+1,k']$. Therefore, we have the following result:

Theorem 3 *The time complexity of the optimal (m,k)-Sampling Algorithm is $O(k \cdot m^2)$.* □

4.3 Summary of the IME Algorithm

To summarize, the IME algorithm still retains the flow of other DME-based algorithms, namely compute a tree of merging regions in a bottom-up fashion and perform embedding in a top-down manner. The key difference between the IME and other Boundary-Based algorithm is that a set of no more than k irredundant merging regions are associated with each node and most importantly, some of these merging regions are due to merging of interior points.

In our implementation, a sampling set for each node is obtained by representing each irredundant merging region using a set of s parallel Manhattan arcs. The slope of the Manhattan arcs is chosen such that max-delays and min-delays for any two

IME Merging Procedure

Let u and v be the children of w;
$SS(\mathcal{R}_u) \leftarrow$ Sample $\mathcal{R}_u$ using $\leq s$ Manhattan arcs;
$SS(\mathcal{R}_v) \leftarrow$ Sample $\mathcal{R}_v$ using $\leq s$ Manhattan arcs;
Let $\mathcal{R}_w$ be the set of merging regions of w, initially $\emptyset$;
For each segment from $SS(\mathcal{R}_u)$ and each segment from $SS(\mathcal{R}_v)$
    Merge to produce a merging region of w; Put in $\mathcal{R}_w$;
Sort $\mathcal{R}_w$ in descending order of region capacitance of w;
Scan $\mathcal{R}_w$ in one pass to weed out "redundant" merging regions;
Apply Optimal (m,k)-Sampling Algorithm on IMR of w if necessary;

Figure 13: Outline for the merging operation of the IME algorithm.

points on the Manhattan arc in the merging region are the same. Therefore, the merging operation is performed on Manhattan arcs which is straightforward as illustrated in Figure 6.

We then sort the resultant merging regions in descending order of their capacitance. Redundant regions are weeded out and the resultant irredundant merging regions are sampled by the optimal (m,k)-sampling algorithm. The outline of the merging process is given in Figure 13. 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 complexity of the clock tree construction algorithm is still in the same order.

5 Experimental Results

We have implemented the BME and IME algorithms in ANSI C on Sun SPARC-20 machines. The benchmark test cases r1–r5 [30] were used to evaluate our algorithms for skew bounds in the range of 0–10ns. Table 1 compares the zero-skew clock routing costs of the best known algorithm (CL+I6 from [15]) with the various bounded-skew routing costs obtained by our algorithms. Also included in the Table is the CPU time for both BME and IME.

In this experiment, the IME algorithm keeps at most $k = 5$ merging regions for each internal node, and slices each merging region to $s = 7$ sampling segments for merging with other nodes. Both of our methods have comparable results. It appears that IME produces better results for larger circuits when the skew bound is large, but at the expense of longer running time. In general, we see a decrease in total wirelength as the skew increases. However, our results do not compare favorably with [15] when it comes to zero skew. 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 [15] used the best result from eight different values of the k parameter ranging from 2 to 4 (recall the discussion of the Greedy-DME algorithm in Section 2.1), while we only set one value of the k parameter of [15] to be 1.

We alo compare the performance of BME and IME using the same topologies, which are generated by BME and IME respectively. The results are shown in Tables 2 and 3. For the zero-skew bound, IME is equivalent to DME while BME is equivalent to

Skew Bound (ps)	r1	r2	r3	r4	r5
	Wirelengths (CPU time: hr:min:sec)				
0 (CL+I6 [15])	1253347	2483754	3193801	6499660	9723726
0(BME)	1307637 (00:00:12)	2647476 (00:00:28)	3344828 (00:00:46)	6934030 (00:01:44)	11066897(00:02:43)
(IME)	1445555 (00:00:10)	2907593 (00:00:47)	3605778 (00:01:29)	7255455 (00:04:39)	*10706940(00:11:05)
1(BME)	1223125 (00:01:48)	2397494 (00:04:05)	3427426 (00:05:28)	6415233 (00:18:32)	9470000 (00:22:08)
(IME)	1274819 (00:03:30)	2526961 (00:09:33)	*3060284 (00:14:54)	6751435 (00:40:20)	9896655 (01:08:39)
10(BME)	1087703 (00:02:16)	2155481 (00:05:11)	2789321 (00:08:19)	5411936 (00:22:26)	8140187 (00:29:25)
(IME)	1112512 (00:06:06)	2353202 (00:17:33)	*2727299 (00:28:44)	*5350241 (01:18:07)	*8065110 (02:45:30)
100(BME)	926205 (00:03:02)	2201023 (00:07:40)	2515178 (00:12:51)	4860958 (00:26:22)	7375204 (00:37:12)
(IME)	930426 (00:13:18)	*1978378 (00:37:30)	*2366874 (01:01:34)	4953715 (03:09:16)	*7003996 (06:00:21)
1000(BME)	793498 (00:03:26)	1839666 (00:08:03)	2506399 (00:12:29)	4971769 (00:29:25)	7134697 (00:41:58)
(IME)	861561 (00:25:24)	1995665 (01:13:03)	*2097784 (02:06:22)	*4740962 (06:41:31)	*6280007 (13:42:13)
10000(BME)	780100 (00:03:38)	1668872 (00:09:10)	2102182 (00:14:26)	4017261 (00:32:20)	6136574 (00:47:29)
(IME)	790285 (00:32:30)	*1574153 (02:07:21)	*1998007 (03:12:15)	4326234 (10:30:53)	*5912472 (24:39:45)

Table 1: The cost-skew tradeoff and running time of the BME and IME algorithms for benchmark circuits r1-5 [30]. We mark the cases where IME outperforms BME by *.

DME only if no detour wiring is required.[5] Thus, IME is expected to have better performance than BME. However, for non-zero skew bounds, IME will not compete favorably with BME unless IME uses a large sampling set (i.e., at the expense of larger CPU time). For instance, Table 2 shows that BME outperforms IME for non-zero skew bounds when IME uses $k = 5$ and $s = 7$. On the other hand, Table 3 shows that when $s = 25$, IME can closely match the BME performance, albeit with much longer CPU time.

Skew Bound(ps)	r1	r2	r3	r4	r5
	Wirelengths (CPU time: min:sec)				
0(BME)	1445555 (00:01)	*2908184 (00:02)	3605778 (00:04)	*7256852 (00:13)	*10708710(00:27)
(IME)	(00:01)	(00:01)	(00:01)	(00:01)	(00:02)
1(BME)	1273637 (00:01)	2518172 (00:03)	3053668 (00:05)	6727448 (00:13)	9859749 (00:38)
(IME)	(00:17)	(00:43)	(00:59)	(02:17)	(03:48)
10(BME)	1099628 (00:01)	2329842 (00:03)	2697450 (00:05)	5294651 (00:13)	7987811 (00:35)
(IME)	(00:16)	(00:37)	(00:53)	(01:51)	(03:01)
100(BME)	913598 (00:01)	1949013 (00:03)	2326170 (00:05)	4864974 (00:13)	6879762 (00:34)
(IME)	(00:11)	(00:26)	(00:37)	(01:54)	(02:17)
1000(BME)	*904989 (00:01)	1966189 (00:03)	2056433 (00:05)	4657888 (00:13)	6159917 (00:35)
(IME)	(00:12)	(00:22)	(00:33)	(01:45)	(02:01)
10000(BME)	775249 (00:01)	1540561 (00:03)	1958586 (00:04)	4245122 (00:13)	5793859 (00:35)
(IME)	(00:09)	(00:48)	(00:33)	(01:11)	(01:55)

Table 2: The performance comparison between BME and IME using the topologies generated by IME. Total wirelengths of IME for different skew bounds are in Table 1. BME results are better than those of IME in most cases except where marked by *.

A more detailed experiment for 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). The results of IME for the benchmark circuits r1–r5 are shown in Figure 14. 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

[5] It is easy to modify the construction rules **M0-M3** so that BME = DME when skew bound $B = 0$. For example, for each node v with children a and b, if $mr(a)$ and $mr(b)$ are Manhattan arcs with constant delays and skew = B, then $mr(v)$ will be a merging segment and can be constructed using the DME method.

Skew Bound(ps)	r1	r2	r3	r4	r5
	Wirelengths (CPU time: hr:min:sec)				
0(BME)	(00:00:01)	(00:00:02)	(00:00:04)	(00:00:10)	(00:00:21)
(IME)	*1307145 (00:00:01)	*2634443 (00:00:01)	*3338992 (00:00:01)	*6933519 (00:00:01)	*11058780(00:00:02)
1(BME)	(00:00:01)	(00:00:03)	(00:00:05)	(00:00:14)	(00:00:26)
(IME)	1225157 (00:13:34)	2530710 (00:30:10)	*3088219 (00:42:34)	*6238926 (01:42:40)	9659309 (02:35:11)
10(BME)	(00:00:01)	(00:00:03)	(00:00:04)	(00:00:12)	(00:00:25)
(IME)	1137216 (00:16:58)	2220246 (00:30:31)	*2744910 (00:35:19)	5718161 (01:13:25)	8247660 (01:56:17)
100(BME)	(00:00:01)	(00:00:03)	(00:00:04)	(00:00:13)	(00:00:25)
(IME)	954209 (00:07:28)	*2009243 (00:15:26)	*2445907 (00:24:58)	5325425 (00:54:11)	*7137548 (01:21:31)
1000(BME)	(00:00:01)	(00:00:03)	(00:00:04)	(00:00:12)	(00:00:25)
(IME)	809191 (00:06:46)	*1706973 (00:13:33)	2956226 (00:18:50)	*4750333 (00:49:12)	*6414957 (01:09:22)
10000(BME)	(00:00:01)	(00:00:03)	(00:00:04)	(00:00:12)	(00:00:24)
(IME)	792955 (00:05:23)	1685845 (00:13:26)	2123887 (00:18:20)	4067494 (00:44:40)	6201162 (01:05:30)

Table 3: The performance comparison between BME and IME using the same topologies generated by BME. The total wirelengths of BME for different skew bounds are given in Table 1. We mark the cases where IME outperforms BME by *.

when compared to the best reported zero-skew solutions (by the CL+I6 algorithm in [15]).

6 Conclusion and Future Work

In conclusion, we have presented new bounded-skew routing tree approaches under the Elmore delay model. We prove several key properties of the merging regions under the Elmore delay model. 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.

Recall from Section 4 that the IME algorithm can handle more general sampling segments other than Manhattan arcs. As our current implementation uses only Manhattan arcs for sampling, future work includes extending the IME approach to include well-behaved sampling segments other than Manhattan arcs. We are also studying better sampling strategies for speed-up of the IME method. Our final goal is to combine the techniques of BST topology generation with our recent work on optimal sizing of interconnects and drivers [9, 11], and develop a practical clock routing algorithm which carries out simultaneous topology generation, buffer insertion, and wiresizing to achieve bounded skew with minimum power dissipation under various layout constraints.

References

[1] H. Bakoglu, J. T. Walker and J. D. Meindl, "A Symmetric Clock-Distribution Tree and Optimized High-Speed Interconnections for Reduced Clock Skew in ULSI and WSI Circuits," *Proc. IEEE ICCD*, Port Chester, 1986, pp. 118–122.

[2] K. D. Boese and A. B. Kahng, "Zero-Skew Clock Routing Trees With Minimum Wirelength", *Proc. IEEE 5th Int'l ASIC Conf.*, Rochester, September 1992, pp. 1.1.1 - 1.1.5.

[3] K. D. Boese, A. B. Kahng, B. A. McCoy and G. Robins, "Near-Optimal Critical Sink Routing Tree Constructions," to appear in *IEEE Trans. on CAD*, 1995.

[4] M. Borah, R. M. Owens and M. J. Irwin, "An Edge-Based Heuristic for Rectilinear Steiner Trees", *IEEE Trans. on Computer-Aided Design*, Dec. 1994, pp. 1563-1568.

[5] T.-H. Chao, Y.-C. Hsu and J.-M. Ho, "Zero Skew Clock Net Routing," *Proc. ACM/IEEE Design Automation Conf.*, 1992, pp. 518-523.

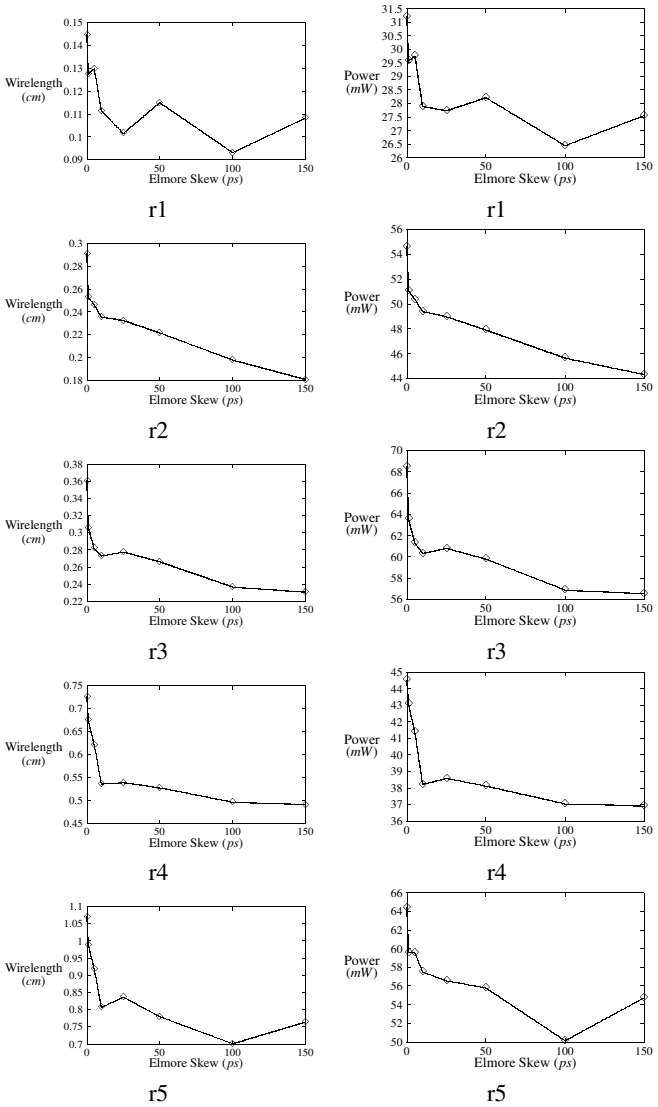


Figure 14: Tradeoff between (a) total wirelength and skew bound, and (b) power dissipation and skew bound.

[6] T.-H. Chao, Y.-C. Hsu, J. M. Ho, K. D. Boese and A. B. Kahng, "Zero Skew Clock Routing With Minimum Wirelength," *IEEE Trans. on Circuits and Systems* 39(11), Nov. 1992, pp. 799–814.

[7] J. D. Cho and M. Sarrafzadeh, "A Buffer Distribution Algorithm for High-Speed Clock Routing," *Proc. ACM/IEEE Design Automation Conf.*, 1993, pp. 537–543.

[8] J. Chung and C.-K. Cheng, "Skew Sensitivity Minimization of Buffered Clock Tree," *Proc. IEEE Int'l Conf. on Computer-Aided Design*, 1994, pp. 280–283.

[9] J. Cong and C.-K. Koh, "Simultaneous Driver and Wire Sizing for Performance and Power Optimization," *IEEE Trans. on VLSI Systems*, Dec. 1994, pp. 408-425.

[10] J. Cong and C.-K. Koh, "Minimum-Cost Bounded-Skew Clock Routing," *Proc. IEEE Int'l Symp. on Circuits and Systems*, Apr. 1995, vol. 1, pp. 215–218. Also available as *UCLA Computer Science Department Technical Report #950003*, Jan. 1995.

[11] J. Cong and K.-S. Leung, "Optimal Wiresizing Under Elmore Delay Model," *IEEE Trans. on Computer-Aided Design*, Mar. 1995, pp. 321-336.

[12] J. Cong, A. B. Kahng and G. Robins, "Matching-Based Methods for High-Performance Clock Routing," *IEEE Trans. on Computer-Aided Design*, 12(8), Aug. 1993, pp. 1157–1169.

[13] M. Edahiro, "Minimum Skew and Minimum Path Length Routing in VLSI Layout Design," *NEC Research and Development*, 32(4), Oct. 1991, pp. 569–575.

[14] M. Edahiro, "Minimum Path-Length Equi-Distant Routing," *Proc. IEEE Asia-Pacific Conf. on Circuits and Systems*, Dec. 1992, pp. 41–46.

[15] M. Edahiro, "A Clustering-Based Optimization Algorithm in Zero-Skew Routing," *Proc. ACM/IEEE Design Automation Conf.*, Jun. 1993, pp. 612–616.

[16] M. Edahiro, "Delay Minimization for Zero-Skew Routing", *Proc. IEEE Int'l Conf. on Computer-Aided Design*, 1993, pp. 563-566.

[17] E. G. Friedman, "Clock Distribution Design in VLSI Circuits - An Overview", *Proc. IEEE Int'l Symp. on Circuits and Systems*, 1993, pp. 1475-1478.

[18] L. P. P. P. van Ginneken, "Buffer Placement in Distributed RC-Tree Networks for Minimal Elmore Delay,", *Proc. Int'l Symposium on Circuits and Systems*, 1990, pp 865–868.

[19] J. H. Huang, A. B. Kahng and C.-W. A. Tsao, "On the Bounded-Skew Routing Tree Problem", *Proc. ACM/IEEE Design Automation Conf.*, San Francisco, Jun. 1995. pp. 508-513. Also available as *UCLA Computer Science Department Technical Report #940026x*, Nov., 1994.

[20] M. A. B. Jackson, A. Srinivasan and E. S. Kuh, "Clock Routing for High Performance ICs," *Proc. ACM/IEEE Design Automation Conf.*, 1990, pp. 573–579.

[21] A. B. Kahng and G. Robins, "A New Class of Iterative Steiner Tree Heuristics with Good Performance", *IEEE Transactions on Computer-Aided Design*, 11(7), July 1992, pp. 893-902.

[22] A. B. Kahng and G. Robins, *On Optimal Interconnections for VLSI*, Kluwer Academic Publishers, 1994.

[23] A. B. Kahng and C.-W. A. Tsao, "Planar-DME: Improved Planar Zero-Skew Clock Routing with Minimum Pathlength Delay," *Proc. European Design Automation Conf.*, 1994, pp. 440-445.

[24] A. B. Kahng and C.-W. A. Tsao, "Low-Cost Single-Layer Clock Trees with Exact Zero Elmore Delay Skew," *Proc. IEEE Int'l Conf. on Computer-Aided Design*, 1994, pp. 213–218.

[25] S. Lin and C. K. Wong, "Process-Variation-Tolerant Clock Skew Minimization," *Proc. IEEE Int'l Conf. on Computer-Aided Design*, 1994, pp. 284–288.

[26] N. Menezes, S. Pullela and L. T. Pillage, "Skew Reduction in Clock Trees Using Wire Width Optimization", *Proc. IEEE Custom Integrated Circuits Conf.*, 1993.

[27] S. Pullela, N. Menezes and L. T. Pillage, "Reliable Non-Zero Skew Clock Tree Using Wire Width Optimization", *Proc. ACM/IEEE Design Automation Conf.*, 1993, pp. 165-170.

[28] S. Pullela, N. Menezes, J. Omar and L. Pillage, "Skew and Delay Optimization for Reliable Buffered Clock Trees," *Proc. IEEE Int'l Conf. on Computer-Aided Design*, 1993, pp. 556–562.

[29] M. Seki, K. Inoue, K. Kato, K. Tsurusaki, S. Fukasawa, H. Sasaki and M. Aizawa, "A Specified Delay Accomplishing Clock Router Using Multiple Layers," *Proc. IEEE Int'l Conf. on Computer-Aided Design*, 1994, pp. 289–292.

[30] R. S. Tsay, "Exact Zero Skew," *Proc. IEEE Int'l Conf. on Computer-Aided Design*, 1991, pp. 336-339.

[31] A. Vittal and M. Marek-Sadowska, "Power Distribution Topology Design", *Proc. ACM/IEEE Design Automation Conf.*, San Francisco, June 1995, pp. 503–507.

[32] Q. Zhu and W. W.-M. Dai, "Perfect-balance planar clock routing with minimal path-length," *IEEE Int'l Conf. on Computer-Aided Design*, 1992, pp. 473–476.

[33] Q. Zhu and W.M. Dai, "Delay Bounded Minimum Steiner Tree Algorithms for Performance-Driven Routing", *UCSC Technical Report UCSC-CRL-93-46*, Oct. 10, 1993.

[34] Q. Zhu and W.M. Dai, manuscript, 1994.

Appendix: Proof of Theorem 1

Lemma 1 *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 concave, then $\max\{f_1(x), f_2(x)\}$ is an n-piecewise-linear concave function, where $n \leq m_1 + m_2$. (ii) If both $f_1(x)$ and $f_2(x)$ are convex, then $\min\{f_1(x), f_2(x)\}$ is an n-piecewise-linear convex function, where $n \leq m_1 + m_2$. (iii) If $f_1(x)$ is concave and $f_2(x)$ is convex, then $f_1(x) - f_2(x)$ is also an n-piecewise-linear concave function, but $n \leq m_1 + m_2 - 1$.*

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 concave function, where $n \leq m_1 + m_2 - 1$. □

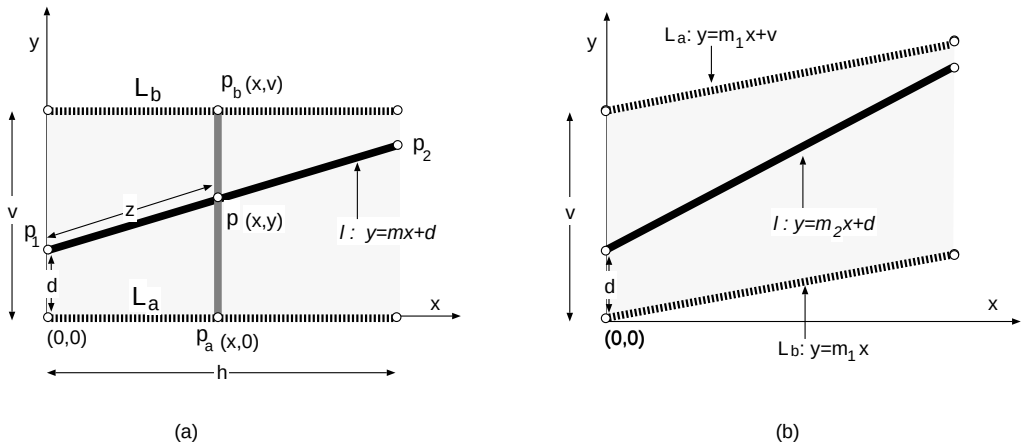


Figure 15: Any line segment $l \in SDR(L_a, L_b)$ (the dotted region) is well-behaved if L_a and L_b are well-behaved segments and the delay functions defined over L_a and L_b have the same quadratic term.

Lemma 2 *Let L_a and L_b be two parallel well-behaved segments and $R = SDR(L_a, L_b)$. Then any line segment $l \in R$ is well-behaved if the delay functions defined over L_a and L_b have the same quadratic term. Furthermore, if the linear terms of the max-delay (or min-delay) functions defined over L_a and L_b are respectively s_1-piecewise-linear and s_2-piecewise-linear, then the linear term of the max-delay (or min-delay) function defined over l will be s-piecewise-linear with $s \leq s_1 + s_2$.*

Proof: To simplify the discussion, we first assume that L_a and L_b are two horizontal line segments, as shown in Figure 15a. The same arguments as in the 2-sink example in Figure 3 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 $l = \overline{p_1 p_2} \in R$ is well-behaved. As shown in the Figure, line segment l is described by the equation $y = mx + d$ where $m \neq \infty$, $0 \leq x \leq h$, and d is a constant. Also, for any $p = (x, y) \in l$ we define $p_a = (x, 0) \in L_a$ and $p_b = (x, v) \in L_b$.

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 $t_{max}(p_a) = \max_{i=0,\cdots,s_1}\{m_i x + d_i\} + K \cdot x^2$ for point $p_a = (x,0) \in L_a$, and $t_{max}(p_b) = \max_{i=0,\cdots,s_2}\{n_i x + e_i\} + K \cdot x^2$ for point $p_b = (x,v) \in L_b$. For point $p = (x,y) \in l$, we have $t_{max}(p) = \max\{A(x,y), B(x,y)\}$ with

$$
\begin{aligned}
A(x,y) &= K \cdot y^2 + \alpha_1 y + t_{max}(p_a) \qquad \text{where } \alpha_1 = rC_a \\
&= K \cdot (mx+d)^2 + \alpha_1(mx+d) + \max_{i=0,\cdots,s_1}\{m_i x + d_i\} + K \cdot x^2 \\
&= m'x + d' + \max_{i=0,\cdots,s_1}\{m_i x + d_i\} + K' \cdot x^2 \qquad \text{where } K' = (1+m^2)K,\ m' = 2mdK + \alpha_1 m, \text{ and } d' = K \cdot d^2 + \alpha_1 d \\
&= \max_{i=0,\cdots,s_1}\{m'_i x + d'_i\} + K' \cdot x^2 \qquad \text{where } m'_i = m_i + m' \text{ and } d'_i = d_i + d' \\
&= f_1(x) + K' \cdot x^2 \qquad \text{where } f_1(x) = \max_{i=0,\cdots,s_1}\{m'_i x + d'_i\}
\end{aligned}
$$

$$
\begin{aligned}
B(x,y) &= K \cdot y^2 + \alpha_2 y + \gamma + t_{max}(p_b) \qquad \text{where } v = d(L_a, L_b),\ \alpha_2 = -r(cv + C_b) \text{ and } \gamma = K \cdot v^2 + rvC_b. \\
&= K \cdot (mx+d)^2 + \alpha_2(mx+d) + \gamma + \max_{i=0,\cdots,s_2}\{n_i x + e_i\} + K \cdot x^2 \\
&= n'x + e' + \max_{i=0,\cdots,s_2}\{n_i x + e_i\} + K' \cdot x^2 \qquad \text{where } n' = 2mdK + \alpha_2 m \text{ and } e' = K \cdot d^2 + \alpha_2 d + \gamma \\
&= \max_{i=0,\cdots,s_2}\{n'_i x + e'_i\} + K' \cdot x^2 \qquad \text{where } n'_i = n_i + n' \text{ and } e'_i = e_i + e' \\
&= f_2(x) + K' \cdot x^2 \qquad \text{where } f_2(x) = \max_{i=0,\cdots,s_2}\{n'_i x + e'_i\}.
\end{aligned}
$$

Therefore, $t_{max}(p) = \max\{f_1(x), f_2(x)\} + K' \cdot x^2 = f_3(x) + K' \cdot x^2$, where by Lemma 1 $f_3(x)$ is an s-piecewise-linear concave function of x with $s \le s_1 + s_2$. Let $z = d(p, p_1) = mx$. We then have $t_{max}(p) = f_3(z/m) + K' \cdot (z/m)^2 = f'_3(z) + (1 + \frac{1}{m^2})K \cdot z^2$, where $f'_3(z)$ is still a piecewise-linear concave function. Similarly, $t_{min}(p)$ consists of a piecewise-linear convex function of x (or z) and the same quadratic term $K' \cdot x^2$ (or $(1 + \frac{1}{m^2})K \cdot z^2$). Therefore, l is a well-behaved segment.

The above argument generalizes to show that if L_a and L_b are well-behaved with slope $-1 < m_1 < 1$ (see Figure 15b) 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) $(1 + (m_1 - m_2)^2)K' \cdot x^2$ 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 3 and 4 and Theorem 1 refer to Figure 16. As illustrated in the Figure, we define a *strictly monotone region* to be a region where skew values strictly linearly increase at a constant rate as we traverse horizontally from the boundary of $mr(v)$ to L_a or L_b.

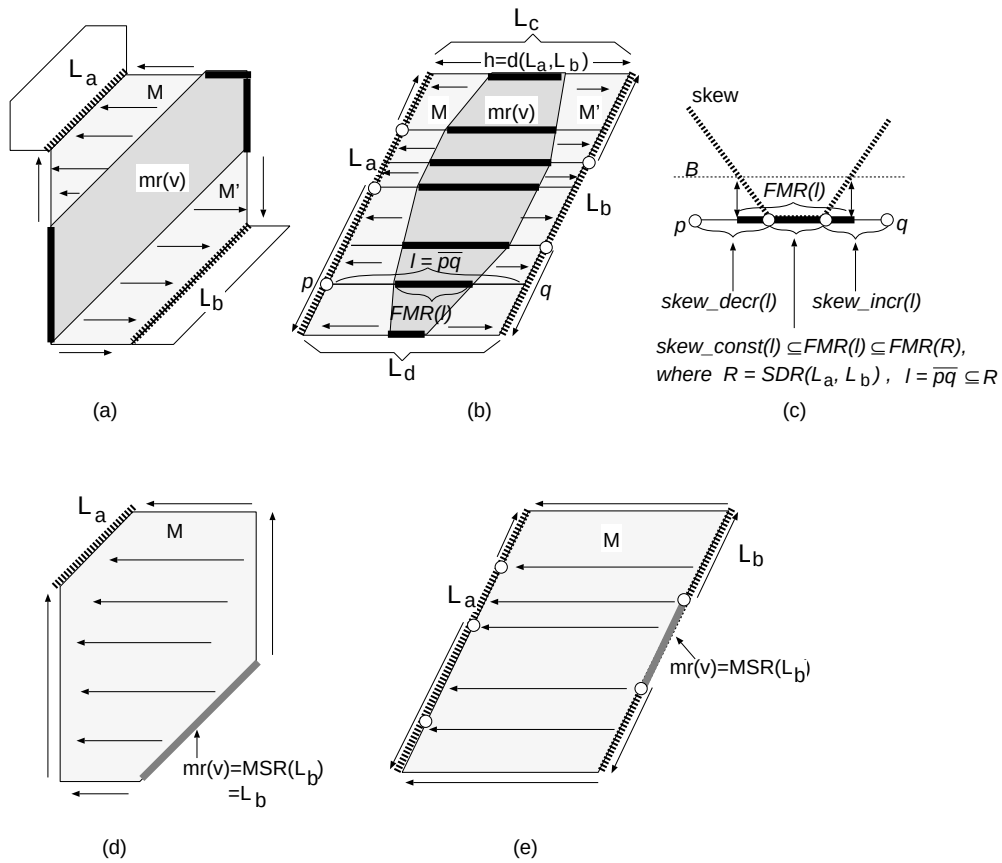


Figure 16: In (a) and (b), $SDR(L_a, L_b)$ is divided by $mr(v)$ (the heavily dotted areas) into two strictly monotone regions M and M' (the lightly dotted areas). Arrows indicate the direction of increasing skew within the strictly monotone regions and on the boundary of $SDR(L_a, L_b)$. In (d) and (e), $mr(v) = MSR(L_b)$ since the skew values monotone increase from right to left. The hollow points in (b) and (e) are the skew turning points of L_a and L_b. Each thick solid line in (b) represents the $FMR(l)$ for a line segment l such that l is either a boundary segment of $SDR(L_a, L_b)$ or a horizontal line segment incident to a skew turning point on L_a or L_b.

Lemma 3 *Any horizontal line segments $l \in SDR(L_a, L_b)$ 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 $-r(C_a + C_b + hc)$, 0, and $+r(C_a + C_b + hc)$, with $h = d(L_a, L_b)$ and C_a and C_b being the total capacitance of the subtrees rooted at nodes a and b.*

Proof: This follows from the discussion of the 2-sink example in Figure 3, except that sinks a and b are replaced by the subtrees rooted at the endpoints of l. □

Lemma 4 *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 $mr(v) = FMR(R) \neq \emptyset$, then R will be divided by $mr(v)$ into at most two strictly monotone regions, where the skew changing rate is $+r(C_a + C_b + hc)$ as we traverse from the boundary of $mr(v)$ to L_a or L_b horizontally.*

Proof: Let l be a horizontal line segment in R. By Lemma 3, we have the same result of Lemma 2 in [19]; that is, $skew_const(l) \subseteq FMR(l) \subseteq FMR(R)$ (Figure 16c). If $FMR(R) = \emptyset$, we must have $skew_const(l) = \emptyset$. Thus, R will contain only $skew_incr(l)$ or $skew_decr(l)$, and therefore will be a single *strictly monotone region*. Therefore, $MSR(R) = MSR(L_a)$ if skew values increase along any horizontal line from L_a to L_b, 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.

On the other hand, if $mr(v) = FMR(R) \neq \emptyset$, then $skew_const(l) \in FMR(R) = mr(v)$. Therefore, R will be divided by $mr(v)$ into at most two strictly monotone regions. □

Lemma 5 *For any node $v \in G$ with merging region $mr(v)$, the linear term of the max-delay (or min-delay) function on any boundary segment of $mr(v)$ will be an s-piecewise-linear function, where $s \leq n$ and n is the number of leaves in the subtree T_v rooted at v.*

Proof: If node v is a sink, then $s = 0 \leq 1$.

Otherwise, for any internal node v with children a, and b, assume 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 $s_1 \leq n_1$, $s_2 \leq n_2$, and n_1 and n_2 are the numbers of leaves in the subtree rooted at nodes a and b respectively. By Lemma 2, the linear term of the max-delay function defined over any line segment within $SDR(L_a, L_b)$ will be an s-piecewise-linear function with $s \leq s_1 + s_2 \leq n_1 + n_2 = n$. Note that n is equal to the number of leaves in the subtree T_v rooted at v. Since $mr(v) \in SDR(L_a, L_b)$, the linear terms of the max-delay function defined over any boundary segment of $mr(v)$ will be an s-piecewise-linear function with $s \leq s_1 + s_2 \leq n_1 + n_2 = n$.

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 $s \leq n$. □

Theorem 1 *Under the Elmore delay model, each merging region $mr(v)$ for a node $v \in G$ is a well-behaved region with at most $4n$ sides, and can be computed by construction rules $M0 - M3$ in $O(n)$ time, where n is the number of leaves in T_v.*

Proof: The proof is similar in structure to that in [19], which used the well-behaved properties of L_a and L_b to prove that under the pathlength delay model each merging region is a convex polygonal region bounded by Manhattan arcs and rectilinear line segments.

It is easy to show that if L_a and L_b are Manhattan arcs with constant delay values, then the resulting merging area will be a convex region with at most 6 boundary segments which are either Manhattan arcs or rectilinear line segments. Since $mr(v) \in SDR(L_a, L_b)$, Lemma 2 implies that any boundary and interior segments of $mr(v)$ are well-behaved. Thus, $mr(v)$ is well-behaved for the case in Figure 16a.

We next consider the case in Figure 16b, where we assume (without loss of generality) that the slopes of L_a and L_b are

greater than $+1$. If $FMR(R) = \emptyset$, then $mr(v) = MSR(L_a)$ or $MSR(L_b)$, which can be computed in constant time. After adding detour wiring, $mr(v)$ will be a line segment with constant delays and skew $= B$, and therefore is still a well-behaved region. If $FMR(R) \neq \emptyset$, then by Lemma 4, we know $mr(v) = FMR(R)$ divides R into at most two strictly monotone regions, where the skew increases at a constant rate $+r(C_a + C_b + hc)$ as we traverse from the boundary of $mr(v)$ horizontally to L_a or L_b. Based on (i) $skew(l) = B$ where l is a boundary segment between $mr(v)$ and strictly monotone region M or M'; and (ii) the skew functions along L_a and L_b are both piecewise-linear concave, we can easily see that (iii) $mr(v)$ must be a convex polygon, and (iv) the vertices of $mr(v)$ lie either on the boundary segments of R or opposite the skew turning points of L_a or L_b. By Lemma 2, any boundary and interior segments of $mr(v)$ are well-behaved. Thus, $mr(v)$ is a well-behaved region and can be constructed by rules M0-M3.

Finally, we show that $mr(v)$ has at most $4n$ sides and can be computed in $O(n)$ time. By Lemma 5, the linear terms of the delay functions defined over L_a and L_b will be s_1-piecewise-linear and s_2-piecewise-linear functions, where $s_1 \leq n_1$, $s_2 \leq n_2$, and n_1 and n_2 are the number of leaves in the subtree rooted at node a and b, respectively. By Lemma 2, after merging nodes a and b the linear terms of the new delay functions defined over L_a or L_b will be s-piecewise-linear functions, where $s \leq n_1 + n_2 = n$. Thus, Lemma 1 implies that $skew(p)$ defined over L_a and L_b after merging nodes a and b will be an m-piecewise-linear function, where $m \leq 2n-1$; that is, $skew(p)$ has at most $2n-2$ skew turning points. Therefore, on L_a and L_b there will be at most $(4n-4)$ skew turning points, each of which corresponds to a possible vertex of the merging region. As shown in Figure 16b, there are at most 4 vertices on the other two boundary segments L_c and L_d. So $mr(v)$ will have at most $4n$ vertices (sides). Consequently, construction rule M2 computes feasible merging regions for at most $4n-4$ horizontal line segments passing through skew turning points. Since construction rule M1 takes only constant time, $mr(v)$ is computed in $O(n)$ time. □