Improving the Survivability of a Simple Evolved Circuit Through Co-evolution

Evolvable Hardware 2004

Michael L. Harrison

University of Idaho Neural-Fuzzy Group

May 7, 2004

University of Idaho

Goals and Hypothesis

Goal: Demonstrate that fault tolerant designs can be targeted by selection.

Hypothesis: Co-evolving faults will improve the fault tolerance of our evolved circuits.

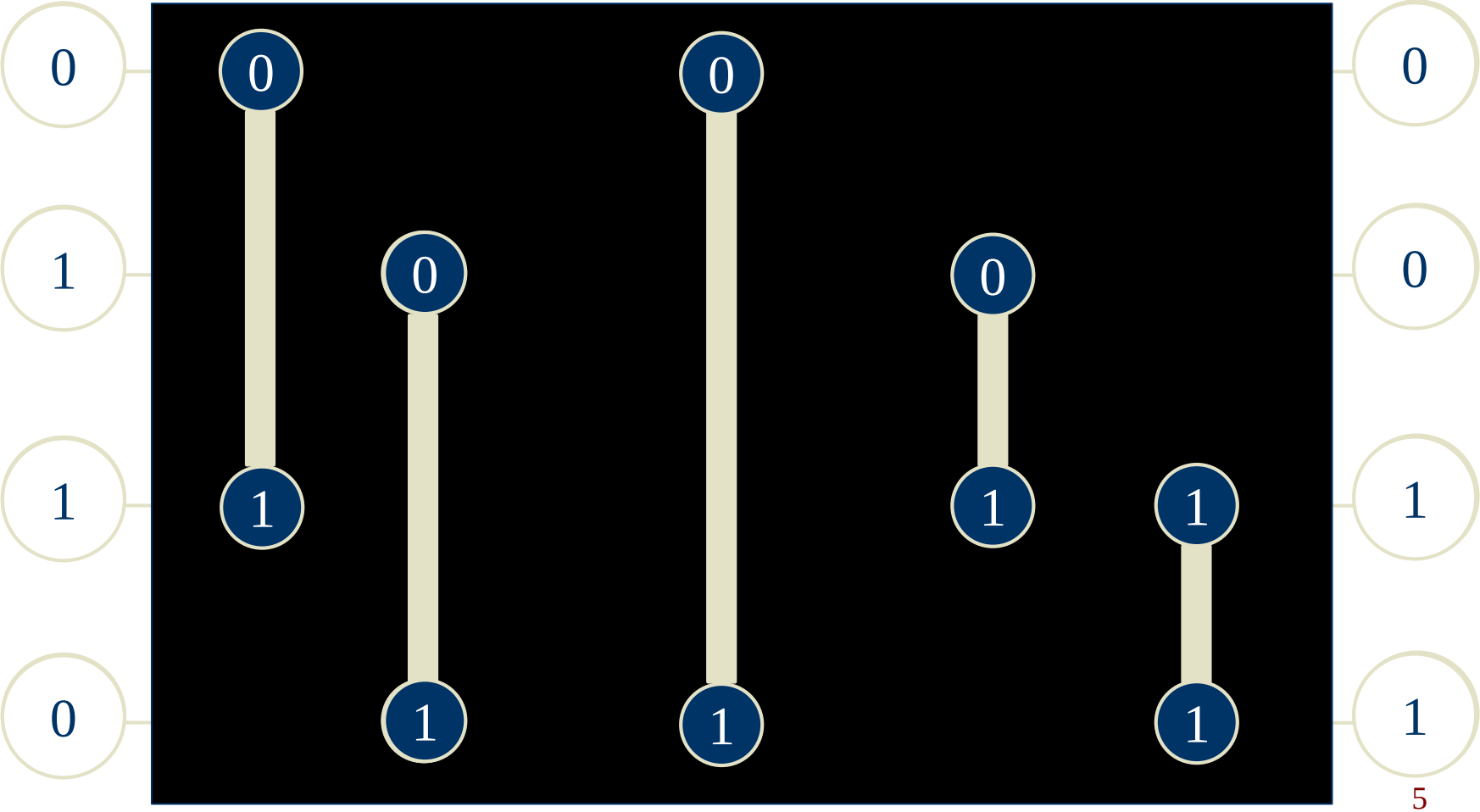
Previous Work

- Fault tolerance as an artifact of the evolution process in sorting networks(Masner, et al.)
  - Tree and array based encodings
  - Network size
- Sorting networks and co-evolution(Hillis)
  - Finding minimal sized sorting networks

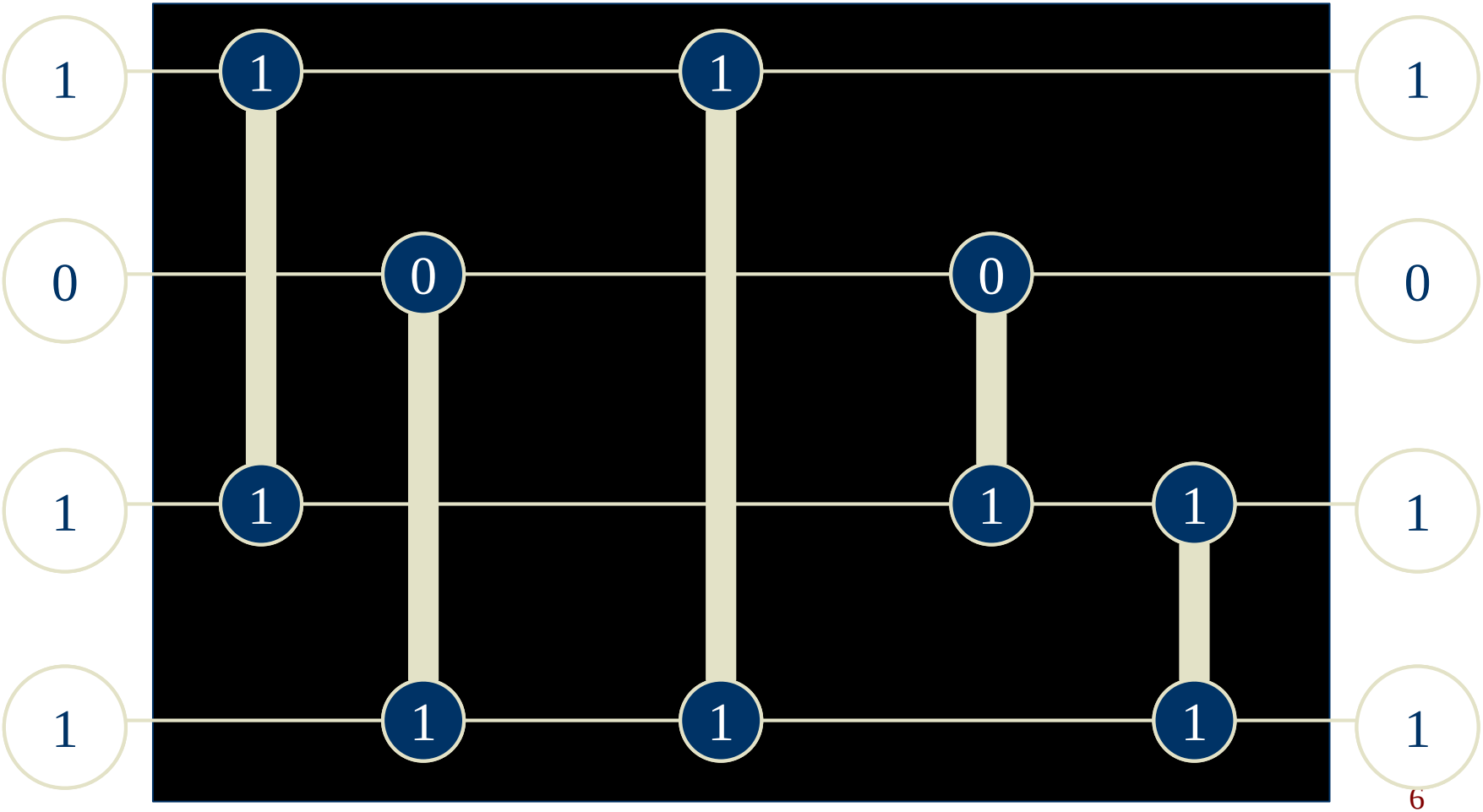
Fault Tolerance & Survivability

- Fault tolerance is a measure of how a system reacts to external or internal pressures.
  - Component failure
  - Hostile environment
- Survivability is an aspect of fault tolerance concerned with the level of service after a fault occurs (spread the risk, improve worst case scenario).

Introduction to Sorting Networks

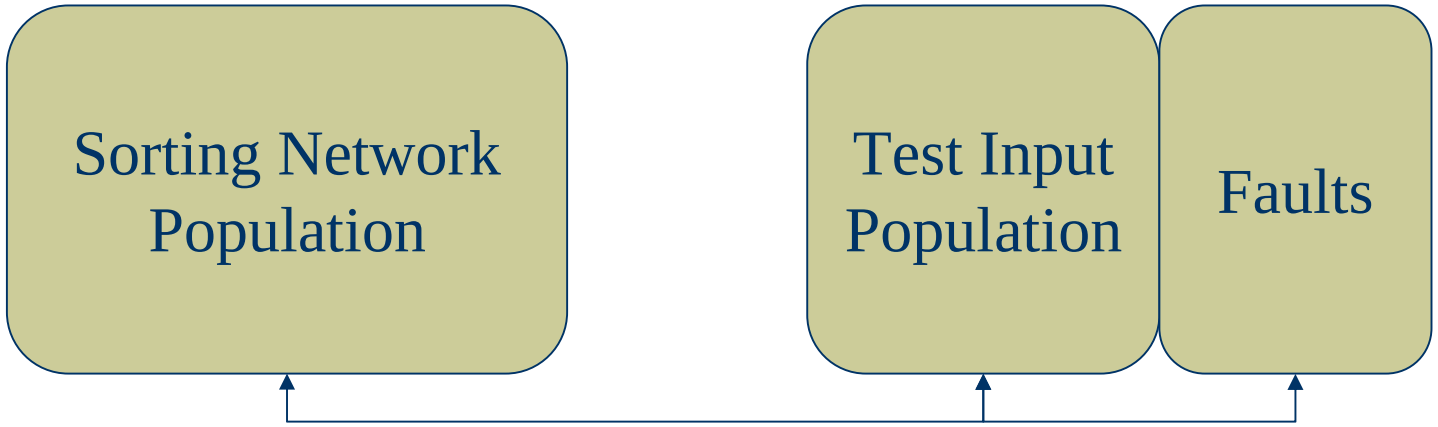


Introduction to Sorting Networks



Co-Evolve the Test Inputs

- Sorting networks selected to sort test inputs
- Test inputs selected to break networks
- Add a fault to each test input
  - Fault is defined as the removal of a single CE-gate.



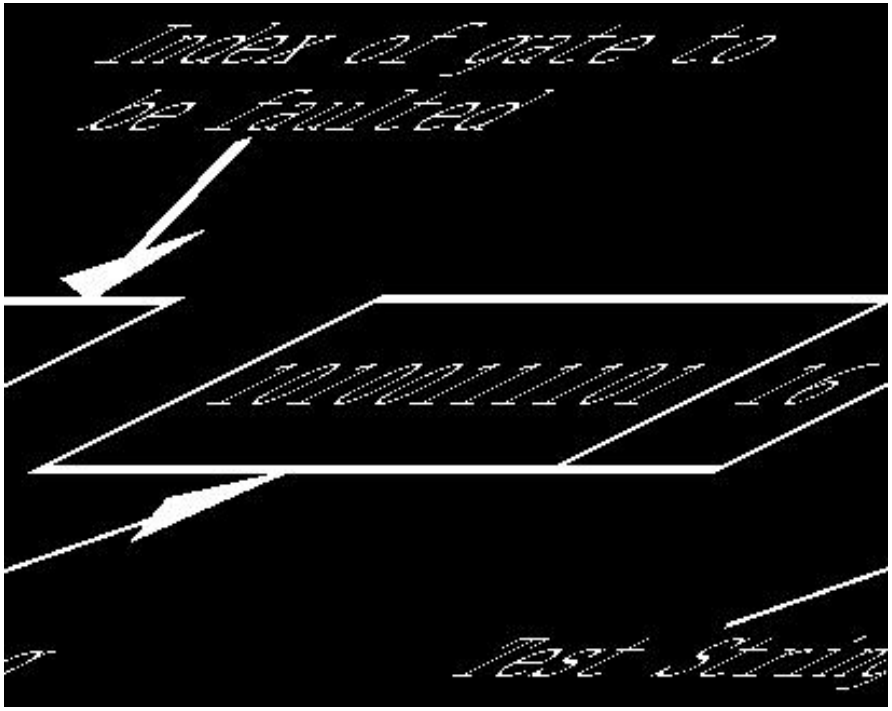
Evolutionary Parameters

Parameter	Value
Network pop. size	15,000
Input string pop. size	100
Selection*(both pops.):	7-way tournament
Mutation rate(both pops.)	0.01
Elitism in network pop.	top 30%

*Selection in the populations is from top 30%, repeated until bottom 70% replaced.

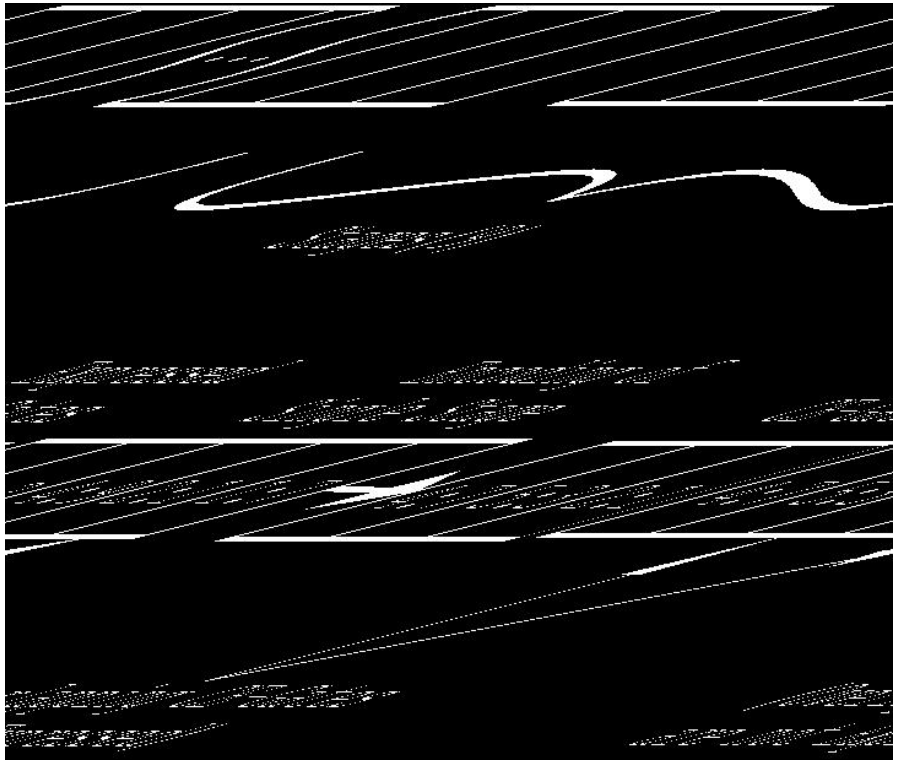
Input String Encoding, Crossover and Mutation

- 2-point crossover along binary string
- Faults randomly assigned to children
- Mutation selects any bit or the fault and reassigns.



Sorting Network Encoding, Crossover and Mutation

- 11 input network
- 45 gates allowed grouped into 9 stages
- 2-point random crossover
- Mutation randomly selected a gate and modifies one input



Evaluation with Faults

Fault Analysis of Sorting Networks

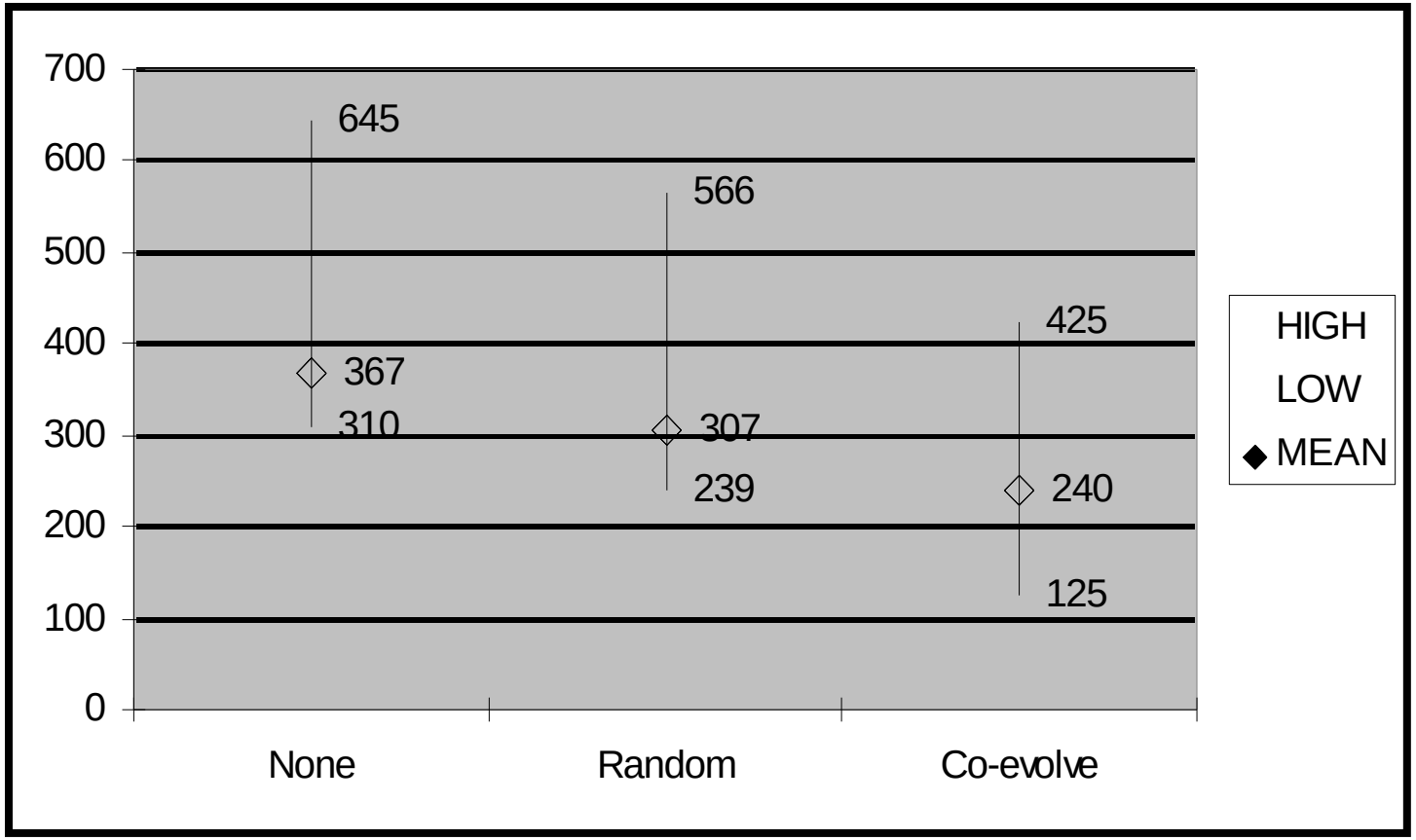
- For each compare-exchange gate in the network
  - Fault the compare exchange gate
  - Do full evaluation (apply all 2048 input strings)
  - Record number of failures (out of position bits in output)

The Experiments

- 3 types of experiments
  - Use co-evolved faults
  - Use random faults
  - Faults not used
- Performed each test 36 times for statistical comparison

Summary Data for Worst Case Single Point Failures

Out of position bits



Summary Data for Summed Faults

Sum of out of position bits

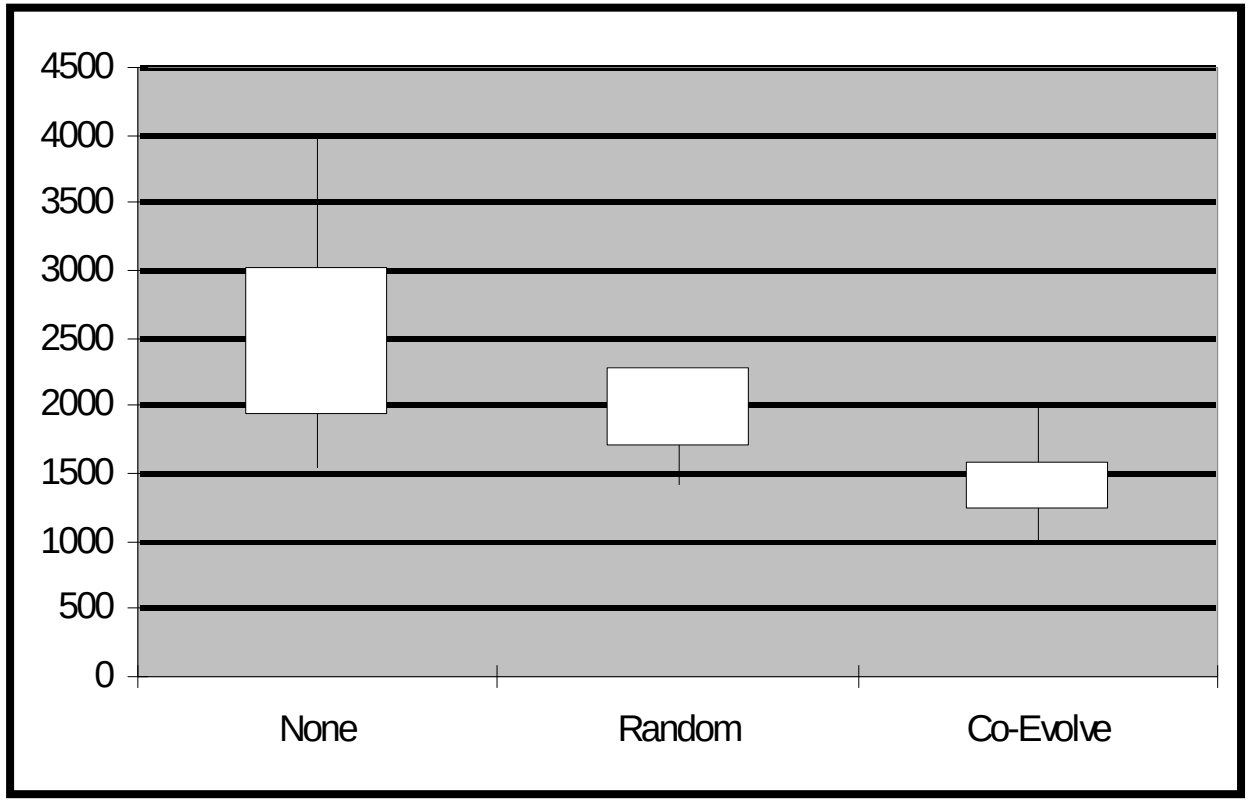
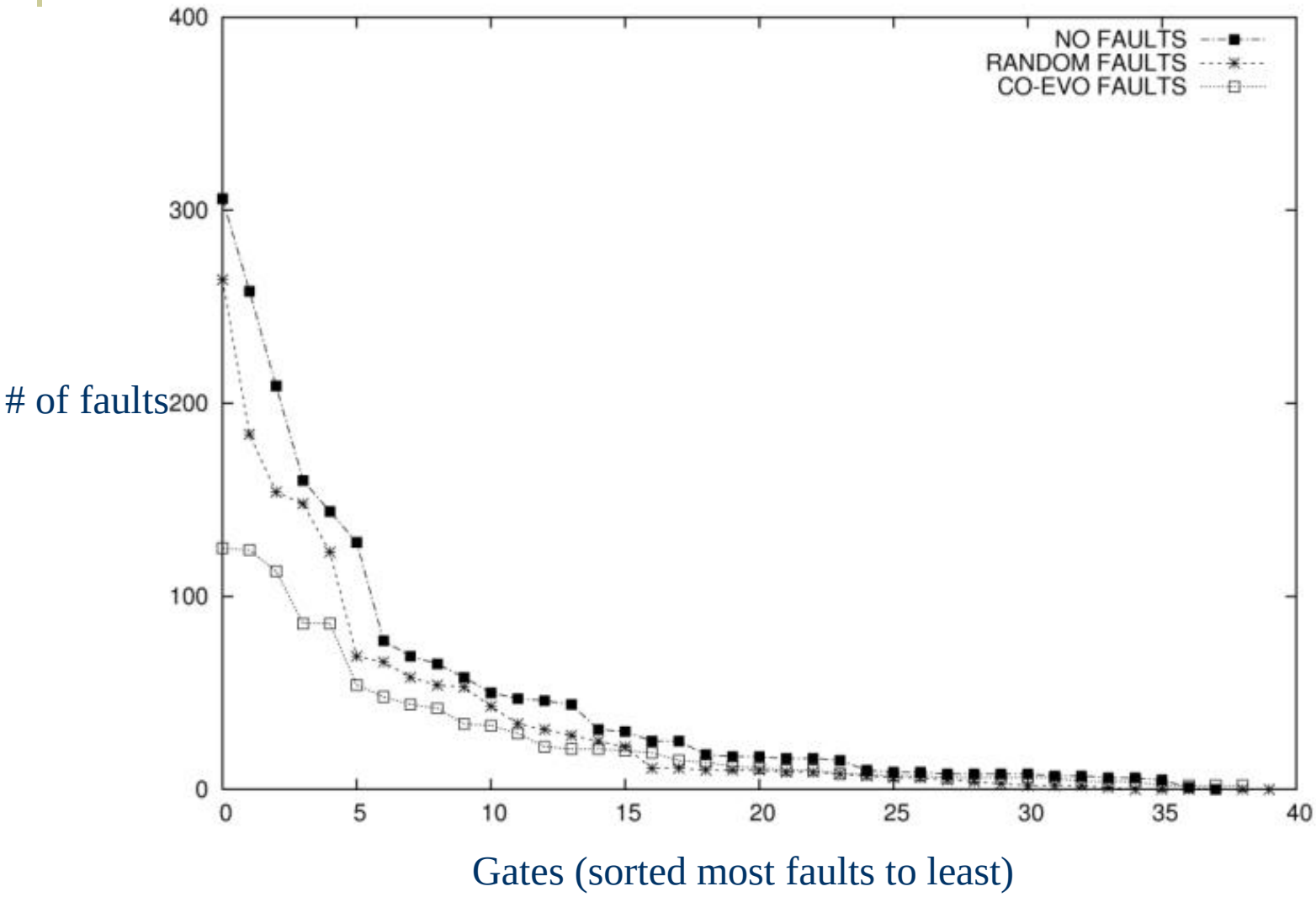


Chart shows summary data across runs. Boxes show 1 standard deviation from the mean and whiskers show high and low ranges.

Comparison of the best



Conclusions

- Significant improvement in overall fault tolerance of sorting networks (expected failure rate)
- Reduction of failure rate for worst case scenario
- Improvement correlates with degree of fault pressure

Future Work

- Extend to realistic circuits
  - Neural networks, etc.
- Realistic fault models
- Tune the co-evolution process
  - Evolutionary parameters (population sizes, various rates)
- Other areas of interest:
  - Fault tolerant scheduling
  - Fault tolerant system architectures

Acknowledgements

- Jason Masner (M.S., 2000)
- John Cavalieri (B.S., 2002)
- Rob Shepherd (M.S., 2003)
- Discussions with James Foster, Rick Wells, Terry Soule, and the entire Neuro Fuzzy group
- Funding from NSF/EPSCoR

Questions?

Why Sorting Networks

Pros

- Easier to implement understandable fault models than in more realistic circuits.
- Components are independent

Cons

- Very simple (too simple?)
- Hard to evolve large topologies
- Everyone thinks we are looking for minimal ones!