

Table-Lookup Methods for Improved Performance-Driven Routing*

John Lillis
Dept. of EECS, U.I. Chicago
Chicago, IL 60607-7053
jllillis@eecs.uic.edu

Premal Buch
Magma Design Automation
Palo Alto, CA 94303
premal@magma-da.com

Abstract

The inaccuracy of Elmore delay [3] for interconnect delay estimation is well-documented. However it remains a popular delay measure to drive performance optimization procedures such as wire-sizing and topology construction. This paper studies the merits of incorporating "better-than-Elmore" delay measures into the optimization process. The proposed delay metrics use a table-lookup method to incorporate better load modeling and approximate the effect of signal slew. We demonstrate that the proposed metrics exhibit a much narrower error distribution than Elmore delay, eliminating Elmore's frequent gross delay over-estimation. Finally we show the improvement in solution quality which can be had by incorporating the new metrics into a timing driven topology construction algorithm.

1 Introduction

The advent of deep submicron (DSM) technology has produced a great deal of interest in CAD algorithms for interconnect delay optimization. Past work includes algorithms for buffer insertion, wire sizing, high-performance topology construction and various hybrids of these techniques. Most of these algorithms are based on Elmore delay or even simpler delay models. This paper explores the amount of improvement in end solution quality that may be had by incorporating a more accurate delay model into an optimization procedure.

For any particular optimization technique such as topology construction, an algorithm designer must make several key decisions:

- (1) What *solution space* should be considered?
- (2) What should be the precise *problem formulation*?
- (3) What *delay measure* should be incorporated?

A reasonable taxonomy reflecting these decisions appears in Figure 1. The figure gives three partial orders, one for

*This research was funded in part by grants from NSF project number MIP-9625910 and SRC task 324.012.

Permission to make digital/hard copy of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
DAC 98, San Francisco, California
©1998 ACM 0-89791-964-5/98/06..\$5.00

each decision above. The partial orders proceed from less-expansive solution spaces, simpler problem formulations, and simpler delay models on the left to progressively more ideal spaces, formulations and delay models on the right. An algorithm may either be heuristic or optimal for a particular triple of choices. For instance the P-Tree algorithm [5] is optimal for choices (1B, 2F, 3C). On the other hand, it can only be considered a heuristic solution to, for example, the ideal of (1D, 2F, 3F). Similarly, a shortest paths tree (physical path length) is optimal for (1A, 2A, 3A), but heuristic for more realistic combinations.

In principle, the merit of an optimization algorithm (in addition to such criteria as run-time) is measured by the quality of the solutions it produces versus those produced by a hypothetical "ideal" algorithm — e.g., one which finds optimal solutions over the entire space of Steiner trees (1D), under the min-cost timing feasible formulation (2F) and with Spice providing delay measurements.¹ In [2], the authors argued that the "fidelity" of Elmore is high — i.e., that an 'optimal' Elmore solution was near an 'optimal' Spice solution. This was done experimentally by finding spice optimal solutions through enumerative techniques for the the combination (1A, 2C, 3C) and (1A, 2C, 3F) — i.e., it was assumed that only the delay to an arbitrarily chosen sink was of importance and the remaining delays could be neglected. It was concluded that optimal Elmore solutions were very close in performance to the optimal Spice solutions. The variance of the Elmore/Spice and a 2-pole/Spice ratios were also statistically studied and shown to be nearly identical.

While the results in [2] are interesting and provide some degree of assurance that following Elmore can produce high quality solutions, we cannot conclude that, for instance, Elmore produces high quality solutions for the min-cost timing feasible problem formulation.

This paper explores this question by incorporating a "better-than-Elmore" delay model into an established timing-driven Steiner routing algorithm [5, 4]. It is our goal to give an idea of how much improvement can be had by incorporating more accurate models (or more precisely to give an empirical lower-bound on this possible improvement). In the context of Figure 1, this is achieved by comparing the results of two algorithms: one which solves the combination (1B, 2F, 3C) to one which attacks the combination (1B, 2F,

¹Of course, the definition of the "ideal" algorithm can become arbitrarily complex when one considers such things as coupling capacitance, or false-paths at the logic level, etc.

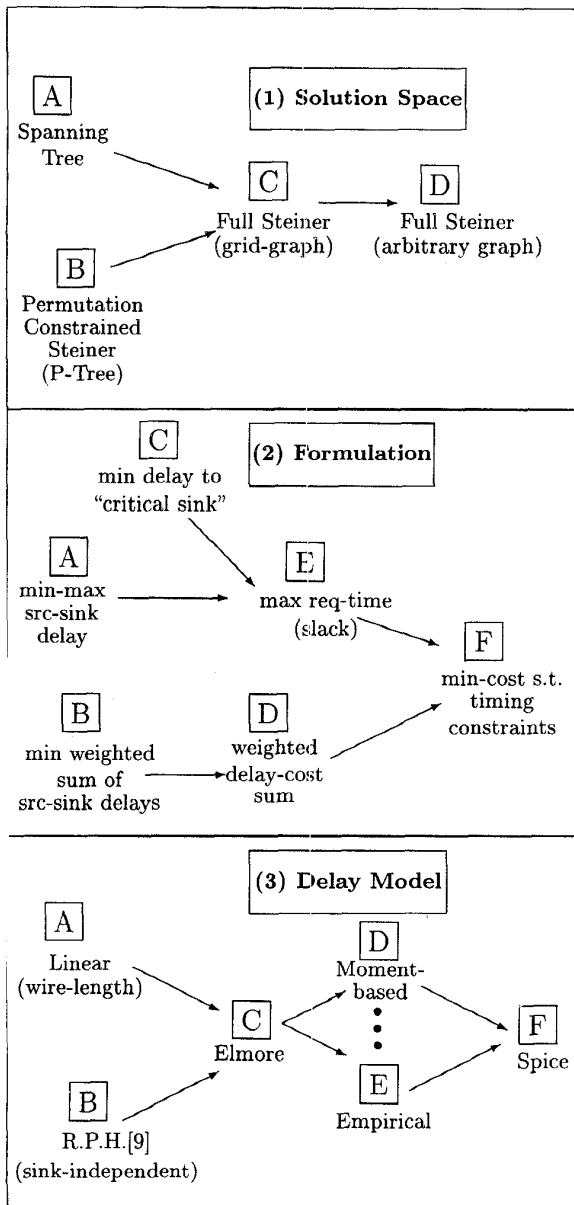


Figure 1: Taxonomy of Timing-Driven Steiner Routing

3E). Thus, we have a fixed solutions space and formulation while varying the delay model dimension.

We selected the permutation constrained solution space (1B) used by the P-Tree algorithm for several reasons. First, it has produced the best published results for formulation 2F under Elmore delay. Second, it provides a framework in which new delay models can be incorporated relatively easily (provided the models can be evaluated “bottom-up”). Additional background on the P-Tree algorithm is presented in Section 4.

2 The Nature of Elmore’s Error

In RC trees two main factors contribute to Elmore’s error [7]:

- its inability to account for resistive shielding (an increasingly important phenomenon in deep submicron)
- its inability to account for the effect of signal slew rate τ .

As a result, Elmore can grossly overestimate the delay to nearby sinks for which the resistive shielding effect is significant and the signal waveform is relatively sharp. Conversely, as the signal slew increases (i.e., as we travel to more distant sinks), Elmore becomes increasingly accurate.² The danger in an optimization algorithm is that this error will cause misidentification of critical paths and lead the algorithm to make incorrect decisions by optimizing delay to a false critical sink at the expense of the true critical sink.

3 Proposed Delay Models

Clearly it is desirable to improve upon the deficiencies of Elmore described in the previous section. In addition, for our purposes a proposed model should be relatively easily incorporated into an optimization procedure. With these two goals in mind, we have devised the table-lookup method summarized below.

Interpolation in a fairly sparse 5D table is used to estimate the delay of wire segments by interpolation as we proceed bottom-up. Each segment is restricted to be short enough so that it can be approximated by a lumped RC model. A wire link in the topology then corresponds to one or more segments. The delay of a segment AB depends on the resistance and capacitance of the segment AB, the downstream RC subtree rooted at B and the slew of the driving signal at A. All the relevant resistances and capacitances are captured with a pi-model approximation of the load at B (see Figure 2). We then estimate the delay from A to B (50%-to-50% delay) with the 5 parameters $R1, R2, C1, C2$ and τ . The lookup table is constructed for a range of $R1, R2, C1, C2$ and τ via multiple Spice runs. The values of $R1, R2, C1, C2$ and τ for a given segment AB are computed as follows:

- At each stage we estimate the signal slew τ at A by a multiple of the Elmore delay from the root to A (or an approximation thereof). The coefficient for the Elmore delay based slew estimate is determined as follows: the signal slew at an internal node A in an RC tree depends on the downstream RC subtree seen by node A. If the Elmore delay to this subtree is

²Indeed, for ramp inputs Elmore approaches the true 50% delay as τ tends toward infinity [7].

$ED(v_d, A)$, the voltage waveform at node A is given by $v_A(t) = V_{DD}(1 - e^{-t/ED(v_d, A)})$ under the assumption that a single dominant time-constant exists for the RC subtree rooted at node A . The signal slew τ_A at node A is then given by $\tau_A = t(v_A(t) = 0.9V_{DD}) - t(v_A(t) = 0.1V_{DD}) = 2.197ED(v_d, A)$. Note that the single dominant time-constant assumption is also made in deriving the Elmore delay model, and the above formulation of the signal slew is thus exact under Elmore delay.

- For the pi-model, we accumulate the first three coefficients of the Taylor series expansion of the driving point admittance at node B , in a bottom-up fashion as in [6]. From this a pi-model approximation is derived for an RC tree (the subtree rooted at B in this case) using the algorithm of [6]. This pi-model of the RC sub-tree is combined with the resistance and capacitance of the segment AB to obtain the model in Figure 2. In this case, $R1$, the “driver” resistance, corresponds to the resistance of the segment AB , $C2$ is the sum of half the capacitance of the segment AB (i.e., AB itself is approximated by a pi-model) and the upstream branch capacitance of the pi-model of the RC subtree rooted at B , and $R2$ and $C1$ are the other two branch elements of this subtree’s pi-model.

Note that when computing the delay across a segment driven by the root node, $R1$ only serves as the ‘driver’ resistance. By applying the above lookup procedure while travelling bottom-up, we can compute the delays along all paths from sinks to the root.

As an aside, we emphasize the importance of the often overlooked process of finding an appropriate scaling coefficient β for any delay model – i.e., for a model such as Elmore, we select β so as to “center” the error distribution (0.69 being the typical coefficient for Elmore). While a scaling coefficient will not alter the solutions produced by an algorithm designed for minimizing max source-to-sink delay, this is not the case for required-arrival-time (or just “required-time”) formulations; in the first case, $d_1 < d_2 \Leftrightarrow \beta d_1 < \beta d_2$ while in the latter case each sink u introduces its own offset in the form of its required-arrival-time q_u and we cannot claim such a property (i.e., the choice of β can alter the critical path).

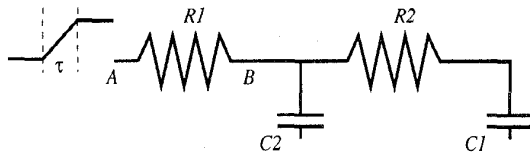


Figure 2: Load modeling by a pi-model circuit

To further simplify the incorporation of our new models into the P-Tree Steiner routing algorithm [5], we also introduced another model which uses an estimate of the Elmore delay from the driver to an arbitrary point in the tree. This is because in the bottom-up dynamic programming algorithm, we do not yet know the structure of the unexplored portion of the tree.

The three models for the delay from node A to node B (refer to Figure 2) are summarized below (note the only difference between TL1 and TL2 is the method used for approximating τ ; the coefficient 2.197 follows the rationale presented in the previous section).

- **Scaled Elmore:** $ED_s(A, B) = \beta_E R1(C1 + C2)$
- **Table Lookup Exact Elmore (TL1):** Let v_d be the driver node and $ED(v_d, A)$ be the Elmore delay to A . Let $\tau_1 = 2.197 \cdot ED(v_d, A)$. Then we have, $TL1(A, B) = \beta_{TL1} \cdot \text{lookup}(\tau_1, R1, R2, C1, C2)$.
- **Table Lookup Appx Elmore (TL2):** Let l be the distance from the driver pin to A and $r(l)$ and $c(l)$ the resistance and capacitance of such a wire. Let c_0 be a lower-bound on the total capacitance of the entire routing tree. Let $\tau_2 = 2.197(r_d c_0 + r(l)(c(l)/2 + c_0/2 + C1 + C2))$. Then we have, $TL2(A, B) = \beta_{TL2} \cdot \text{lookup}(\tau_2, R1, R2, C1, C2)$ of a wire of length l with load $(c_0/2 + C1 + C2)$.

4 The P-Tree Algorithm

We briefly review the concepts behind the P-Tree algorithm. We refer the reader to [5] and [4] for further details. The main principle behind the algorithm is to identify a constrained solution space in which optimal solutions (under Elmore and the min-cost timing feasible formulation) can be identified relatively efficiently and yet still provides a large and flexible set of candidate solutions. The constrained solution space is that imposed by a (carefully constructed) permutation on the sinks of the net: a topology is permissible if and only if the sinks contained in each subtree of the topology is exactly a consecutive subsequence in the given permutation. This describes the topology space. The algorithm also *simultaneously* finds the embedding of the topology in the grid-graph formed by the pin locations. The algorithm guarantees optimality over all permissible topologies and embedding of those topologies.³ Figure 3 illustrates this solution space for a given sink permutation “a, b, c, d, e, f, g” with a few examples of topologies and possible embeddings giving an idea of the flexibility of the solution space.

The method employed for finding a “good” sink permutation follows the intuition that subsequences of the permutation should represent relatively closely clustered sets of pins. The tour length (in the sense of the Travelling Salesman Problem) is adopted as a reasonable measure of this clustering quality. We refer the reader to [5] and [4] for further details on permutation construction and details of the P-Tree dynamic programming algorithm itself.

5 Experiments

After constructing the 5D lookup table (with 10 entries in each dimension or 100k total entries) from spice data we performed two sets of experiments.

Our first set of experiments studies the error distribution of the three models themselves. We first generated a large set of routing topologies of 4, 6 and 8 pin nets. These were drawn from routing regions of size 1cm x 1cm, 0.5cm x 0.5cm and 0.25cm x 0.25cm. For each net cardinality and

³The algorithm actually produces a family of topologies giving a performance vs. cost (total wire-length) tradeoff.

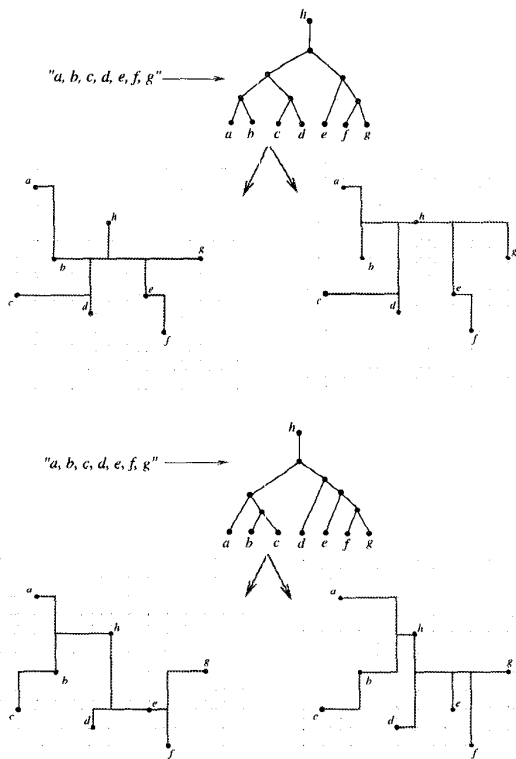


Figure 3: The P-Tree Solution Space

routing region size pair we generated 50 point sets and corresponding topologies using the P-Tree algorithm (not to produce high performance topologies, merely reasonable topologies for our testbed). Technology parameters (corresponding closely with $0.5\mu\text{m}$ technologies in the literature) were as follows:

- driver resistance: 300Ω
- wire resistance: $0.112\Omega/\mu\text{m}$
- wire capacitance: $0.15\text{fF}/\mu\text{m}$
- sink cap: $0.05\text{fF}/\mu\text{m}$

For each topology, the delay to each sink was computed for each model in addition to Spice. After this data was collected, scaling factors were selected so as to minimize the mean-squared error of each model versus Spice. Figure 4 shows histograms for each of the delay models; for each model there is one histogram showing error in pico-seconds vs. spice and another showing relative error as the ratio of the predicted delay to the spice delay. As can be seen, all three models produce reasonable delay predictions most of the time; however, as can be seen in the case of Elmore, there is a long tail of large over-estimations while the table-lookup models are more well-behaved. This trend is even more dramatic when only the topologies from the $1\text{cm} \times 1\text{cm}$ routing

region were considered as shown in Figure 5.

The standard deviation of the delay models to spice ratios were as follows:

- Scaled Elmore: 0.321
- TL1: 0.069
- TL2: 0.073

As may be expected, the table-lookup methods produced markedly more stable delay predictions. It is also interesting that, despite its crude estimation of signal slew, TL2 behaves nearly as well as TL1.

The various scaling coefficients were as follows. For Elmore, the scaling coefficient β_E was 0.69 (matching the theoretically correct $\ln 2$.) The scaling coefficients for TL1 and TL2 were $\beta_{TL1} = 1.13$ and $\beta_{TL2} = 1.15$.

Our second set of experiments give preliminary results on the amount of improvement in final solution quality we observed after we incorporated TL2 into the P-Tree algorithm (P-Tree_{TL}) versus those produced by Elmore-based P-Tree. We generated 100 6-pin Steiner routing instances with the technology parameters from the previous section. Required-times for each sink were drawn at random between 1ns and 3ns. Both algorithms were run on each instance and the resulting topologies were compared in terms of wire length and actual spice-computed required-times (i.e., slack at the root). This was done in two scenarios: first, the fastest topologies (as predicted by the respective algorithms) were compared; second, the fastest topology from each algorithm whose wire-length was not more than 25% greater than that of the min wire length solution (recall that the P-Tree algorithms produce a family of solutions with a wire-length vs. required-time tradeoff).

The results of these experiments are shown graphically in Figures 6 and 7. Each plot shows the difference in spice-computed required-time (in pico-seconds) versus the table-lookup/elmore wire-length ratios. Thus, points in the lower-right quadrant indicate that the table-lookup topology had both better performance *and* shorter wire length than the corresponding Elmore-based topology. With the exception of two outliers in Figure 6, the trend clearly favors the table-lookup method (also, the topologies in Figure 7 are more likely to be of practical interest because of their more modest wire-length overhead). Finally, we note that in a number of cases both algorithms produced identical topologies: where the fastest topologies were selected, 67/100 were identical, where only 25% overhead was allowed, 68/100 were identical.

6 Discussion

In this paper we have presented evidence that indeed some improvement in solution quality may be had by incorporating better-than-Elmore delay models into the performance-driven routing process. The proposed models themselves were shown to have significantly better error distribution than Elmore. While the routing solutions produced by each model agree a good portion of the time, a significant fraction favor the improved models substantially.

Future work along these lines may include a similar method for improved buffer modeling in the buffer insertion process (and to better model the driver in case presented in this paper where we modeled it as a resistor for simplicity). Performing similar experiments for $0.25\mu\text{m}$ technologies is also of interest since Elmore's error distribution presumably

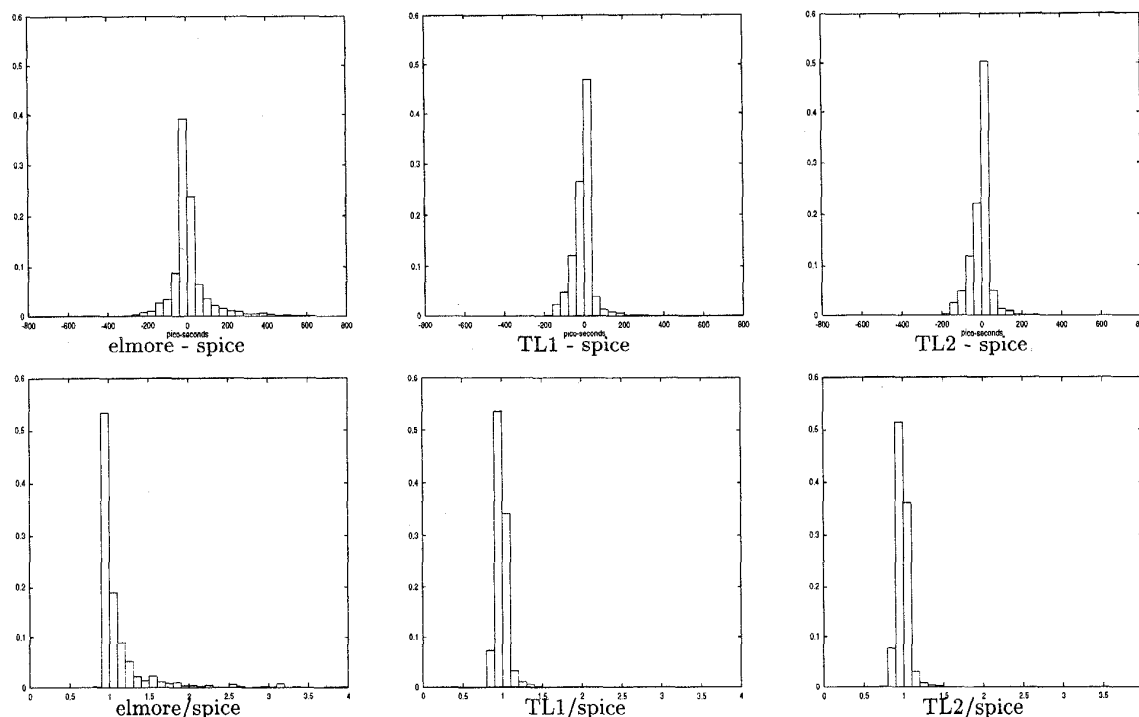


Figure 4: Error Distributions Over All Topologies (all models scaled)

will become even worse.

Finally we note that P-Tree with the table-lookup models does not ensure optimality under the given model (as it does for Elmore). This is a result of a basic pruning premise in P-Tree and many similar algorithms: if a candidate solution for a subtree is faster and has less capacitance than a second solution, the second is considered sub-optimal and discarded. This is correct for Elmore, but it is conceivable that the second solution's *effective capacitance* [8] is less than that of the first and therefore could contribute to an optimal global solution. Exploring this phenomenon is another possible direction for future work. The notion of effective capacitance may also provide a viable alternative to the table-lookup methods presented here.

Acknowledgement

The authors thank Professor Ernest Kuh of U.C. Berkeley for his encouragement and several helpful discussions.

References

- [1] K. D. Boese, A. B. Kahng, B. A. McCoy, G. Robins, "Fidelity and Near-Optimality of Elmore-Based Routing Constructions," *Proc. Proc. IEEE Intl. Conf. Computer-Aided Design*, 1993.
- [2] Boese, K.D.; Kahng, A.B.; McCoy, B.A.; Robins, G. "Near-optimal critical sink routing tree constructions." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Dec. 1995, vol.14, (no.12):1417-36.
- [3] W.C. Elmore, "The Transient Response of Damped Linear Network with particular Regard to Wideband Amplifiers," *J. Applied Physics* 19 (1948), pp 55-63.
- [4] J. Lillis "Algorithms for Performance Driven Design of Integrated Circuits," Technical Report #CS96-492, CSE Dept., U.C. San Diego, Aug. 1996
- [5] J. Lillis, C.-K. Cheng, T.-T. Lin, C.-Y. Ho, "New Techniques for Performance Driven Routing with Explicit Area/Delay Tradeoff and Simultaneous Wire Sizing," *Proc. 33rd ACM/IEEE Design Automation Conference*, Las Vegas, Jun. 1996, pp. 395-400.
- [6] P. R. O'Brien and T. L. Savarino, "Modeling the Driving-Point Characteristic of Resistive Interconnect for Accurate Delay Estimation," *Proc. IEEE Intl. Conf. Computer-Aided Design*, Nov., 1989, pp. 512-515.
- [7] R. Gupta, B. Tutuianu, L.T. Pileggi, "The Elmore delay as a bound for RC trees with generalized input signals." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Jan. 1997, vol.16, (no.1):95-104.
- [8] Qian, J.; Pullela, S.; Pillage, L. "Modeling the "Effective capacitance" for the RC interconnect of CMOS gates". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Dec. 1994, vol.13, (no.12):1526-35.
- [9] J. Rubinstein, P. Penfield, and M.A. Horowitz, "Signal Delay in RC Tree Networks," *IEEE Trans. on CAD* 2(3) (1983), pp 202-211.

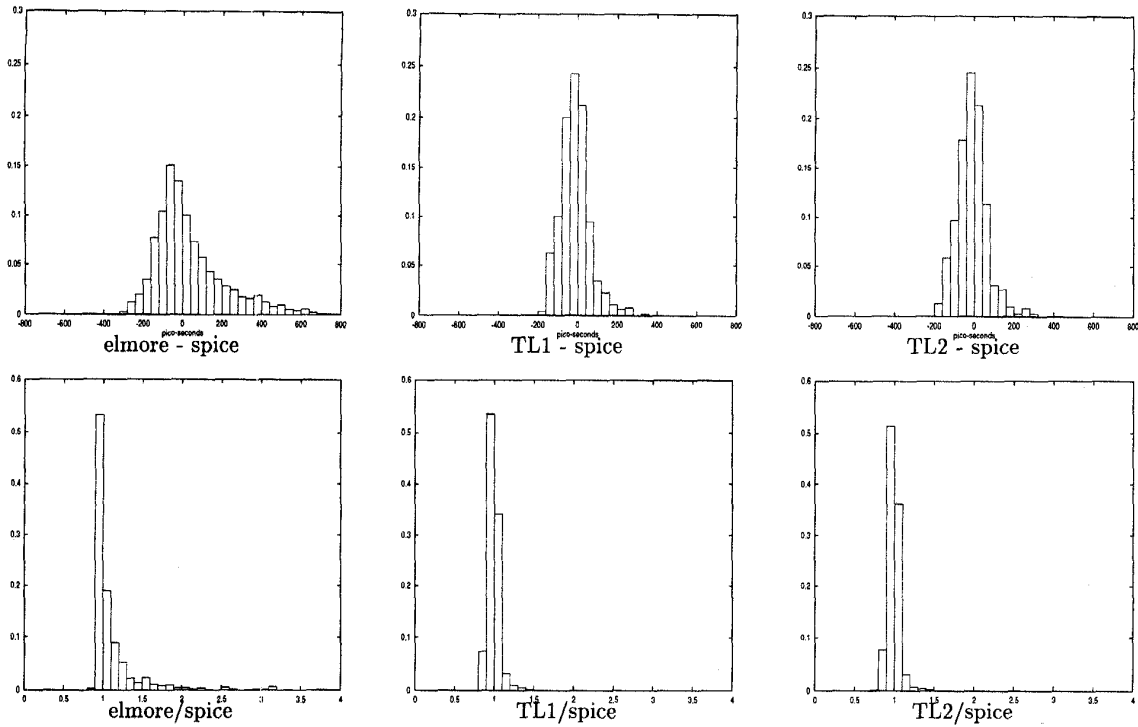


Figure 5: Error Distributions Over Topologies Drawn From a 1cm x 1cm Routing Region (all models scaled)

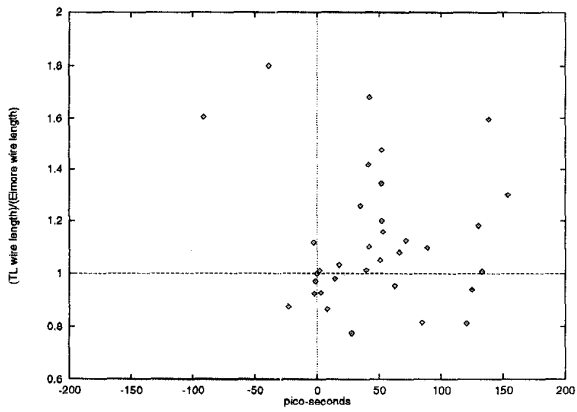


Figure 6: Results for 'fastest' topologies produced by Elmore-based and table-lookup based P-Tree. X-axis is the difference between the table-lookup required time and the Elmore required-time (as computed by Spice).

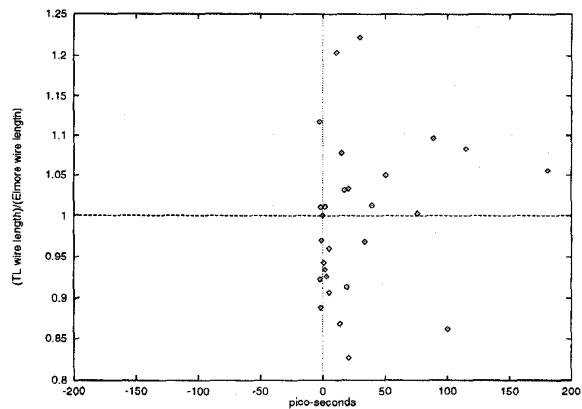


Figure 7: Results for 'fastest' topologies with wire-length no more than 25% greater than that of the min wire-length solution.