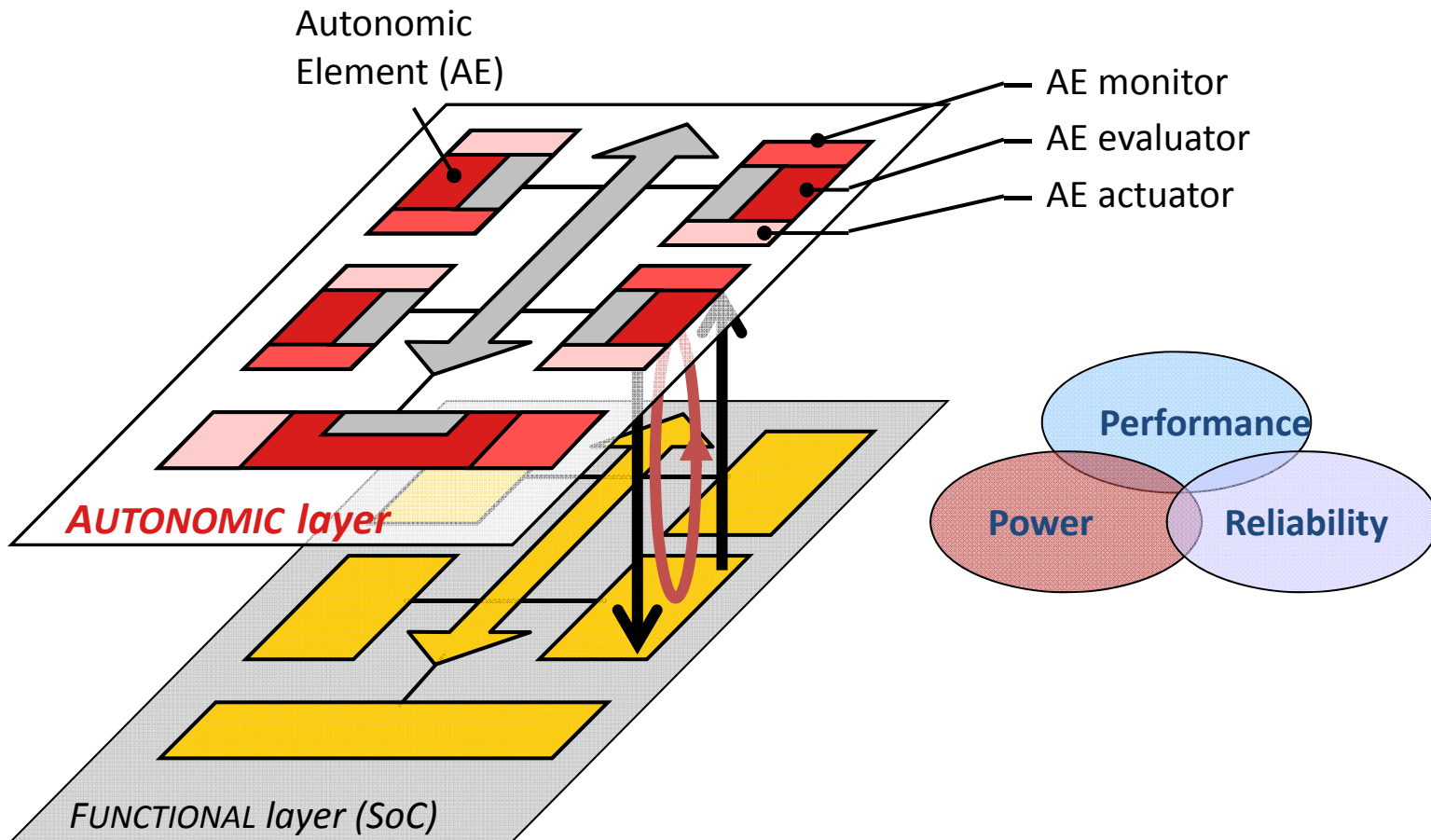


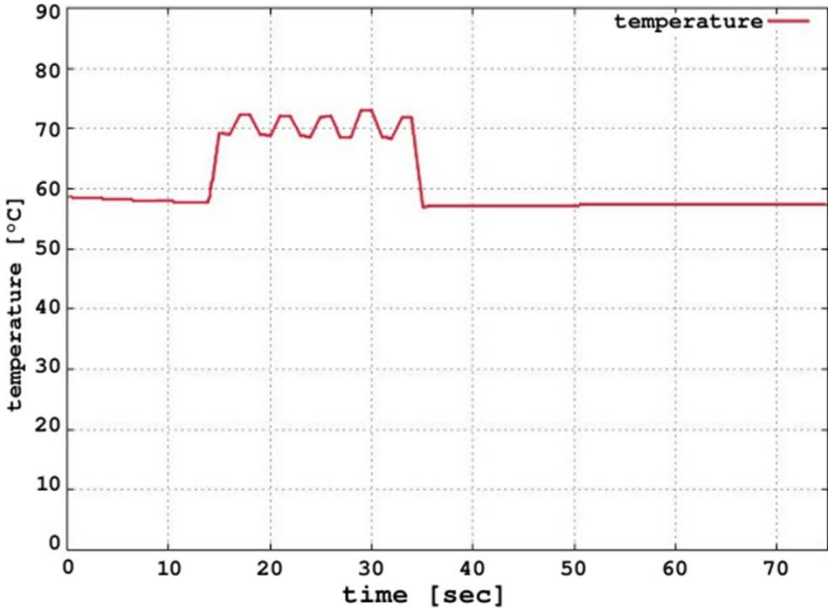
XCS-based design- and run-time learning

Andreas Bernauer
February 22, 2010

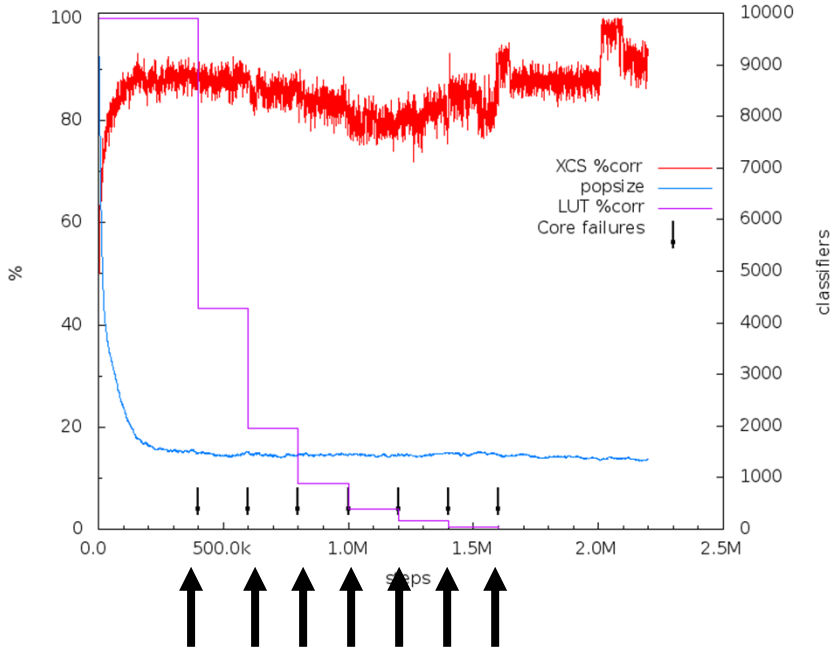
ASoC



XCS as Evaluator

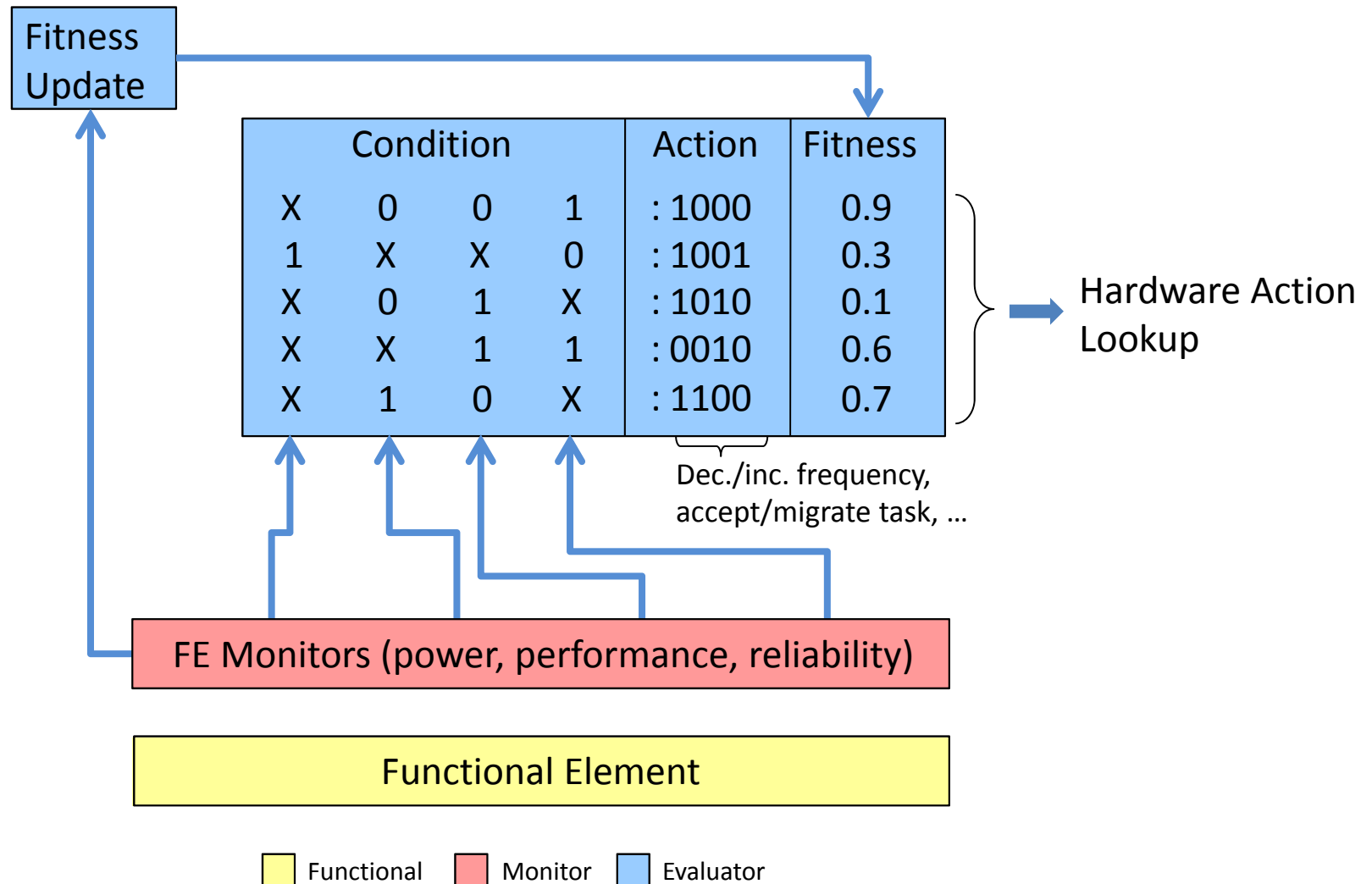


Controlling a SoC

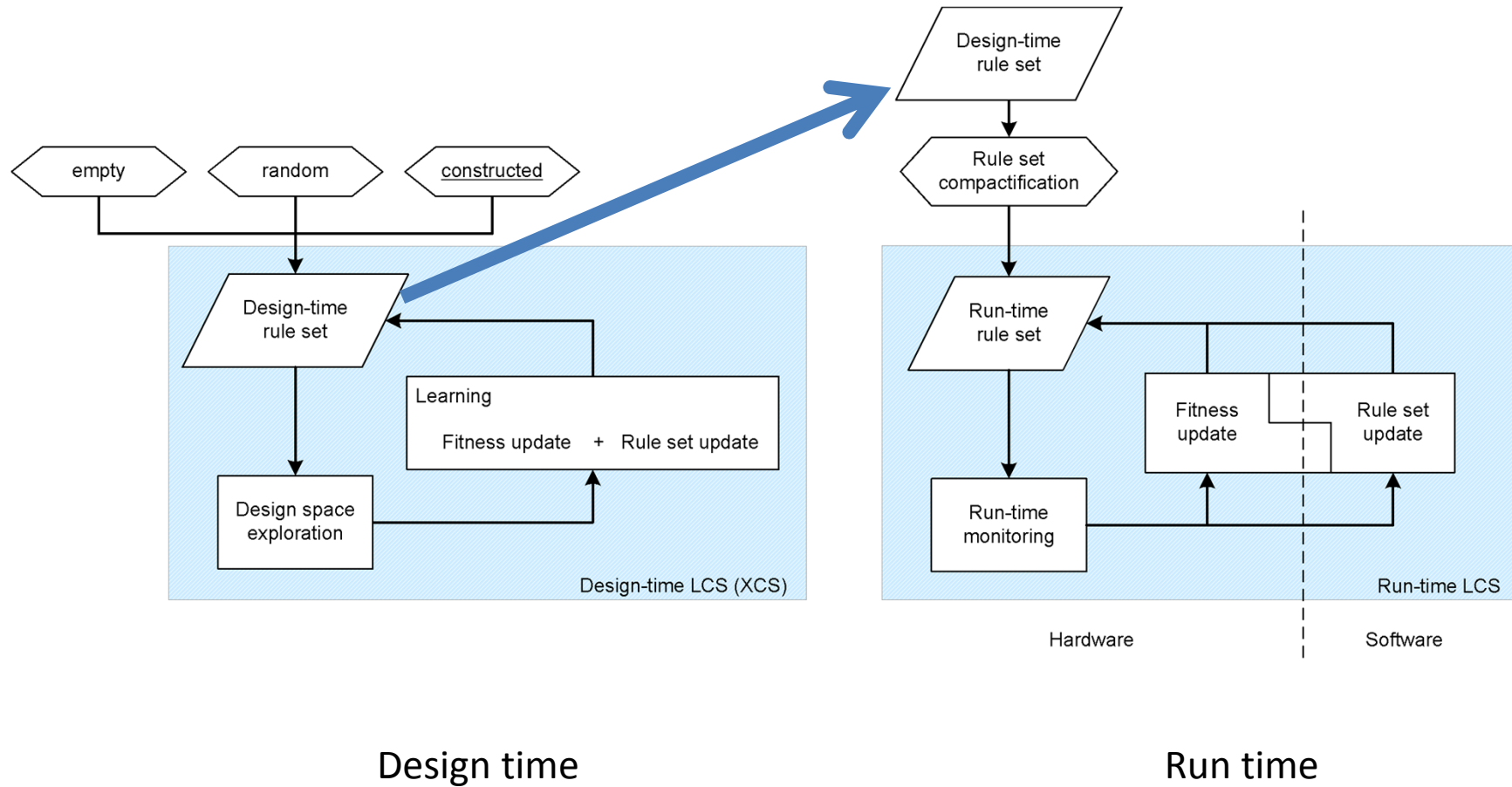


Controlling task allocation

Learning Classifier Table



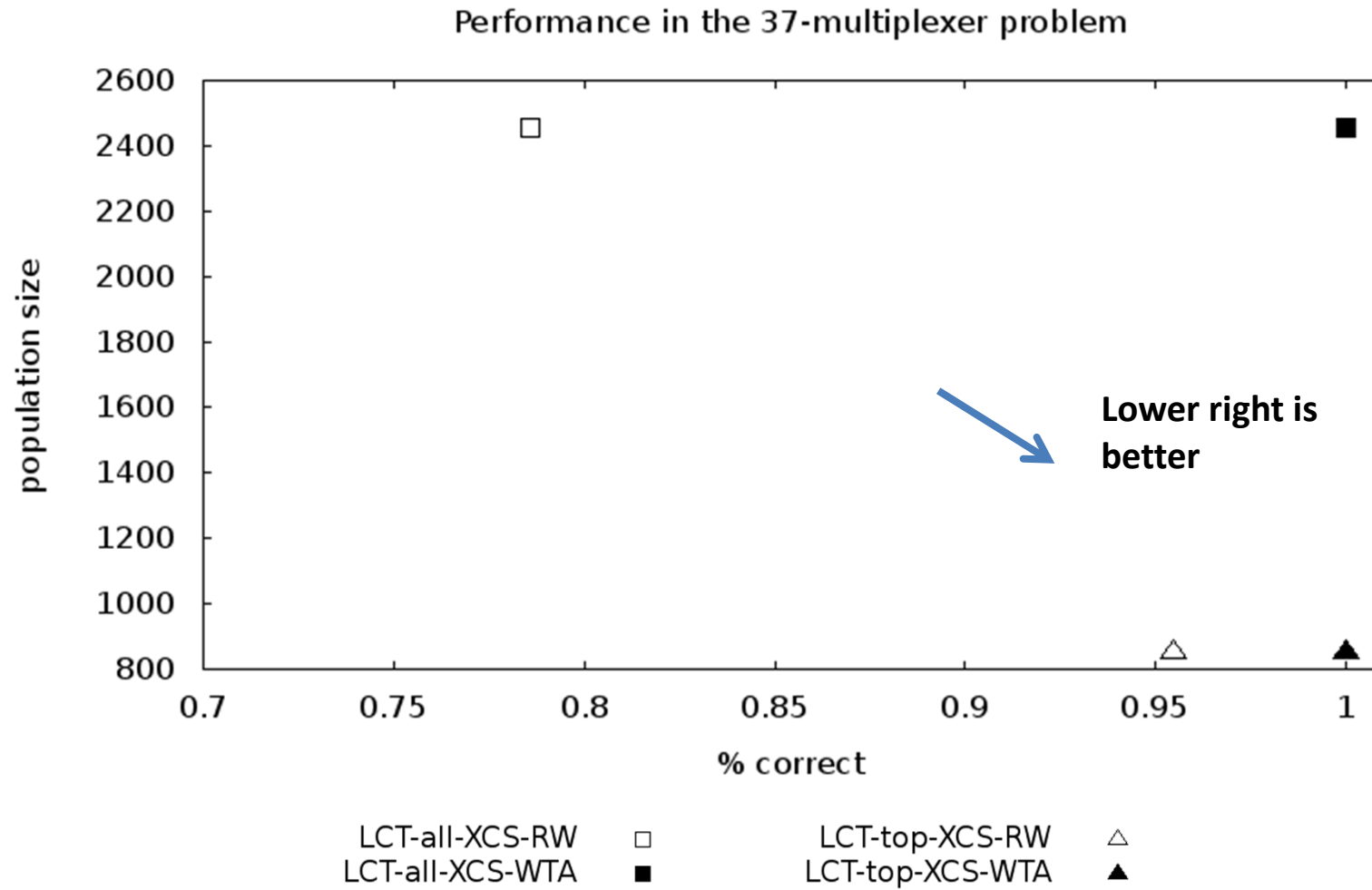
Combining XCS and LCT



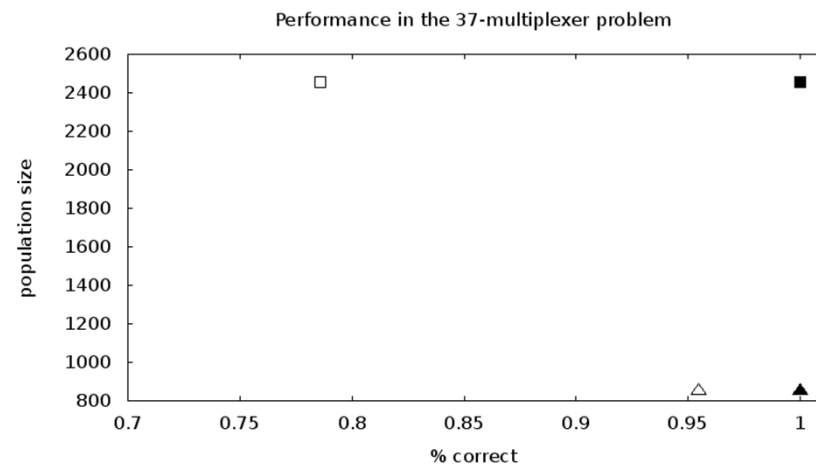
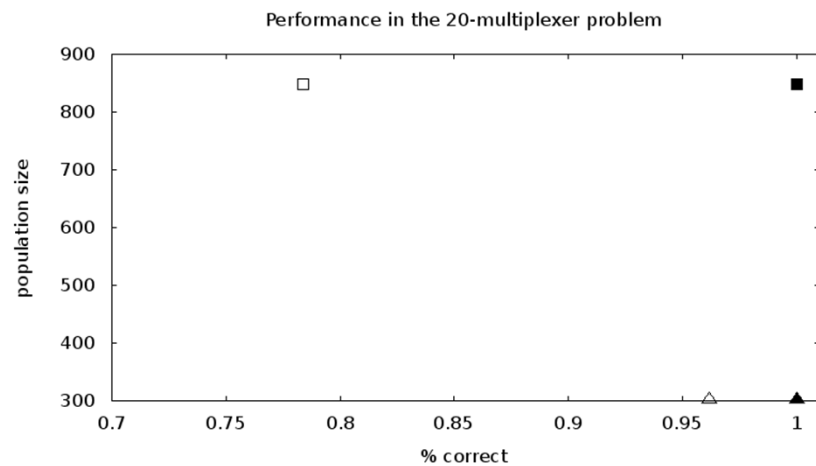
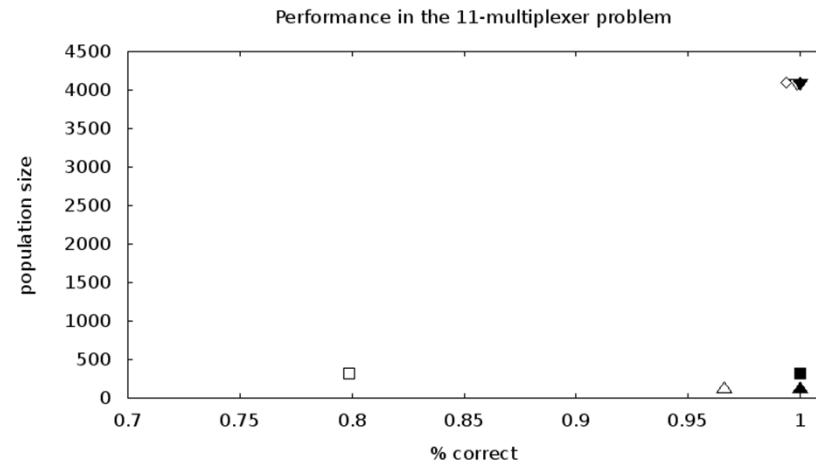
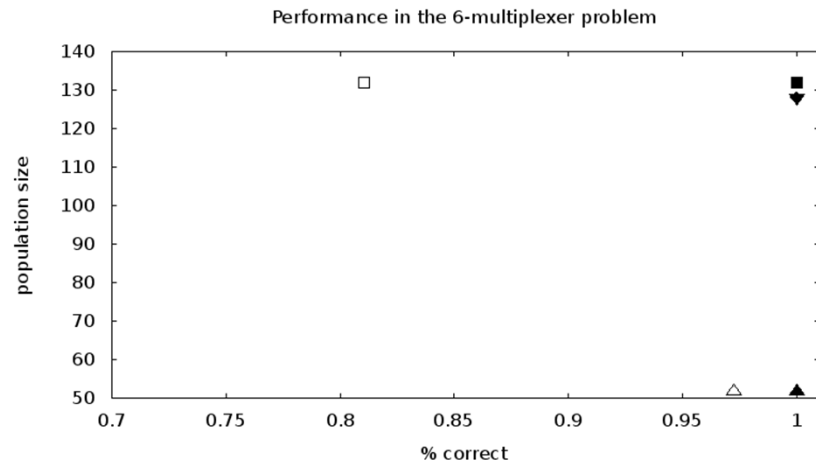
Configurations

- all-XCS: translate all XCS rules to LCT rules
- top-XCS: translate only top XCS rules
- roulette-wheel: randomly, reward-weighted
- winner-takes-all: highest reward

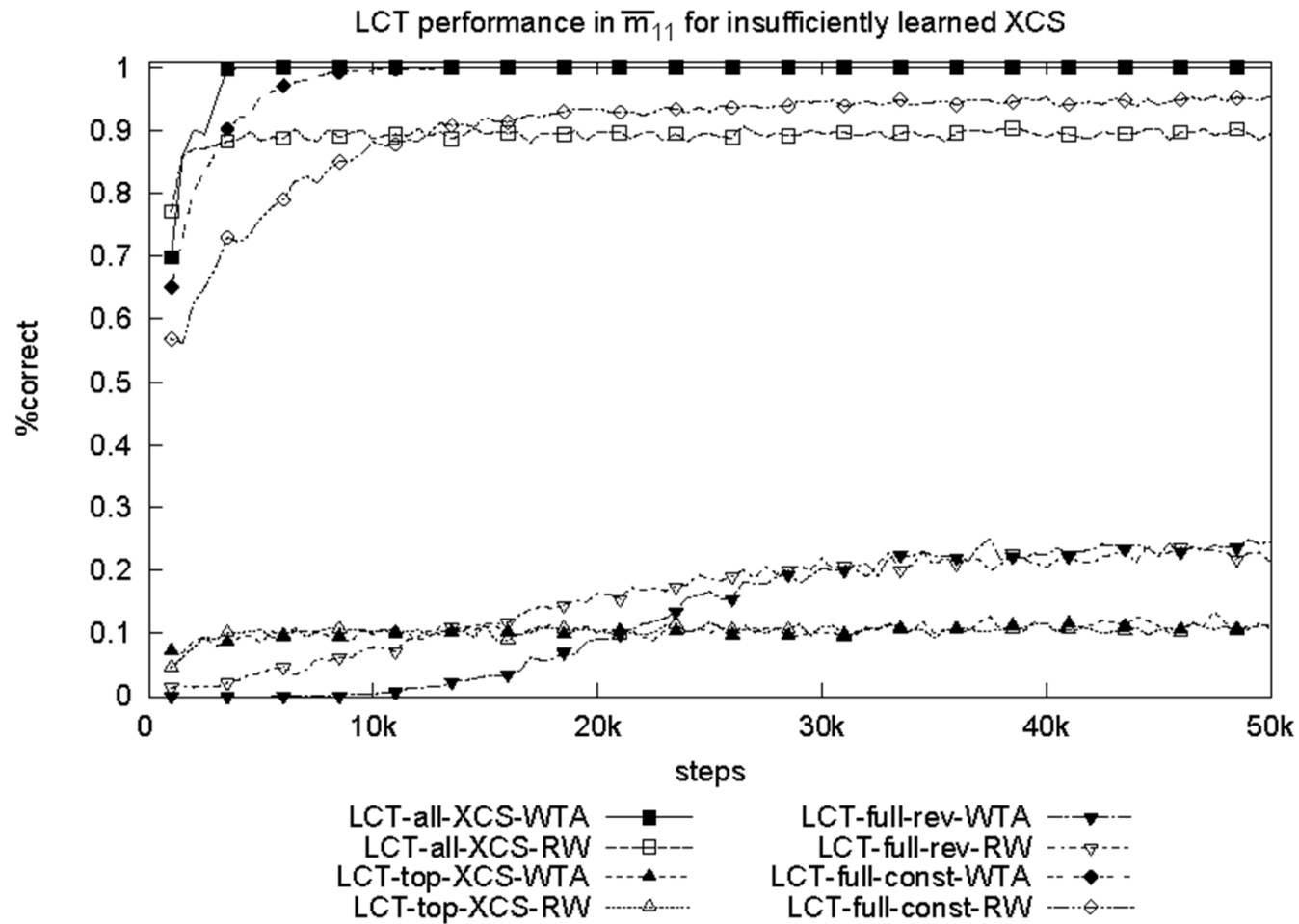
Multiplexer benchmark



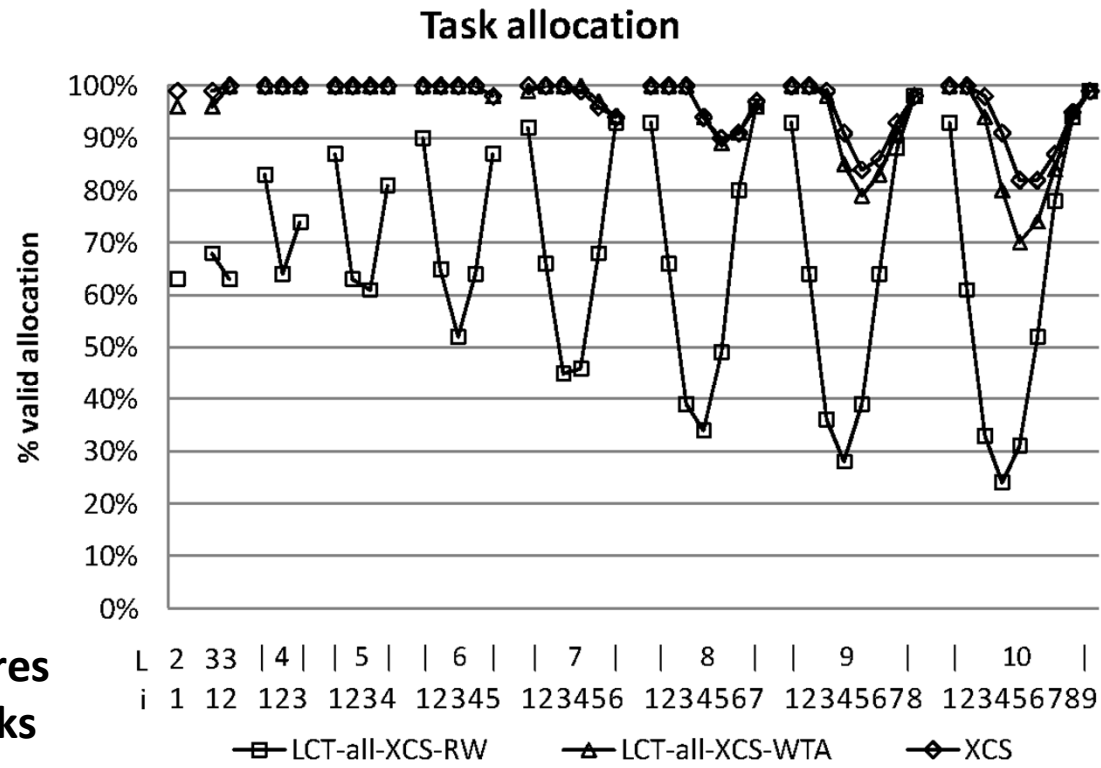
Multiplexer benchmark



Inverted multiplexer

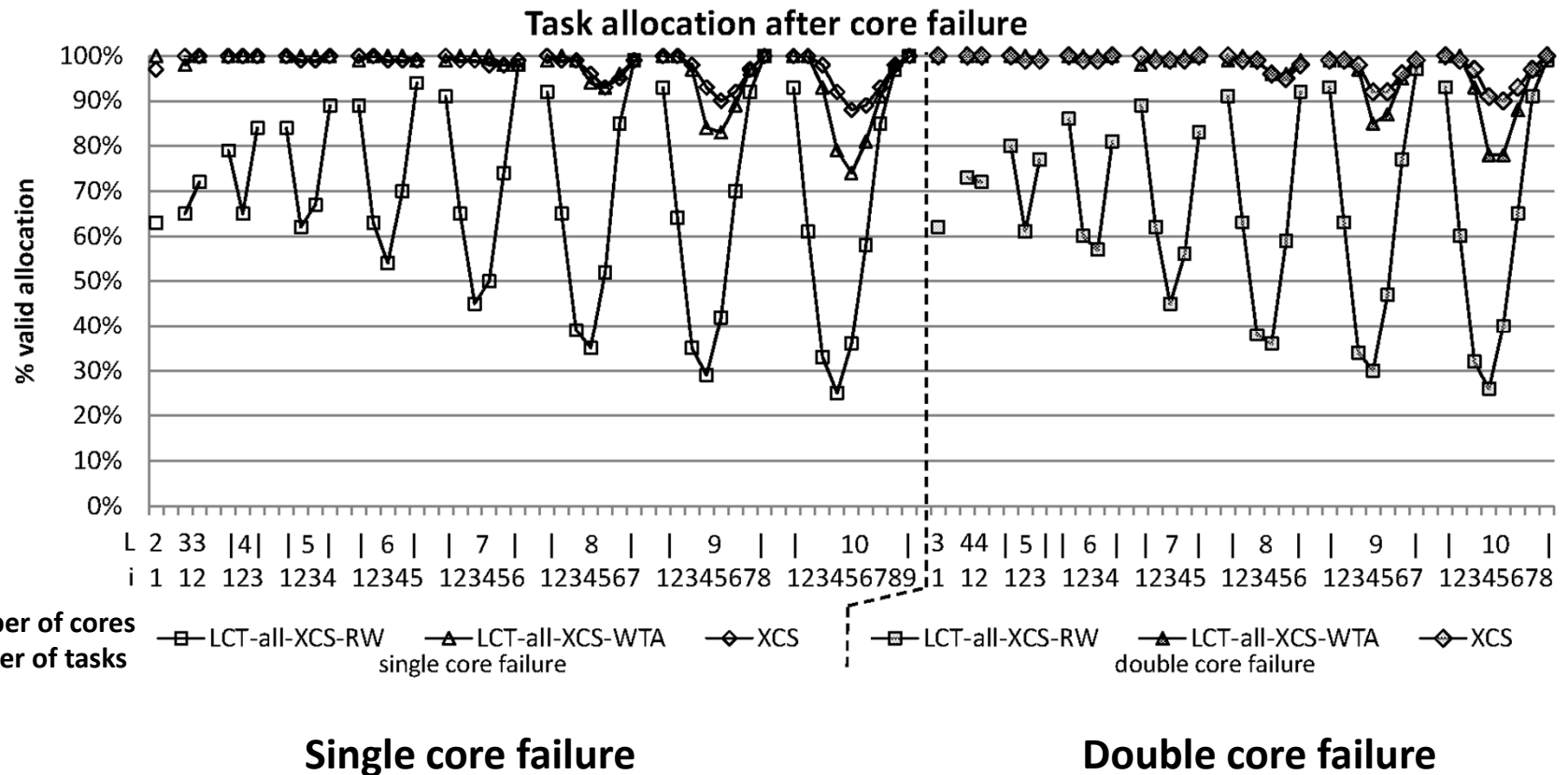


Task allocation

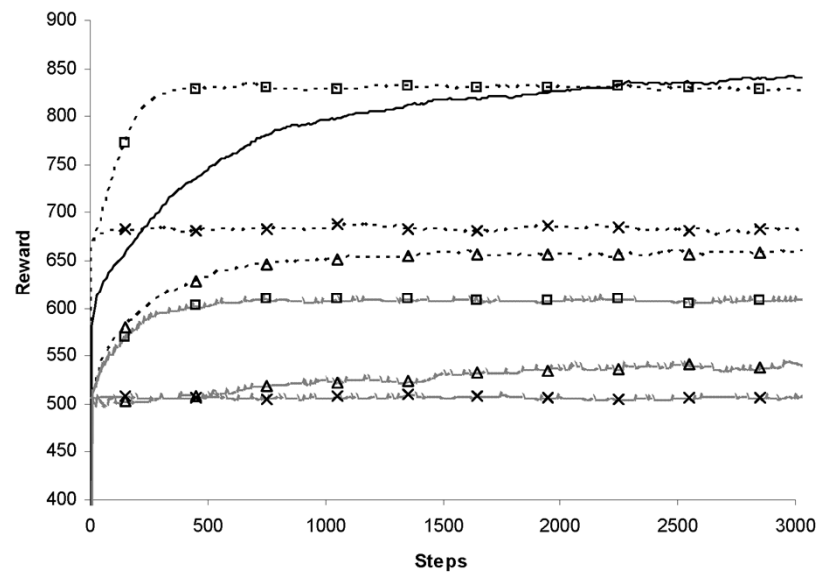


L: Number of cores
i: Number of tasks

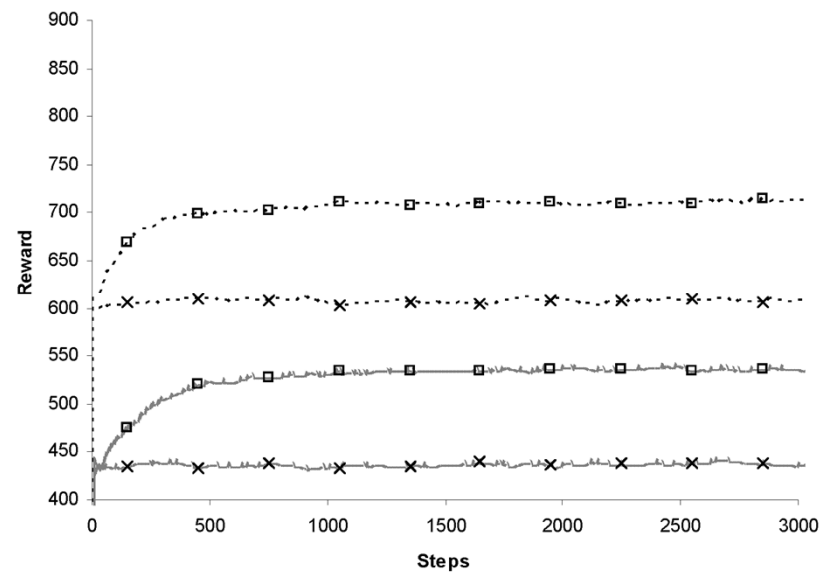
Task allocation after core failure



SoC component parameterization



Before malfunction



After malfunction

— XCS - - - □ - - - LCT-all-XCS-WTA - - - x - - - Static all-XCS-WTA - - - Δ - - - LCT-full-const-WTA - - - □ - - - LCT-all-XCS-RW - - - x - - - Static all-XCS-RW - - - Δ - - - LCT-full-const-RW

Summary

- Learning Classifier System can be used as Autonomic Evaluator.
- We can learn the classifiers in software (XCS) and use them in hardware (LCT).
- Winner-takes-all strategy is preferable.
- Translating all XCS rules retains capability to adapt to unforeseen events (core failures, CPU malfunction).

Call for Participation

3. LCS-Miniworkshop

- Recent Advances in LCS and Learning
- Formal Analysis, Constraints, and Garanties

- When: Friday, July 2, 2010
- Where: University of Tübingen