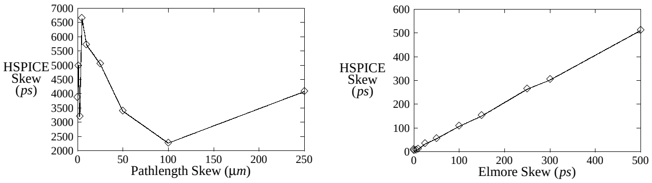
Bounded-Skew Clock and Steiner Routing Under Elmore Delay[*]

Jason Cong, Andrew B. Kahng, Cheng-Kok Koh and C.-W. Albert Tsao
UCLA Dept. of Computer Science, Los Angeles, CA 90095-1596

Abstract: We study the minimum-cost bounded-skew routing tree problem under the Elmore delay model. We present two approaches to construct bounded-skew routing trees: (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 dynamic programming to consider merging points that are interior to, rather than on the boundary of, the merging regions. Our new algorithms allow accurate control of Elmore delay skew, and show the utility of merging points inside merging regions.

1 Introduction

Control of signal delay is important in layout synthesis of high performance systems. Recent works on clock routing have accomplished exact zero skew under the Elmore delay model [17, 5, 11], and given new single-layer (planar) constructions [18, 15, 16]. A detailed review of clock tree and Steiner routing algorithms is given in [14]. In practice, circuits will operate correctly within a given skew tolerance, and indeed "exact zero skew" is never an actual design requirement [14]. Two recent works [8, 12] have addressed the *bounded-skew routing tree* (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 [8, 12] is that of a *merging region*, which generalizes the merging segment concept of [1, 4, 10] for zero-skew clock trees.



(a) (b)

Figure 1: Plots of (a) Pathlength skew versus actual (HSPICE simulation) skew for BST solutions obtained by the ExG-DME algorithm [12] and (b) Elmore delay skew versus actual (HSPICE simulation) skew for BST solutions obtained by the IME algorithm presented below, for the r3 benchmark instance.

In practice, bounding the pathlength skew does not reliably control the actual skew. Figure 1(a) plots HSPICE delay skew against pathlength delay skew for routing trees generated by the ExG-DME algorithm [12] on the r3 benchmark instance. In general, not only is the correlation poor, but the pathlength-based BST solutions of [8, 12] 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. [2]. Converting a zero pathlength skew routing into a zero Elmore delay skew routing via the use of "snaking" [17] usually entails substantial increase in total wirelength. Thus, in this work, we pursue new approaches which directly address the BST problem under Elmore delay. Our three main contributions are:

(i) We prove that the merging region under Elmore delay is a convex polygon bounded by at most 4n *well-behaved segments*, where n is the number of leaves in the tree topology; this region can be constructed in linear time.

(ii) We develop the *Boundary Merging and Embedding* (BME) method, which merges points on the boundaries of merging regions in the same manner as in [8, 12].

(iii) We develop a more general *Interior Merging and Embedding* (IME) algorithm which employs a sampling strategy and the dynamic programming technique to consider merging points that are *interior* to, rather than on the boundary of, the merging regions.

Due to space restrictions, proofs of all lemmas and theorems are given in the technical report [6].

2 Preliminaries

Assume that we are given a set $S = \{s_1, s_2, \cdots, s_n\} \subset R^2$ 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 [8, 12] were the first to address the following problem:

Minimum-Cost Bounded Skew Routing Tree (BST) Problem: Given a set $S = \{s_1, \ldots, s_n\} \subset R^2$ 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 variant with prescribed topology G. We do not include clock source location s_0 in our formulation, but our methods transparently accommodate any prescribed s_0.

In [8, 12], 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. We now address the BST problem under Elmore delay, which is computed as follows. Let r and c denote the resistance and capacitance per unit wirelength, respectively. For any edge e_v, let $r_v = |e_v| \cdot r$ and $c_v = |e_v| \cdot c$ respectively denote its lumped resistance and lumped 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 u-v path. Then, Elmore delay between nodes u and v is given by $t(u,v) = \sum_{e_w \in Path(u,v)} r_w \cdot (\frac{c_w}{2} + C_w)$.

DME-Based Heuristics

The methods of [8, 12], as well as those we propose below, are generalizations of the Deferred-Merge Embedding (DME) algorithm [1, 4, 10]. Given S and 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.

To extend the DME construction to bounded-skew routing, the enabling concept is 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 to v with min-

[*] 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.

imum wiring cost (*merging cost*) $|e_a| + |e_b|$ while still maintaining the skew bound B. [8, 12] showed that under the pathlength 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).

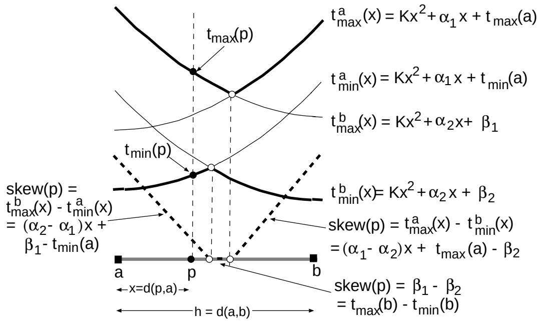
For the case where no topology has been prescribed, several topology constructions have been given which lead to low-cost routing solutions when DME is applied; the most successful is the “Greedy-DME” method of [10], which determines the merging tree topology in bottom-up fashion by iteratively merging the nearest pair of merging segments. A post-processing local search can improve the resulting topology (cf. “CL+I6” in [11]). For a non-zero skew bound, the *Extended-Greedy-DME* (ExG-DME) method [8, 12] 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 [11]. 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 [12] 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. As a result, this version of ExG-DME also very closely matches one of the best known heuristics for unbounded-skew routing [3]. The work of [12] also gives an *Extended-Planar-DME* method which again matches the best known heuristics [15] when the routing tree must be embeddable on a single layer.

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 [8, 12] 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.).

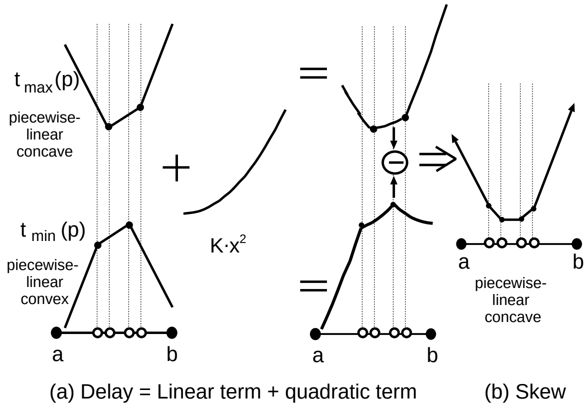
3.1 Notation and Definitions

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. Finally, we use $d(p,q)$ to denote the Manhattan distance between p and q; the distance between two pointsets P and Q is $d(P,Q) = \min\{d(p,q) \mid p \in P, q \in Q\}$.



$t^a_{max}(x) = Kx^2 + \alpha_1 x + t_{max}(a)$
$t_{max}(p)$
$t^a_{min}(x) = Kx^2 + \alpha_1 x + t_{min}(a)$
$t^b_{max}(x) = Kx^2 + \alpha_2 x + \beta_1$
$t_{min}(p)$
skew(p) = $t^b_{max}(x) - t^a_{min}(x)$ = $(\alpha_2 - \alpha_1)x + \beta_1 - t_{min}(a)$
$t^b_{min}(x) = Kx^2 + \alpha_2 x + \beta_2$
skew(p) = $t^a_{max}(x) - t^b_{min}(x)$ = $(\alpha_1 - \alpha_2)x + t_{max}(a) - \beta_2$
a
p
b
skew(p) = $\beta_1 - \beta_2$ = $t_{max}(b) - t_{min}(b)$
x=d(p,a)
h = d(a,b)

Figure 2: Properties of Elmore delays and skew over straight-line segment l connecting sinks a and b.



(a) Delay = Linear term + quadratic term (b) Skew

Figure 3: (a) Delay and (b) skew functions for a well-behaved line segment. Skew turning points are indicated by the hollow circles on $\overline{ab}$.

Well-Behaved Elmore Delay Property. Figure 2 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 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-

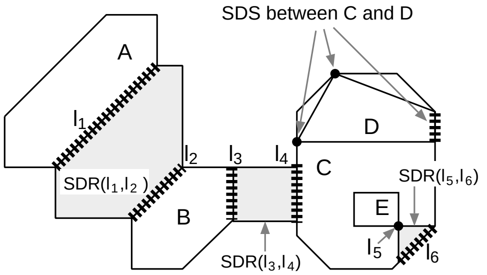


Figure 4: Examples of Shortest Distance Segments (shown as thick dashed lines or solid dots), and Shortest Distance Regions (shown as dotted regions). Note that there are three pairs of Shortest Distance Segments between the overlapping polygons C and D.

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 3). A point $p \in l$ is called a *turning point* if the slope of a piecewise-linear function defined over l changes at p. Lemma 1 in [6] 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) $n \le n_1 + n_2 - 1$, 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 4). For simplicity, we will arbitrarily take one pair of them as $SDS(P)$ and $SDS(Q)$.[1] 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 [6], we prove 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 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 [12], except that the joining segments can be parallel segments with any slopes, i.e., besides Manhattan arcs and rectilinear line segments (see Figure 5).

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 5b). 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 $skew(MSR(L_a)) \le skew(MSR(L_b))$, and $mr(v) = MSR(L_b)$ otherwise.[2]

Similar to the proof of Theorem 1 in [12], we can show that each merging region $mr(v)$ for a node v in topology G is a well-behaved region. Lemma 5 in [6] shows that the delay functions on the boundary segments of $mr(v)$ will consist of m-piecewise-linear functions and a quadratic term, where $m \le n$ (n is the number of leaves in the subtree T_v rooted at node v). Based on this fact, we can show that $mr(v)$ has at most $4n$ sides.

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.*[3] □

[1] 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.

[2] 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 of [6].

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

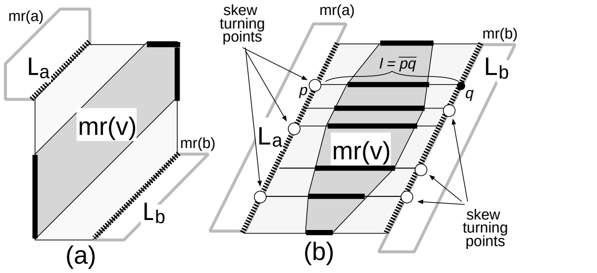


Figure 5: Construction of merging region $mr(v)$ given the merging regions for v's children a and b. The joining segments L_a and L_b, shown as thick dashed lines, are parallel Manhattan arcs (with constant max-delay and min-delay) in (a) and parallel line segments in (b). Note that merging region $mr(v)$ (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.

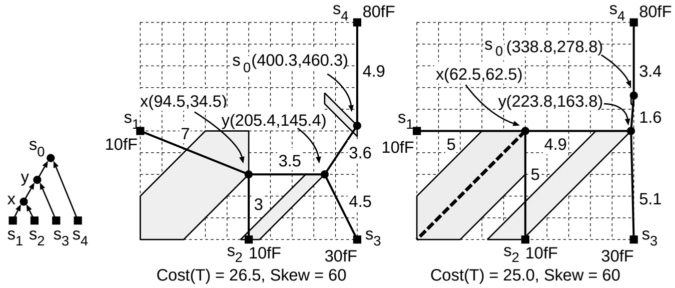
It is easy to see that for node $v \in G$ with children a and b, if (i) $mr(a)$ and $mr(b)$ do not overlap, (ii) $JS(mr(a)) = SDS(mr(a))$ and $JS(mr(b)) = SDS(mr(b))$, and (iii) no detour wiring is needed, then $mr(v)$ will contain all points having minimum merging cost. In practice, the above condition 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.

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. [12] 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 [3].

4 The Interior Merging and Embedding Method

The construction of a merging region in the BME method is based on the *boundary* segments of its child merging regions: no interior point of the child merging regions is used to construct the new merging region. This is, of course, also the case for the pathlength-based BST algorithms in [8, 12]. However, such an approach produces a sub-optimal merging cost when detour occurs during the merging. Merging with interior points may eliminate the detour. 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 (as is the case for the BME method) as in Figure 6(b), we can conserve the slack by merging at interior points and thus reduce the merging cost at a higher level (Figure 6(c)).

Given the above considerations, merging of interior points has the potential to reduce total wirelength. However, merging interior points may cause ambiguity in the delay functions for a given point p in the new merging region: this point may be a 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 that are interior to, rather than on the boundaries of, the merging regions.



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

Figure 6: Routing four sinks (filled squares) with a skew bound of 60 units. Each internal node (filled circle) is embedded in its merging region (shaded polygon). Each pair of values associated with a point represents (max-delay, min-delay). For the fixed topology shown in (a), the routing cost is ≈ 26.5 units if only nearest boundary segments are considered for merging (b). However, we can reduce merging cost by merging interior points (dashed Manhattan arc) in order to save slack for higher level use (c). The example assumes that a unit-length wire has unit resistance and unit capacitance.

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 IME, each node v in the topology G is associated with *a set* $R = \{R_1, R_2, \ldots, R_k\}$ of merging regions. The *interior points* of 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 R , denoted $SS(R$), is $\bigcup_{i=1}^{k} SS(R_i)$. We focus our attention on the use of Manhattan arcs to sample the merging regions.[4] The slope of each Manhattan arc is chosen to be $+1$ or -1 such that the points of 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 R_u and R_v be the sets of merging regions associated with u and v, respectively. The parent of u and v in G has as many as $|SS(R_u)| \times |SS(R_v)|$ possible merging regions due to the merging of each sampling segment in $SS(R_u)$ with each sampling segment in $SS(R_v)$. Since the sampling segments are all Manhattan arcs the merging operation is straightforward, as discussed earlier.

Generally speaking, within a given merging region, there is an infinite number of sampling segments. Furthermore, even if we select only a constant number of sampling segments for each region, 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 total 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 it is being merged with other regions from the sibling node. When we merge two nodes, each with $\leq k \cdot s$ sampling segments, we use dynamic programming 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 these "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 ca-

[4] The IME method can be extended easily to handle sampling segments which are horizontal and vertical. In fact, we can sample a merging region by a mixture of well-behaved sampling segments of different slopes.

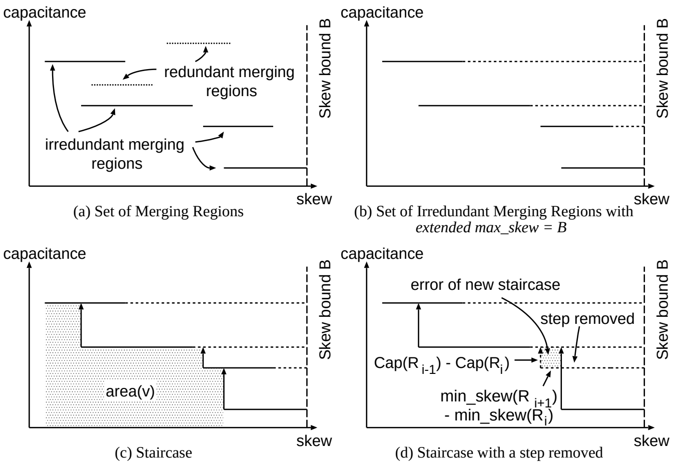

capacitance
Skew bound B
redundant merging regions
irredundant merging regions
skew
(a) Set of Merging Regions
capacitance
Skew bound B
skew
(b) Set of Irredundant Merging Regions with *extended max_skew = B*
capacitance
Skew bound B
area(v)
skew
(c) Staircase
capacitance
Skew bound B
error of new staircase
step removed
Cap(R i-1) - Cap(Ri)
min_skew(R i+1) - min_skew(Ri)
skew
(d) Staircase with a step removed

Figure 7: (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.

pacitance 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 represented by 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 $min\_skew(R') < min\_skew(R)$ and $Cap(R') < Cap(R)$ (See Figure 7(a)). Let $IMR(v) = \{R_1, R_2, \cdots R_m\}$ denote the set of irredundant merging regions of u 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 7. First, we extend the line segment representing such an irredundant merging region so that it touches the vertical line representing skew bound B (Figure 7(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 $1 < i \leq m$, we have a $m-1$ step staircase starting at $min\_skew(R_m)$ as shown in Figure 7(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 $min\_skew(R_m)$ (Figure 7(c)):

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

We can *remove* an intermediate step, say R_i ($1 < i < m$), and obtain an $(m-2)$-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 7(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 $IMR' = \{R_{\pi(1)} = R_1, R_{\pi(2)}, \ldots, R_{\pi(k-1)}, R_{\pi(k)} = R_m\}$ is minimum (or equivalently, $area(IMR') - area(IMR)$ is minimum), where $\pi : \{1 \cdots k\} \rightarrow \{1 \cdots m\}$ is a strictly monotonically increasing function. Note that both R_1 and R_m are always retained in IMR'.

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

We developed an optimal algorithm for 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 $err_i[m',k']$ denote the minimum error for the optimal (m',k')-sampling solution $S_i[m',k']$. For each $2 \leq m' \leq m$, $2 \leq k' \leq k \leq m$ and $1 \leq i \leq m - m' + 1$, we compute an optimal (m',k')-sampling solution $S_i[m',k']$ with the minimum error $err_i[m',k']$ as follows:

Case 1: If $m' = k'$, we can select all m' merging regions and therefore $err_i[m',k'] = 0$.

Case 2: If $m' > k'$ and $k' = 2$, we must retain the regions R_i and $R_{i+m'-1}$. Therefore, the 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 8):

$$\begin{aligned} err_i[m',k'] \;=\; & \{(min\_skew(R_{i+m'-1}) - min\_skew(R_{i+m'-2})) \times \\ & (Cap(R_i) - Cap(R_{i+m'-2}))\} + err_i[m'-1,k'] \end{aligned} \qquad (2)$$

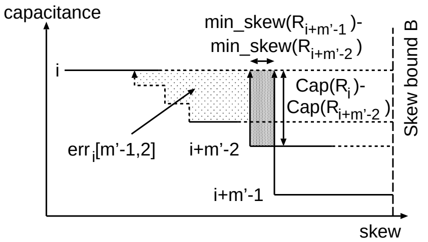


Figure 8: The lightly shaded region represents the error of the optimal solution $S_i[m'-1,2]$; the darker region is the product term given in Eqn. (2).

Case 3: 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 by Eqn. (2). Now, we have to select the optimal solution $S_{i'}[m'-i'+i, k'-1]$ from the regions $\{R_{i'}, \ldots, R_{i+m'-1}\}$. 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 $err_i[m',k']$ as follows:

$$err_i[m',k'] = \min_{i < i' \leq m'-k'+i+1} \{err_i[i'-i+1,2] + err_{i'}[m'-i'+i, k'-1]\} \qquad (3)$$

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 the following result:

Theorem 2 *The (m,k)-sampling problem can be solved optimally in $O(k \cdot m^2)$ time.* □

Skew Bound	Circuits				
(*ps*)	r1	r2	r3	r4	r5
0 (CL+I6 [11])	1253347	2483754	3193801	6499660	9723726
0(BME)	1307637	2647476	3344828	6934030	11066897
(IME)	1445555	2907593	3605778	7255455	10706940
1(BME)	1223125	2397494	3427426	6415233	9470000
(IME)	1274819	2526961	3060284	6751435	9896655
10(BME)	1087703	2155481	2789321	5411936	8140187
(IME)	1112512	2353202	2727299	5350241	8065110
100(BME)	926205	2201023	2515178	4860958	7375204
(IME)	930426	1978378	2366874	4953715	7003996
1000(BME)	793498	1839666	2506399	4971769	7134697
(IME)	861561	1995665	2097784	4740962	6280007
10000(BME)	780100	1668872	2102182	4017261	6136574
(IME)	790285	1574153	1998007	4326234	5912472

Table 1: Total wirelengths and skews of the clock routings generated by the BME and IME algorithms for benchmark circuits r1-5 [17].

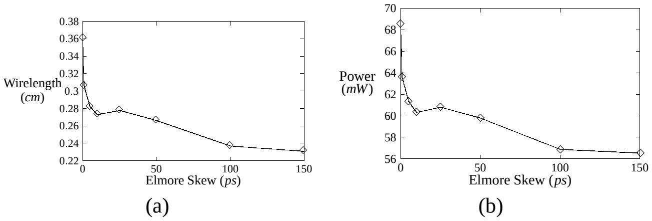


Figure 9: Tradeoff between (a) total wirelength and skew bound, and (b) power dissipation and skew bound.

Although the most expensive operation in the merging process is now due to the optimal (m,k)-sampling algorithm, this is still of polynomial complexity 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 underlying clock tree construction algorithm is retained.

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 [17] were used to evaluate our algorithms for skew bounds in the range of 0–10*ns*. Table 1 compares the zero-skew clock routing costs of the best known algorithm (CL+I6 from [11]) with the various bounded-skew routing costs obtained by our algorithms.

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 [11] 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.

A more detailed experiment for all benchmark circuits was conducted to investigate the tradeoff between total wirelength and skew, and power dissipation (obtained from HSPICE simulations) and skew for realistic skew bounds in the range of 10*ps* to several hundred *ps*. The results for the circuit r3 are shown in Figure 9. When the skew bound is relaxed from zero to 150*ps*, we achieved a power reduction of 17.6%. We also achieved 25.8% wirelength reduction when compared to the best reported zero-skew solution (by the CL+I6 algorithm in [11]).

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 also plan to combine the techniques of BST topology generation with our recent work on optimal sizing of interconnects and drivers [7, 9], 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] K. D. Boese and A. B. Kahng, "Zero-Skew Clock Routing Trees With Minimum Wirelength", *Proc. IEEE 5th Intl. ASIC Conf.*, Rochester, September 1992, pp. 1.1.1 - 1.1.5.

[2] 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 Computer-Aided Design*, 1995.

[3] 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.

[4] 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.

[5] 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.

[6] J. Cong, A. B. Kahng, C.-K. Koh and C.-W. A. Tsao, "Bounded-Skew Clock and Steiner Routing Under Elmore Delay," *UCLA Computer Science Department Technical Report 950030*, Aug. 1995.

[7] 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.

[8] J. Cong and C.-K. Koh, "Minimum-Cost Bounded-Skew Clock Routing," *Proc. IEEE Intl. Symp. on Circuits and Systems*, Vol 1, Apr. 1995, pp. 215–218.

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

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

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

[12] J. H. Huang, A. B. Kahng and C.-W. A. Tsao, "On the Bounded-Skew Routing Tree Problem", *Proc. ACM/IEEE Design Automation Conf.*, Jun. 1995, pp. 508–513.

[13] 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), Jul. 1992, pp. 893-902.

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

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

[16] 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.

[17] R. S. Tsay, "An Exact Zero-Skew Clock Routing Algorithm," *IEEE Transactions on Computer-Aided Design*, 12(2), Feb. 1993, pp. 242-249.

[18] Q. Zhu and W. W.-M. Dai, "Perfect-Balance Planar Clock Routing with Minimal Path-Length," *IEEE Intl Conference on Computer-Aided Design*, 1992, pp 473–476.