Evolutionary Computing in Approximate Circuit Design and Optimization

Lukáš Sekanina and Zdeněk Vašíček
Faculty of Information Technology
Brno University of Technology
sekanina@fit.vutbr.cz

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

WAPCO 2015, Amsterdam
January 19, 2015

IT4Innovations national supercomputing center

How to approximate digital circuits?

- Technology-oriented techniques
  - voltage over-scaling, over-clocking …
- Functional approximation
  - Original function F is replaced by G whose implementation leads to
    - energy/delay/area reduction
    - non-zero error

Motivation

- Methods for functional approximation
  - Manual
  - Automatic (= some heuristics used)
    - SALSA (DAC 2012)
    - SASIMI (DATE 2013)
    - ABACUS (DATE 2014)
    - ASLAN (DATE 2014)
    - …
    - **Evolutionary algorithm (EA)-based methods**

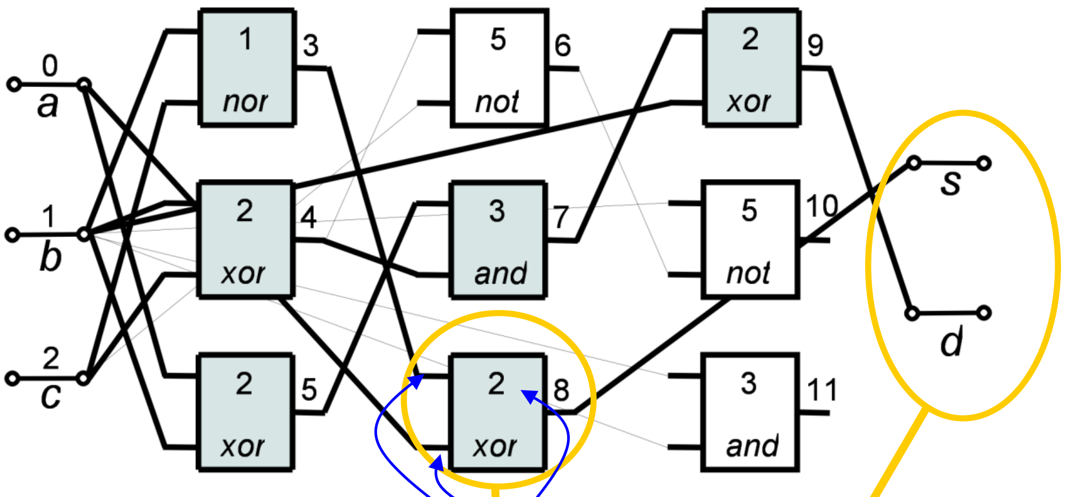
      Hypothesis: Much better approximations can be discovered than conventional methods can provide.

Why evolutionary approximation?

- It is natural!
  - AC: partially working circuits are sought
  - EA: genetic improving of partially working circuits
- EAs are excellent in multi-objective design
- Constraints are easily handled.
- The original (accurate) circuit is not required.

- Problems of evolutionary design:
  - scalability
  - runtime

Cartesian genetic programming (CGP) [Miller JF 1999]



1,2,1; 1,2,2; 0,1,2; 4,2,5; 5,4,3; 3,0,2; 7,1,2; 1,6,5; 1,1,3; 8,9

- Example: CGP parameters
  - n_r=3 (#rows)
  - n_c = 3 (#columns)
  - n_i = 3 (#inputs)
  - n_o = 2 (#outputs)
  - n_a = 2 (max. arity)
  - L = 3 (level-back parameter)
  - Γ= {NAND$^{(0)}$, NOR$^{(1)}$, XOR$^{(2)}$, AND$^{(3)}$, OR$^{(4)}$, NOT $^{(5)}$}

NETLIST = GENOTYPE

Typical fitness function (circuit functionality):

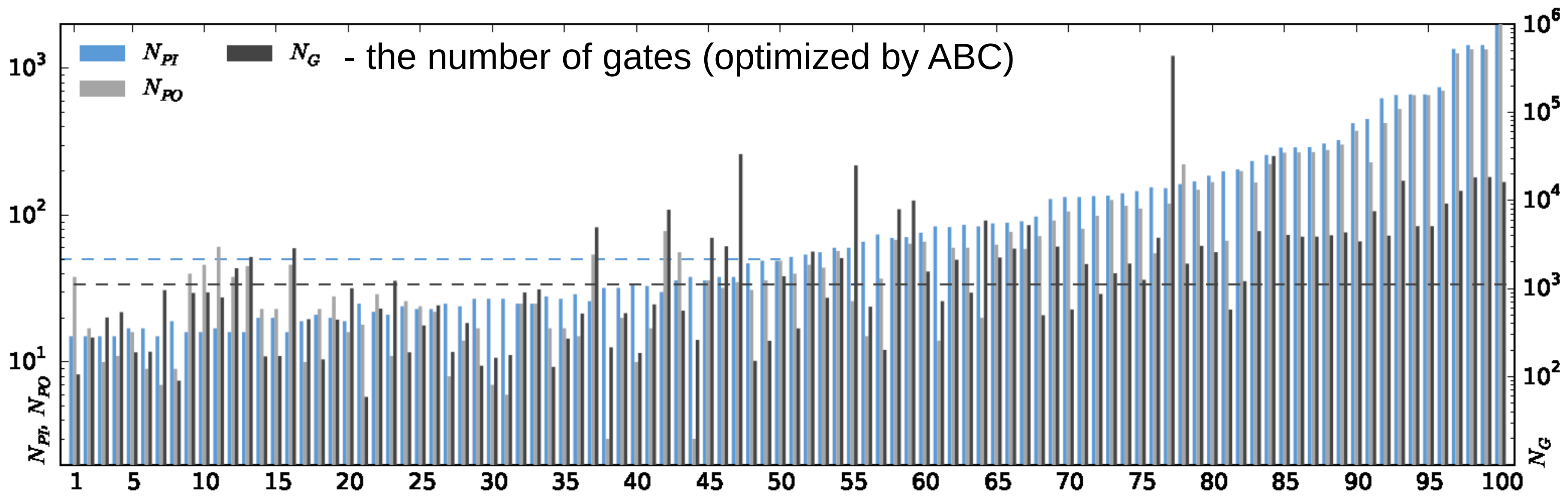
$$f = \sum_{i=1}^{K} |yi - wi|$$

Number of test vectors

Desired response

Circuit response

Example evolutionary optimization (no approx.)



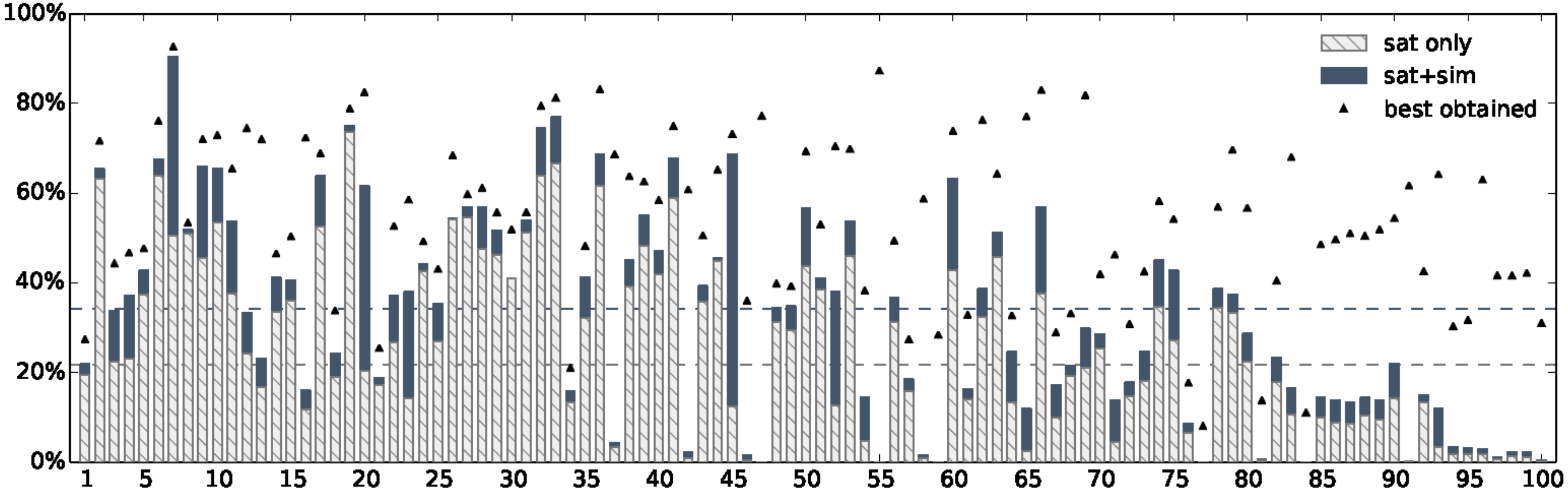
100 combinational circuits (≥15 inputs) - IWLS2005, MCNC, QUIP benchmarks

Heavily optimized by ABC

1: alcom (N_G = 106 gates; N_{PI} = 15 inputs; N_{PO} = 38 outputs)

100: ac97ctrl (N_G = 16,158; N_{PI} = 2,176; N_{PO} = 2,136)

Minimization of the number of gates



CGP + SAT solver + circuit simulation

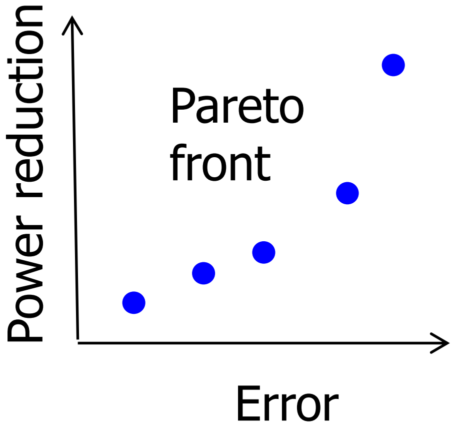
Y-axis: Gate reduction w.r.t. ABC after 15 minutes, 34% on average

▲ Gate reduction w.r.t. ABC after 24 hours

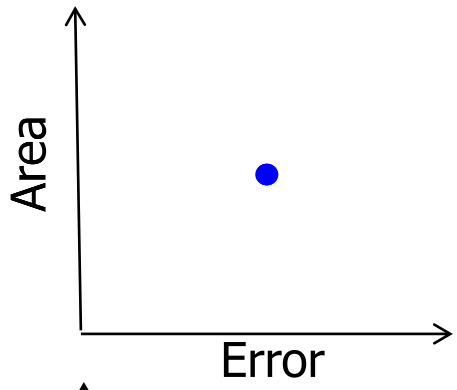
[Vašíček, Sekanina: Genetic Programming and Evolvable Machines 12(3), 2011; DATE 2011; EuroGP 2015]

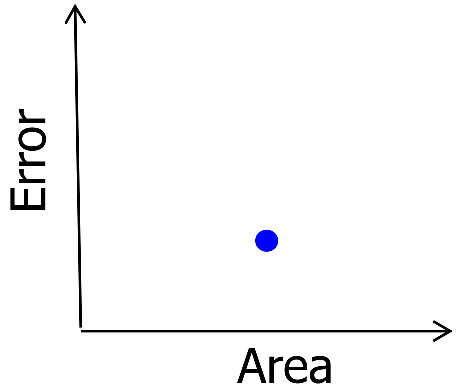
Much better results expected for approximate circuits!

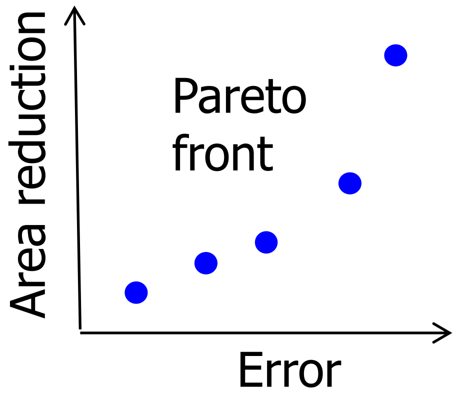
- Approximate circuit design problem is seen as a search problem.
  - Power consumption, error, area and delay are conflicting design objectives
  - Single-objective CGP: *n* runs are needed to obtain *n* points on the Pareto front
  - Multi-objective CGP: Pareto front is the result of a single CGP run.
- Methodology:
  - It is assumed that power consumption is highly correlated with the area. Hence only the area is estimated in the fitness function (which is fast).
  - Power consumption is calculated for evolved circuits at the end of evolution.

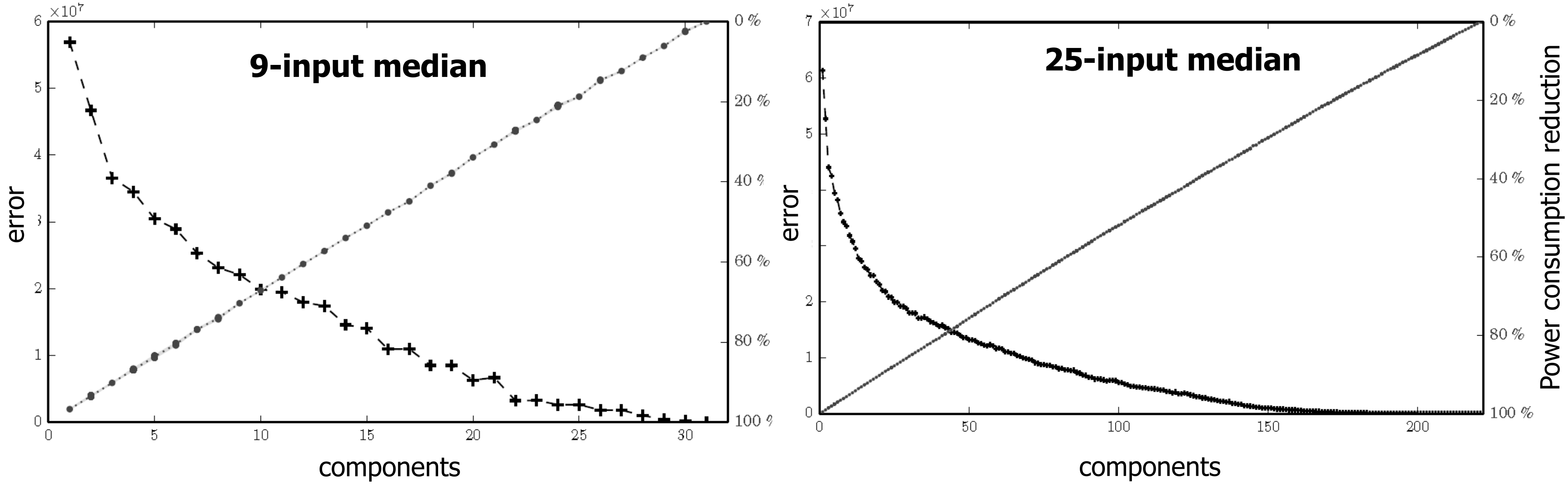


- Error-oriented (single-objective) method
  - CGP gradually degrades a fully functional circuit until a circuit with a <u>required error</u> is obtained. Then, the area (and so power consumption) is minimized for this error.
- Resources-oriented (single-objective) method
  - CGP is used to minimize the error, but only limited resources (components) are provided, insufficient for constructing a fully functional circuit.
- Multi-objective optimization
  - Area, delay, and error are optimized together.









Vasicek, Sekanina: IEEE Tr. on Evol. Comp, 2015 – in press

The evolved correct median circuits

		power [mW]			area [-]			delay [ns]		
w	n_c	best	worst	mean	best	worst	mean	best	worst	mean
9	31	10.8	12.9	12.6	2314.2	2836.7	2750.8	285.9	429.7	295.1
25	221	72.4	72.4	72.4	16497.7	16497.7	16497.7	539.5	539.5	539.5

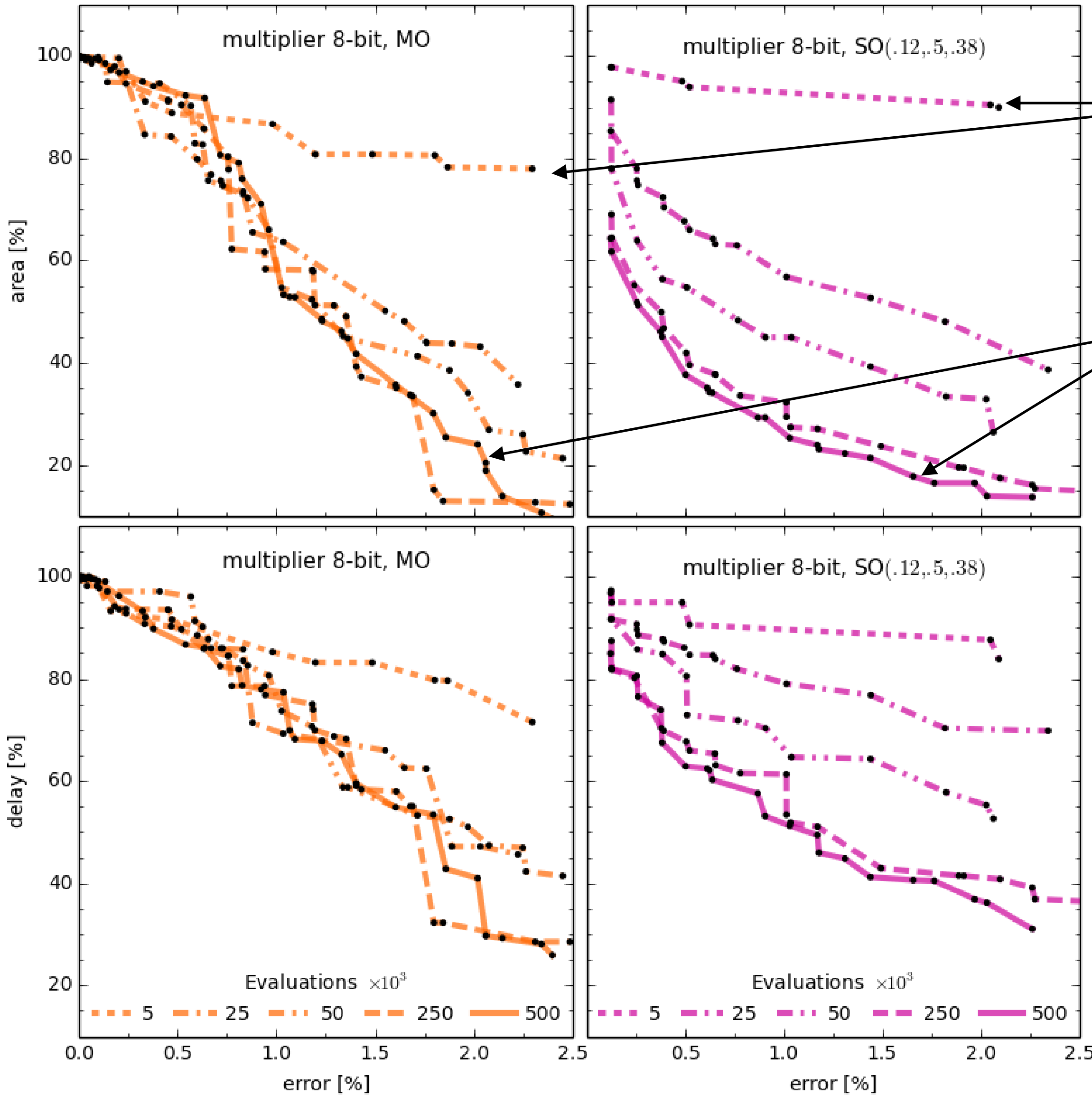
reference solutions (power consumption estimated using SIS)

Multi-objective CGP: 8-bit multiplier

- 3 criteria
  - average error (if E >2.5%, the solution is not accepted)
  - relative area
  - delay
- First scenario: single-objective CGP with weight criteria
  - $(w_{error}; w_{area}; w_{delay}) = (0.12; 0.5; 0.38)$
  - $\lambda = 5$; 50 generations; 20 runs (20 error levels) $\Rightarrow$ 5000 evaluations
- Second scenario: multi-objective CGP (NSGA-II)
  - $\lambda = 50$, 100 generations $\Rightarrow$ 5000 evaluations

Multi- vs. single-objective CGP: 8 bit multiplier

MO SO



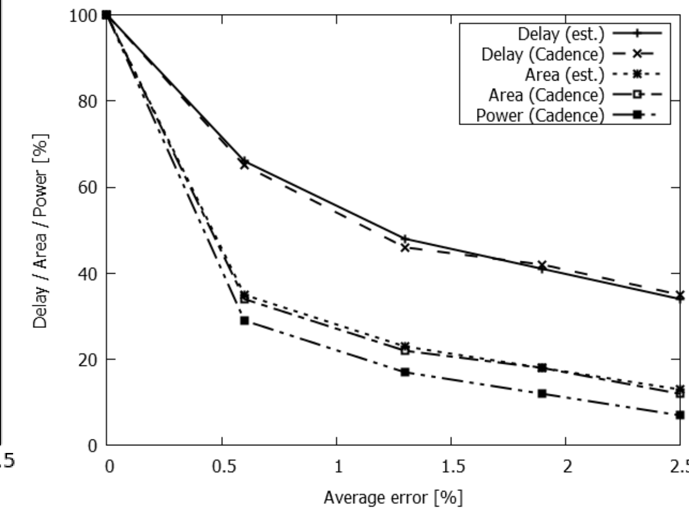
5k evaluations:

MO is better than SO

500k evaluations:

SO is better than MO

Circuit parameters validation (I3T25, 0.35 um)



Unfair comparison: Approximate 8-bit multiplier

Approach (w=8)	Error	Power reduction	CGP
SALSA	$E_{wst} = 10\%$	$P_{impr} = \mathbf{80\%}$	$E_{wst} = 10\%$, $P_{impr} =$ **96%**
SASIMI	$E_{avg} = 0.5\%$	$P_{impr} =$ **45%**	$E_{avg} = 0.5\%$, $P_{impr} =$ **79%**
Kulkarni et al.	$E_{avg} = \mathbf{3.25\%}$, $E_{wst} = \mathbf{22.2\%}$	$P_{impr} = 33\%$	$E_{avg} = \mathbf{0.12\%}$, $P_{impr} = 32\%$ $E_{wst} = \mathbf{0.63\%}$

- Problems:
  - Different original (accurate) multipliers.
  - Different fabrication technology.
  - SW implementations of the methods are not available.
- Why is CGP good? Conventional methods generate and evaluate only several candidate solutions while CGP produces thousands of candidate solutions.

Conclusions

- Approximate circuit design can be formulated as multi-objective search problem and solved by EA
  - The EA methods automatically provide Pareto fronts and probably much better compromised solutions than the state of the art methods.
  - The EA methods estimate key circuit parameters during the evolution to accelerate the whole design process.
- Benchmark problems (with reference results) are needed!
- Future research
  - scalability issues, computation time reduction
  - Applying EAs in other domains of approximate computing

Thank you for your attention!

Lukáš Sekanina and Zdeněk Vašíček
Faculty of Information Technology
Brno University of Technology
sekanina@fit.vutbr.cz

BRNO UNIVERSITY OF TECHNOLOGY

FIT FACULTY OF INFORMATION TECHNOLOGY

IT4Innovations national01$#&0 supercomputing center@#01%101