Statistical Timing Based Optimization using Gate Sizing

Aseem Agarwal, Kaviraj Chopra, David Blaauw, *Vladimir Zolotov

University of Michigan, Ann Arbor

* IBM T. J. Watson Research Center, Yorktown heights

TAU

1st March, 2005

Overview – Circuit optimization

- Optimization formulation: Minimize the circuit Area/Power
  - Constraint on circuit delay (timing yield) or vice versa
  - Determine design variables (ex: gate size)
- Outcome
  - Improved power, area, delay, noise – well tuned circuit

- Deterministic optimization – delay is non-statistical
  - Compute the sensitivity of the obj. fn. to design variables using STA
  - Feed into Non-linear optimizer

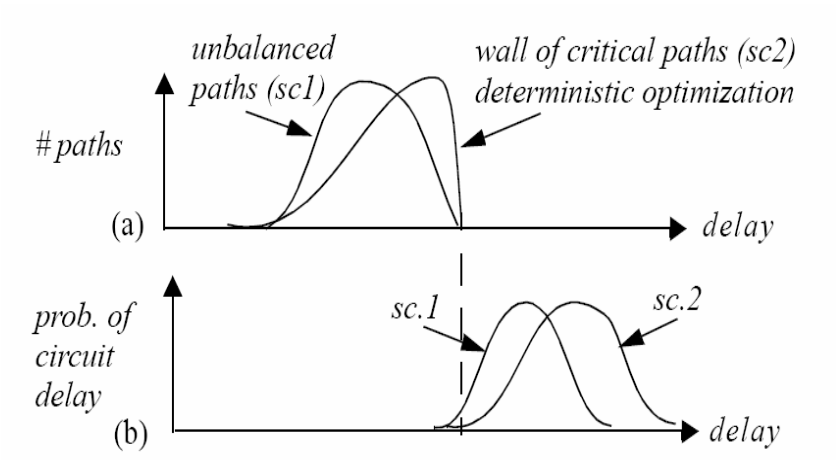
- Process variation not properly accounted for
  - Yield loss
  - Need for a Statistically-aware optimizer (statistical delay models)

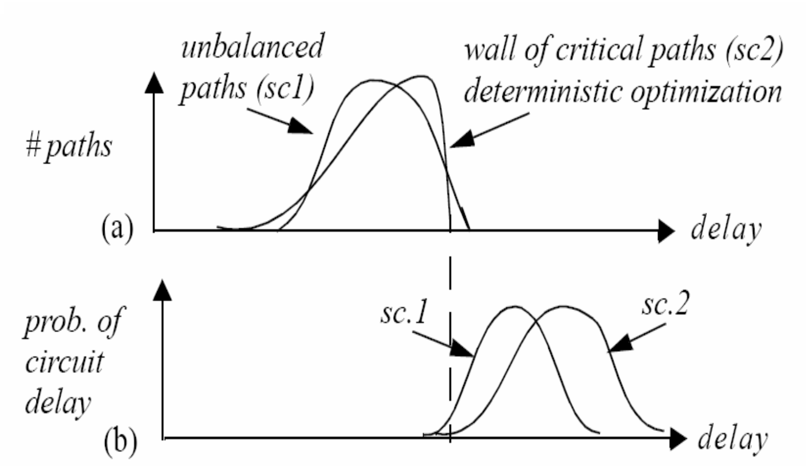
Outline of talk

- Need for Statistical Timing Based Optimization
- Our Contribution
- Results and Analysis
- Summary

Need for Statistical Timing Based Optimization

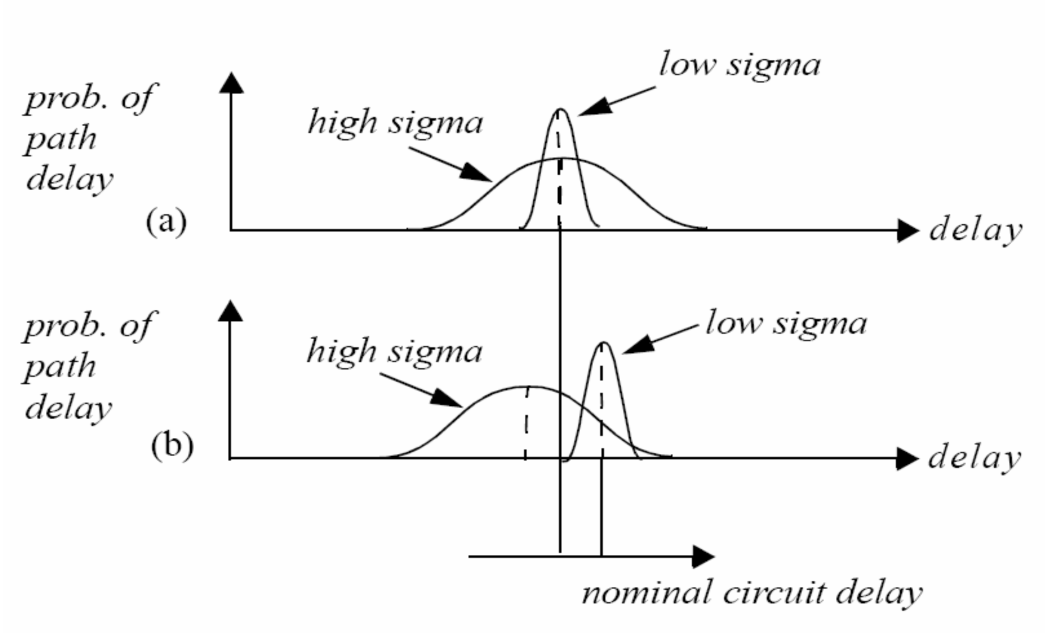
- Deterministic Optimization creates a so-called *timing-wall*
  - No advantage in improving non-critical paths
  - Degrade statistical performance





- Statistical Delay improvement (given a det. sized ckt.)
  - With an additional area penalty
    - Same nominal delay
  - Iso-area
    - Increased nominal delay

Iso-area - example



- Critical and near-critical paths of a design
  - Different standard deviations
- Size down low sigma paths / Size up high sigma paths (one trade-off)
  - Decision made by the optimization objective

Prior Work

- Deterministic delay model based
  - H. Hashimoto [ISCAS '01]
    - Deterministic optimization creates a *timing wall*
    - Height of the *wall* impacts the statistical delay by large amt.
  - X. Bai [DAC '02]
    - Provide incentive in the det. formulation to avoid a *wall*

- Statistical delay model based (Non-linear programming problem)
  - E. Jacobs [DATE '00]
    - Gaussian approximation for max (analytical formulation)
    - Sensitivity computation complexity is O (n^2)
  - S. Raj [DAC '04]
    - Path based approach – enum. all paths in worst case
    - Demonstrate large improvements on benchmark ckts.

Proposed work and assumptions

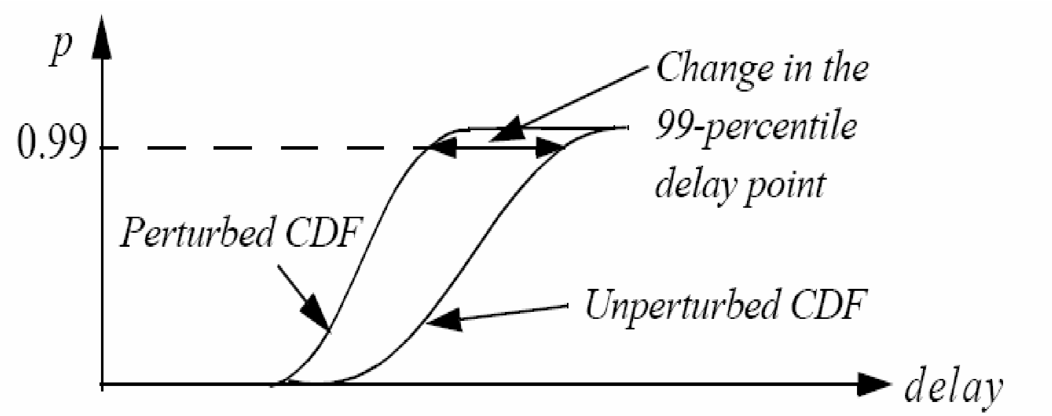
- Statistical delay model based (Co-ordinate descent)
  - Uses our bound based SSTA approach
  - Sensitivity computation
  - Exact approach for finding the high sensitivity gates
    - Fast due to our proposed pruning method
  - Combined approach using a slack based heuristic
    - Runtime improvement upto 2 orders of magnitude
  - Demonstrate significant improvements comp. to det.
  - Consider the ind. random component of intra-die
    - Extension to include spatial correlation
  - Consider pdfs to be bounded on max and min values

Outline of talk

- Need for Statistical Timing Based Optimization
- Our Contribution
  - Different optimization objectives
  - Brute force formulation
  - Theory of perturbation bounds
  - Exact sensitivity computation
  - Heuristic sensitivity computation
- Results and Analysis
- Summary

Optimization Objective

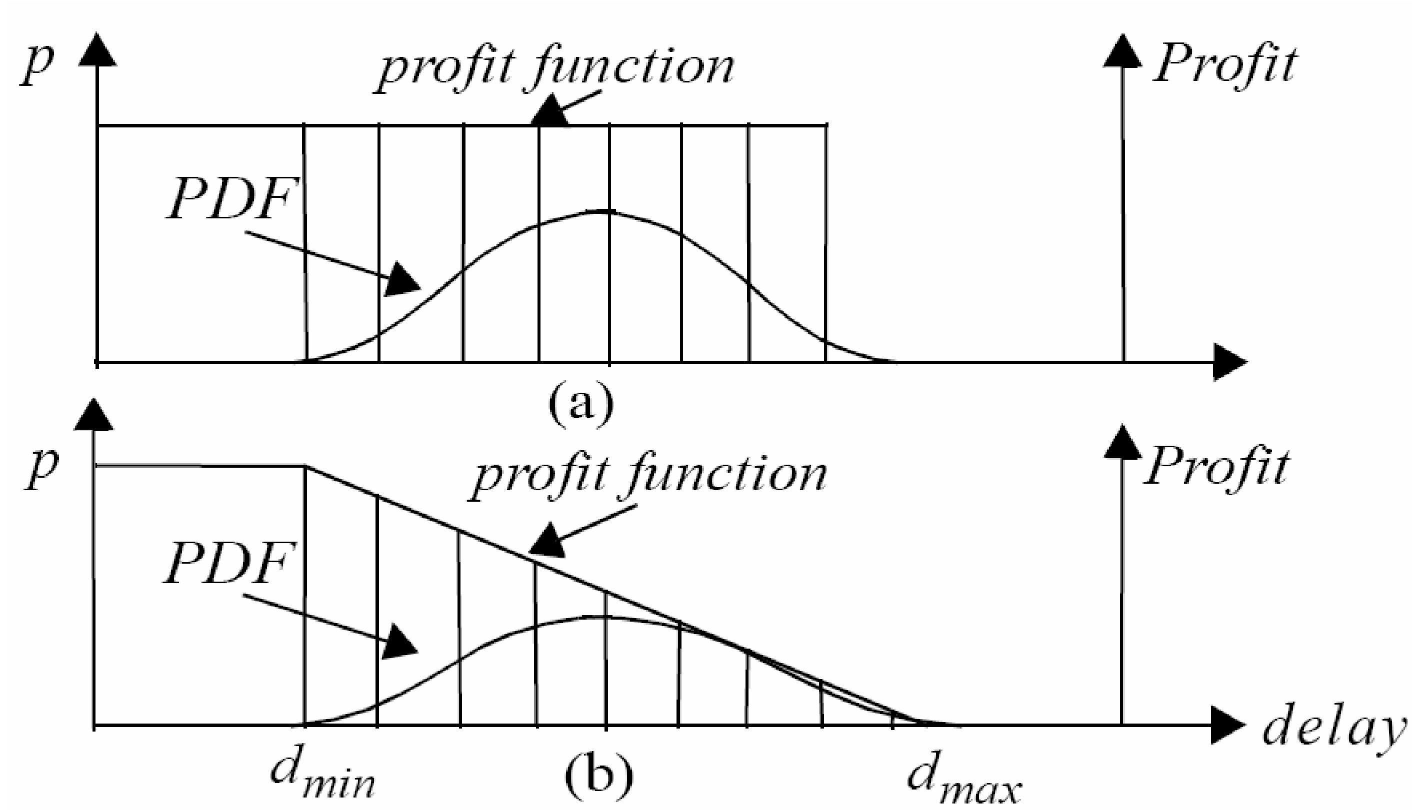
- Optimization changes both mean and shape of PDF
  - Need measure of quality



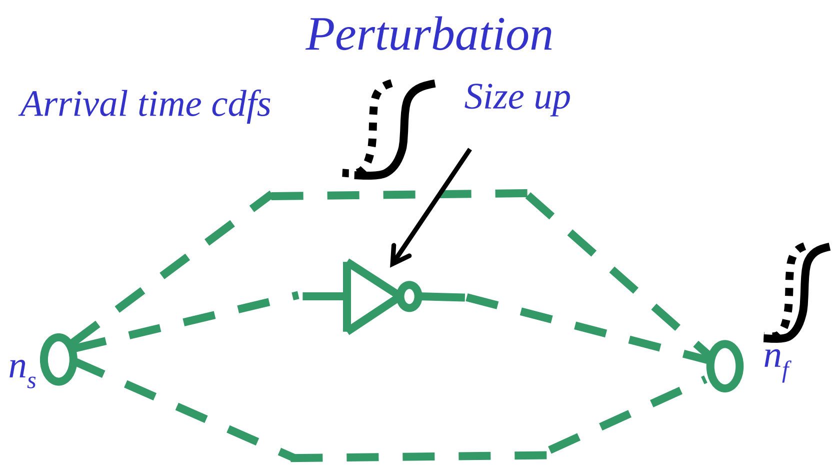
- Simple objective – 99% confidence

Optimization Objective

- Different cost functions for ASIC and Microprocessors



Brute-force formulation

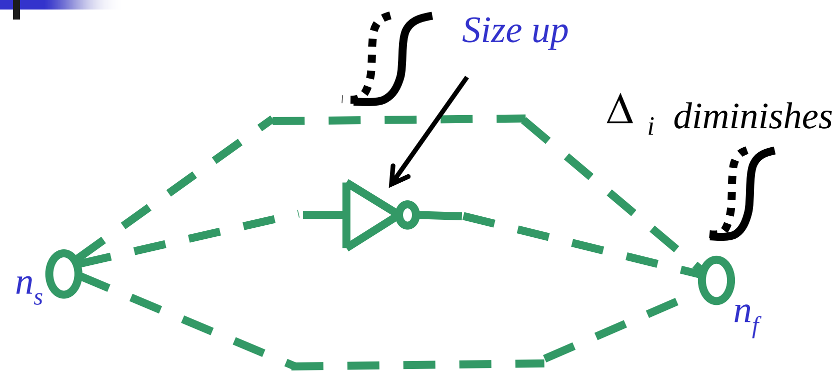


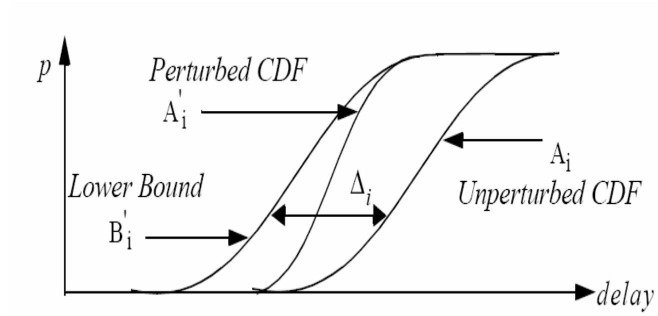
- Statistical objective function
  - helps evaluate the change in the waveform
- Sensitivity Computation
  - Complexity O(V*E)
  - Need better sensitivity computation

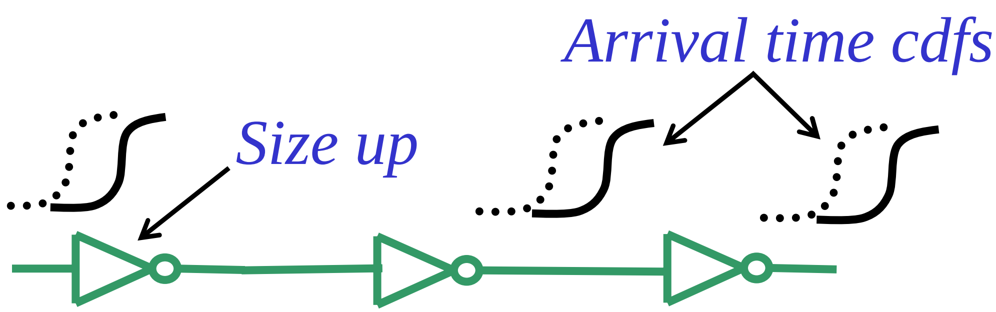
Outline of talk

- Need for Statistical Timing Based Optimization
- Our Contribution
  - Different optimization objectives
  - Brute force formulation
  - Theory of perturbation bounds
  - Exact sensitivity computation
  - Heuristic sensitivity computation
- Results and Analysis
- Summary

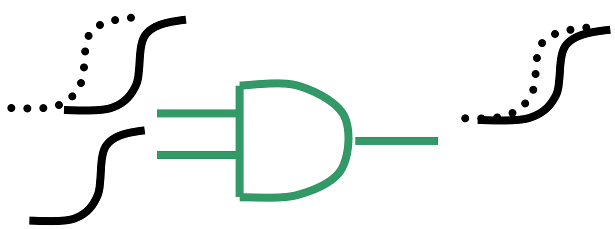
Perturbation bounds







Δ_i *remains same*



Δ_i *diminishes proof in the paper*

Exact pruning of nodes

- Δ_i can only diminish
  - This property can be used for pruning
    - Propagate a perturbation to the sink node
    - Prune other gates in the circuit if they are lesser
      - Reduces complexity

- Need to determine gates to be propagated first
  - Greedy selection
  - Level by level propagation

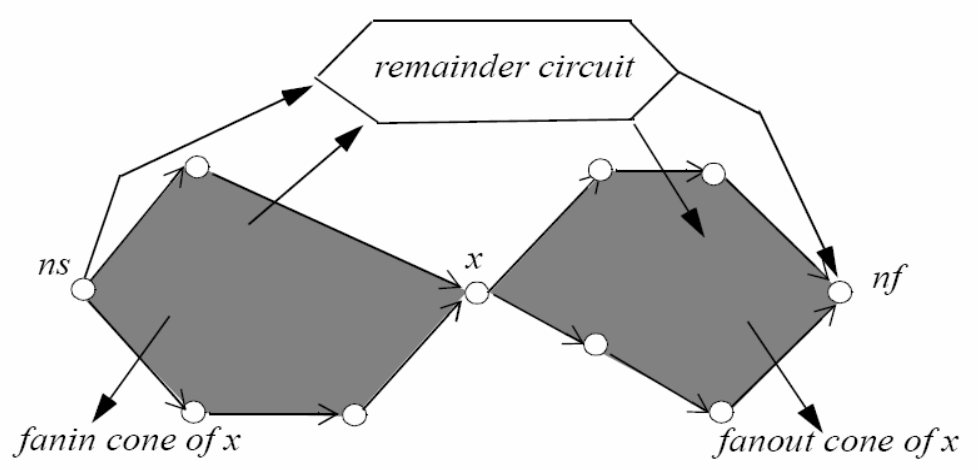
Exact Sensitivity computation

- Perform a run of SSTA
- Propagate pert. arrival times CDFs till the level of the gate
- Sort the gates by perturbation
- Select the best gate greedily and prop. one level forward
- Update the sorted order
- If gate reaches the sink node update lower bound
- Prune gates below the lower bound
  - Exact pruning
- When pruning is over
  - Either size the best gate / provide sensitivities to a n.l.o
- Repeat for next iteration

Heuristic Sensitivity computation

- Compute slack distributions by backward propagation
- Convolve forward and backward pdfs
- Compute remainder cdf (remainder ckt)
- Convolve perturbed forward and backward pdfs
- Max with the remainder cdf
- Compare with circuit delay cdf to obtain sensitivity

- Extremely fast
- Combined appr.



Experimental setup

- Synthesized ISCAS benchmark circuits (180 nm library)
- The delay model used is $D_e = D_{int} + K \times C_{load} / C_{cell}$
- Standard deviation is 10% of mean delay
- Compare w/ det. optim. using MINOS
  - Statistical timing run on the obtained circuit
- Area delay curves plotted for 800 sizing iterations

Runtime Results

- 20 X by bound based prune, 89 X improvement by combined approach

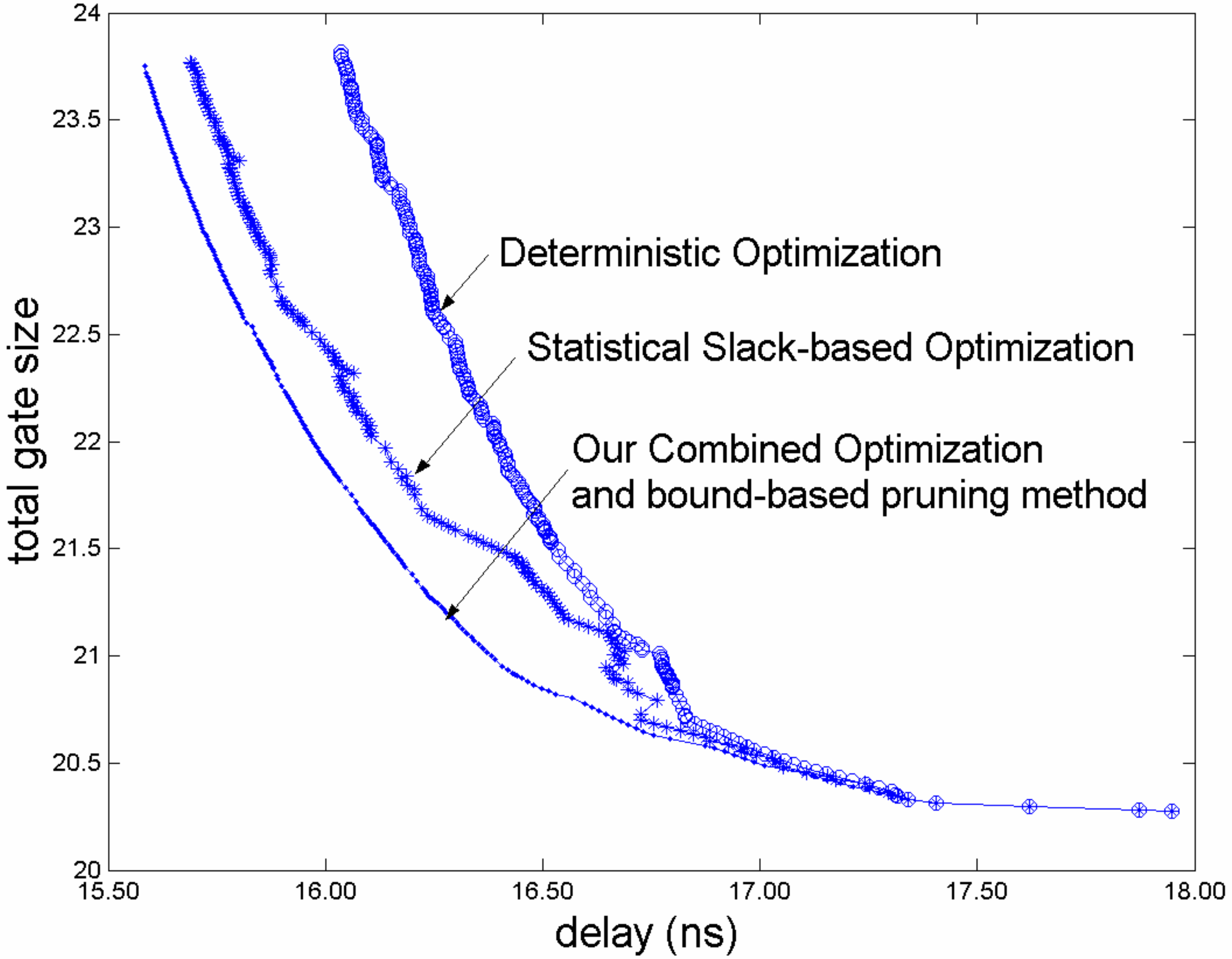
Circuit name	Average time per iteration (sec)						
	brute-f	b.b. prune	imp. f	comb.	imp. f	slack	imp. f
c432	5	1.21	4.13	0.78	6.4	0.49	10.2
c499	90	19.9	4.52	3.8	23.7	1.75	51.5
c880	15	3.37	4.45	1.07	14.0	0.85	17.6
c1355	95	19.8	4.79	3.8	25.0	1.6	59.4
c1908	102	22	4.63	5.97	17.0	2.1	48.6
c2670	43	4.47	9.62	1.36	31.6	1.2	35.8
c3540	194	23	8.43	5.8	33.4	3.4	57.0
c5315	403	33	12.2	6.8	59.3	4.2	96.0
c6288	3600	180	20.0	50.3	71.6	35.0	103
c7552	1190	87	13.68	13.4	89.0	8.4	142

Optimization results – 99% delay

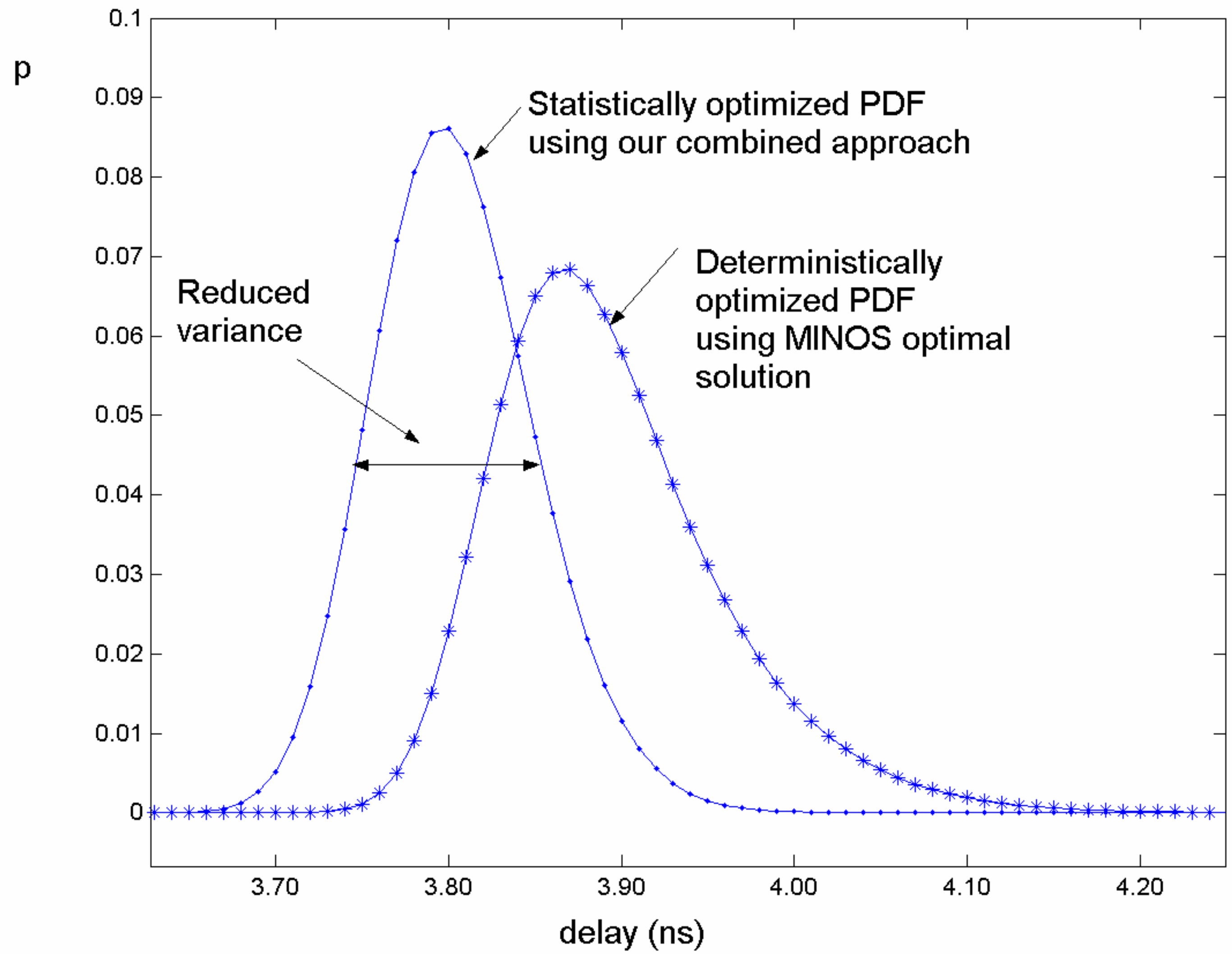
- Av. delay imp – 7.6 % , Av. Sigma imp – 17%

name	det.	slack-based		bound-based		combined approach		
	delay	delay	%impr	delay	%impr	delay	%imp	%sigma imp.
c432	3.45	3.40	1.4	3.25	5.8	3.25	5.8	12.2
c499	4.05	3.48	14.0	3.38	16.5	3.38	16.5	31.4
c880	4.18	4.04	3.3	3.94	5.74	3.94	5.74	13.8
c1355	4.70	4.25	9.5	4.10	12.7	4.10	12.7	30.7
c1908	6.20	6.02	2.9	5.82	6.1	5.82	6.1	10.8
c2670	3.61	3.55	1.7	3.50	3.0	3.50	3.0	3.0
c3540	5.98	5.80	5.2	5.70	6.9	5.70	6.9	13.5
c5315	5.90	5.70	3.4	5.40	8.47	5.40	8.47	14.8
c6288	15.8	15.5	1.9	15.05	4.75	15.05	4.75	23.0
c7552	8.10	7.80	3.8	7.60	6.17	7.60	6.17	13.4

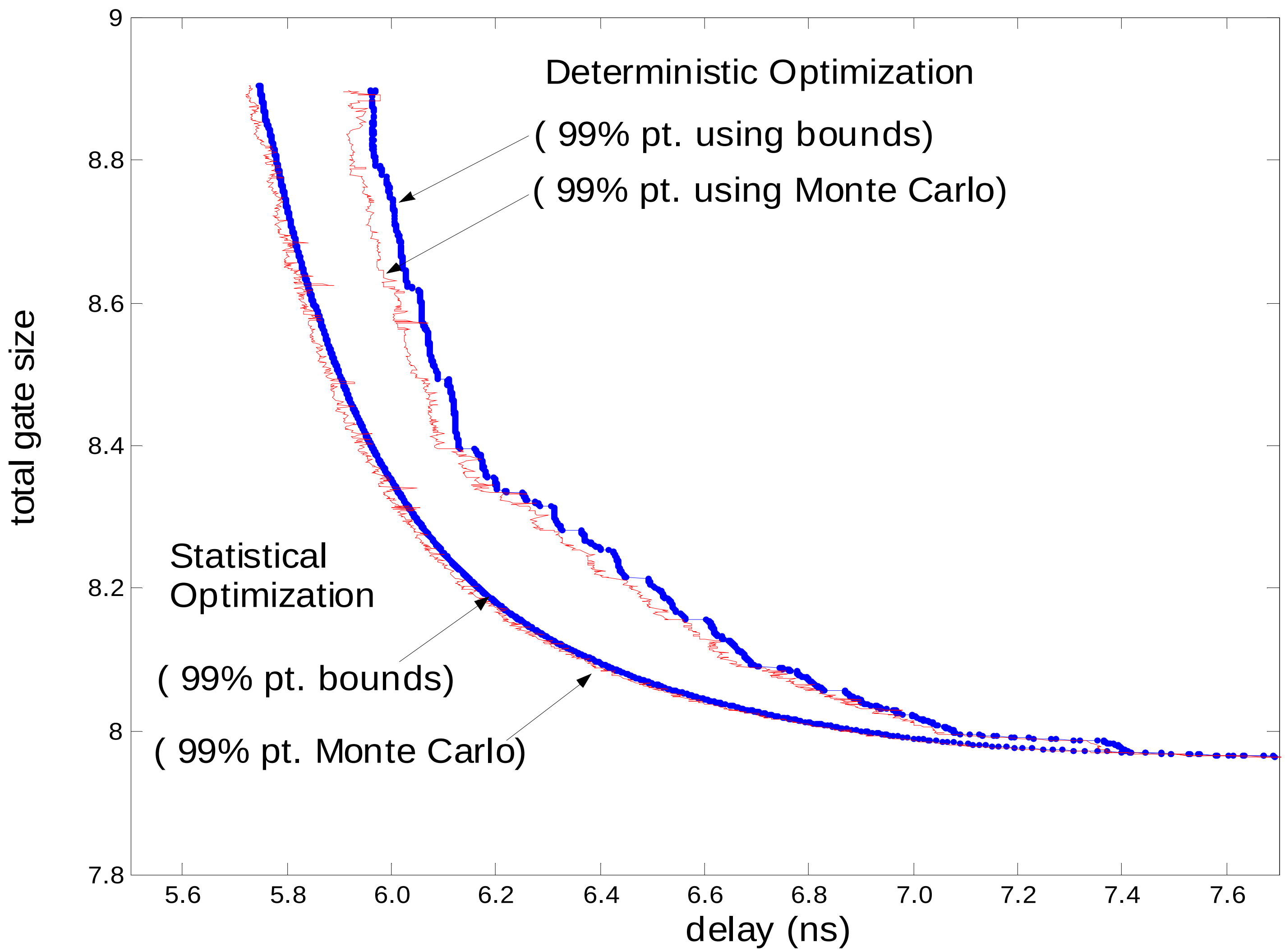
3 different optimizations – c6288



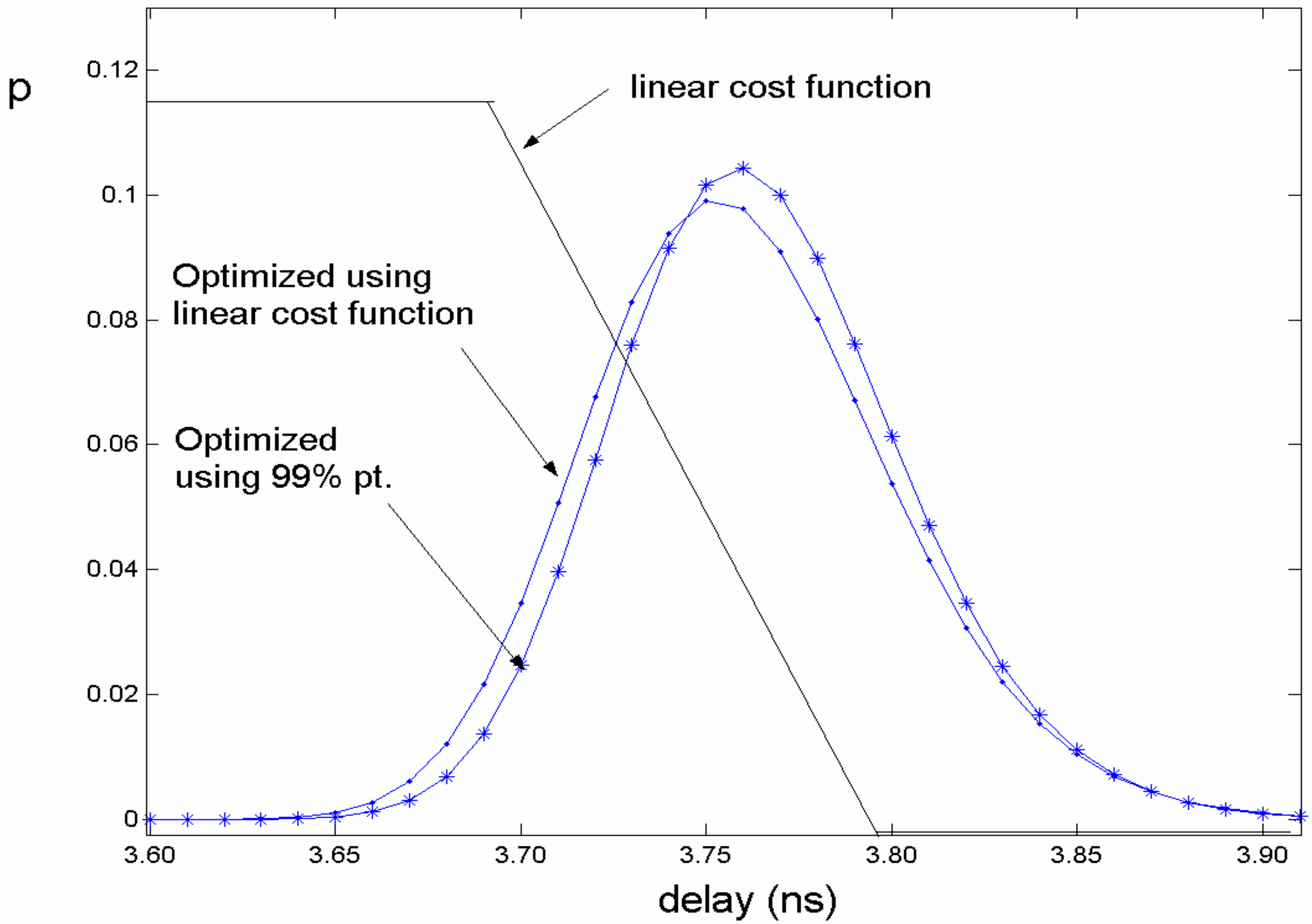
c880 – Optimized circuit delay PDF



Accuracy of bounds – c3540



Cost functions – c880



Summary

- Brute force formulation
- Novel theory of perturbation bounds
- Statistical optimization algorithm
- Exact analysis / reduced complexity -upto 20X run time reduction on benchmarks
- Combined method – upto 89X reduction
- Significant sigma and delay improvement

Thank You