Amoeba-Inspired System Controller for IoT Edge

Anh Hoang Ngoc Nguyen
Tokyo Institute of Technology, Japan
anh@cad.ict.e.titech.ac.jp

Yuko Hara-Azumi
Tokyo Institute of Technology, Japan
hara@cad.ict.e.titech.ac.jp

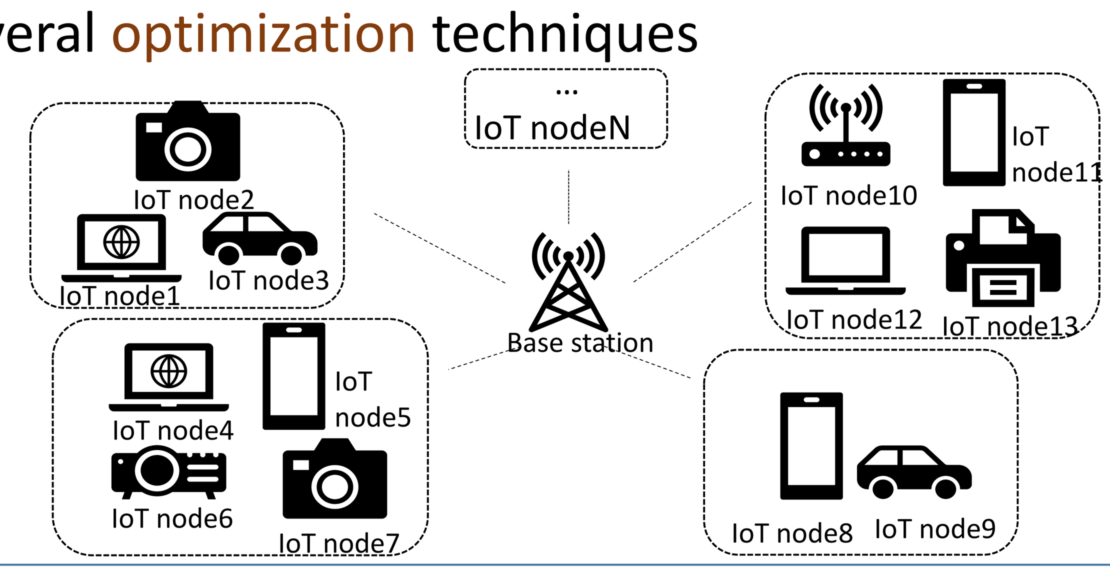
Overview

❑ Motivation
- Needs of real-time control/decision making in IoT systems -> shift from cloud to edge controllers
- Control rules from IoT systems (5G networks, etc.) can be efficiently represented by a Boolean Satisfiability (SAT) problem
- Edge devices are resource-constrained

=> Light-weight SAT-based IoT system controllers on edge devices

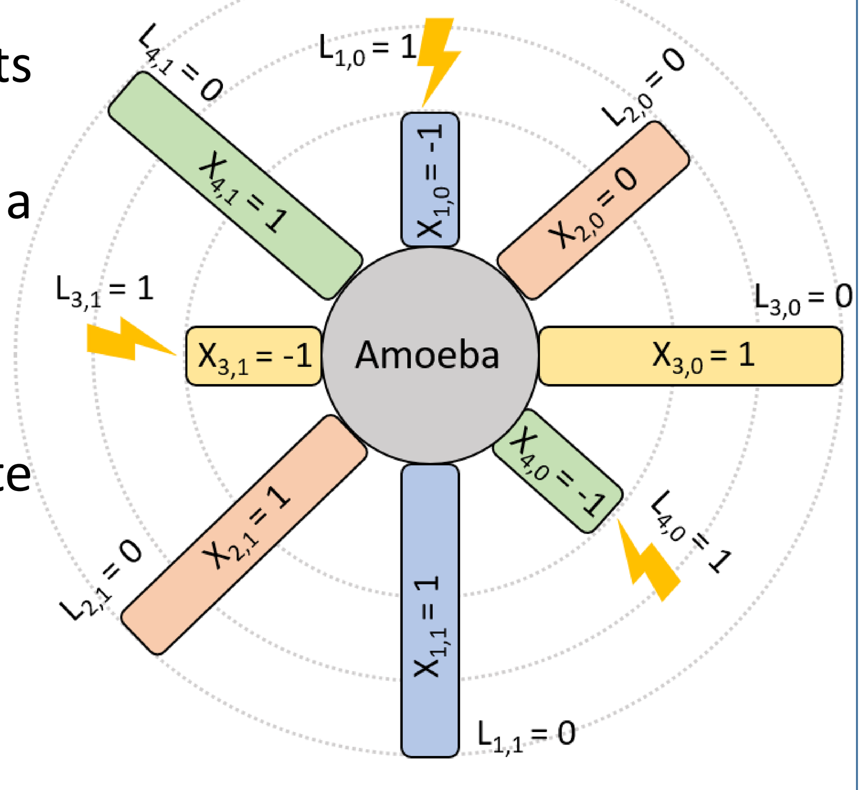
❑ AmoebaSAT
- Inspired by the behaviour of an amoeba
- Efficiently explore the assignment of decision variables in parallel
- Take less iterations# to find a solution than state-of-the-art algorithms

❑ This work:
- Implement AmoebaSAT on FPGA through High Level Synthesis
- Fine-grained (variable-level) parallelism
- Explore several optimization techniques



AmoebaSAT Computing Architecture

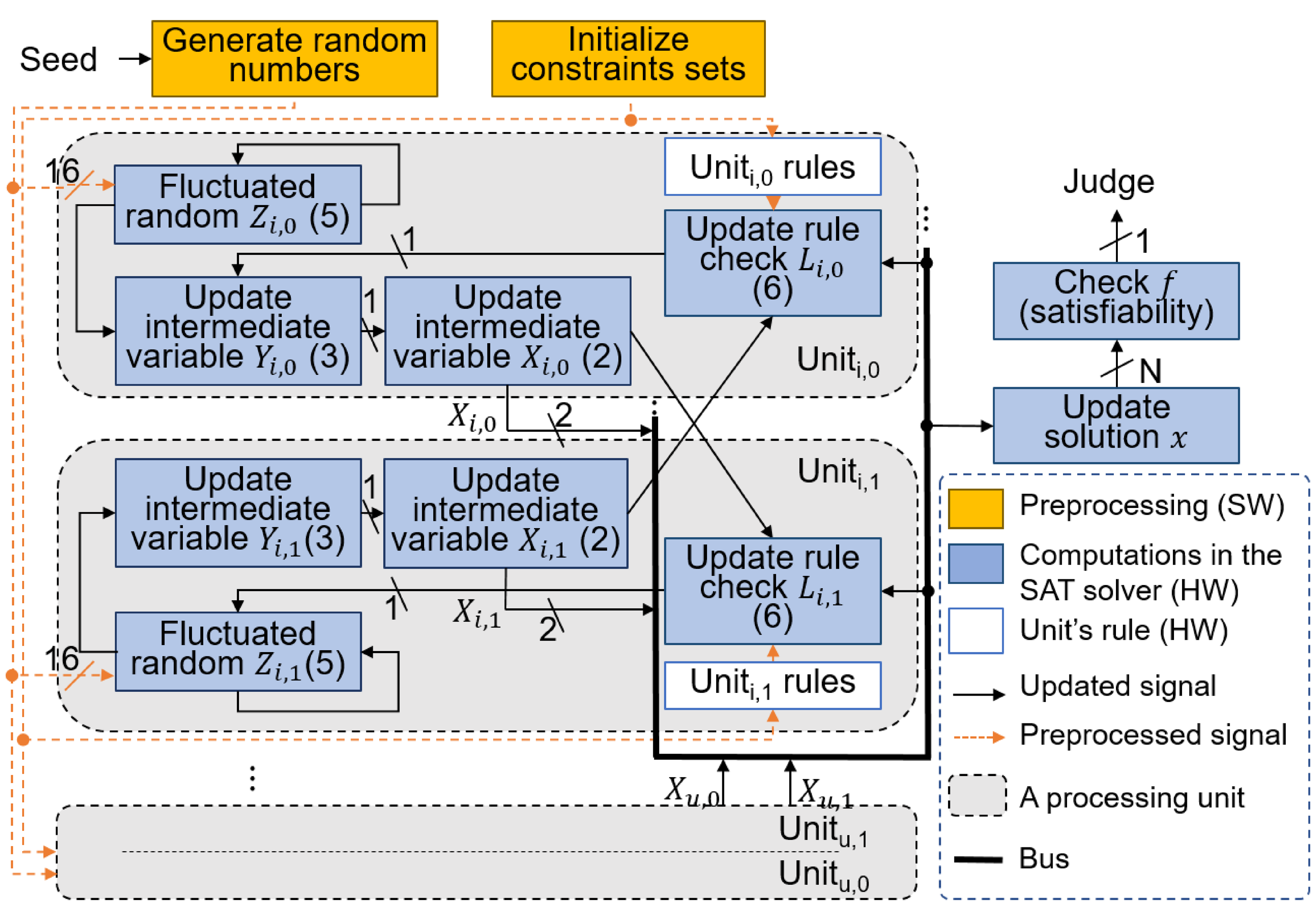
❑ Model
- For N variables, construct 2N units representing expandable amoeba legs
- Each variable x_i is represented by a ternary pair:
  - Unit (i,0): assert $x_i = 0$
  - Unit (i,1): assert $x_i = 1$
- Each unit (i,v) has its own intermediate variables:
  - $X_{i,v} \in \{-1,0,1\}$: equilibrium volume of unit (i,v)
  - $L_{i,v} \in \{0,1\}$: light on unit (i,v)
  - Etc.



Assignment of 4 variables (N=4): x_1=1, x_2=1, x_3=0, x_4=1

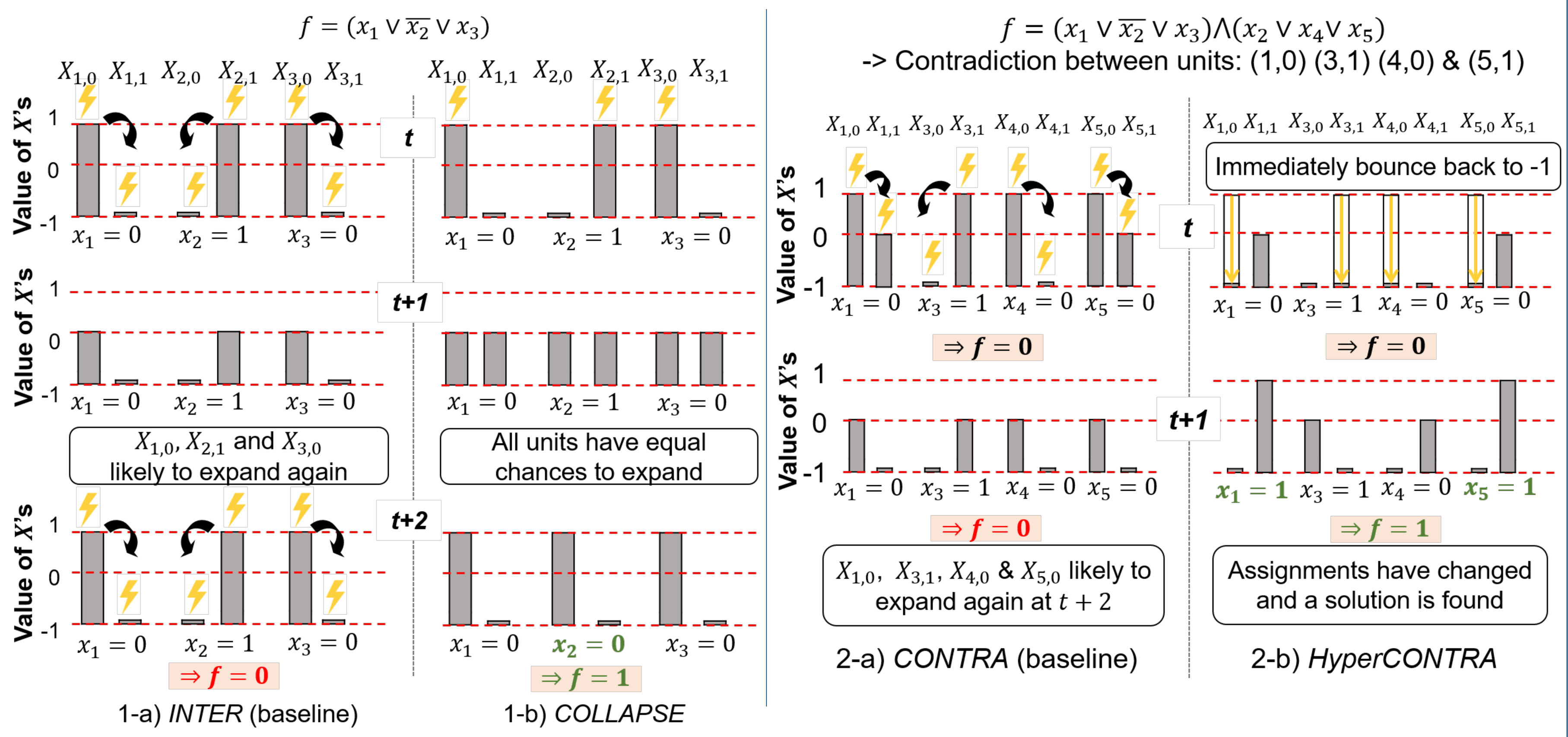
❑ Bounceback control
- Constraints are generated from input SAT instances
- $L_{i,v}$ is turned on when a constraint is violated
  - $L_{i,v}(t) = 1$ => Bounceback signal ON
  - $L_{i,v}(t) = 0$ => Bounceback signal OFF
- Ensure the assignment of all variables satisfies problem constraints

❑ Hardware Architecture

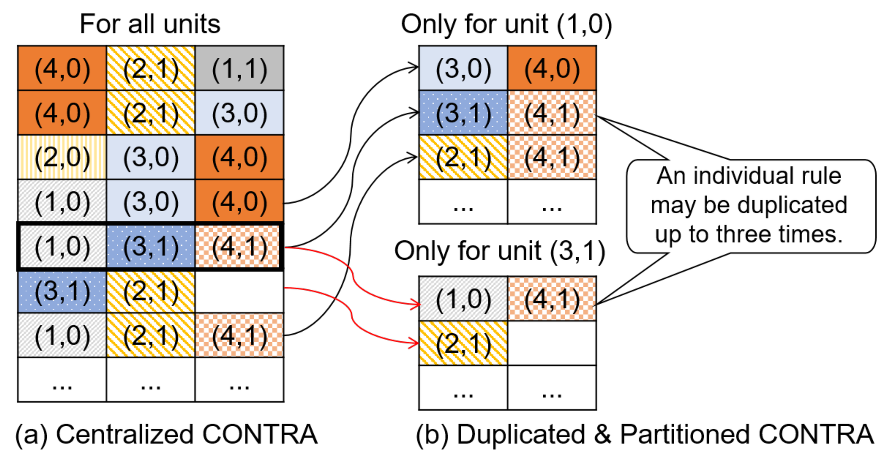


High-level design of Hardware AmoebaSAT Solver

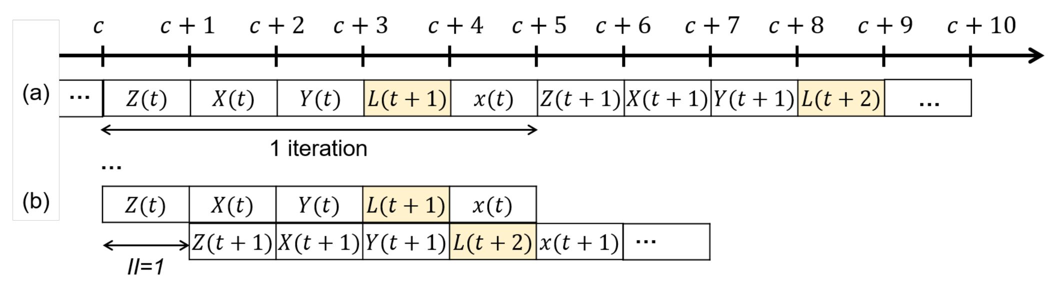
❑ Bounceback rules extension



❑ Data localization
- ➢ Avoid data dependencies
  => Memory partitioning and parallelism



- ➢ Loop pipelining

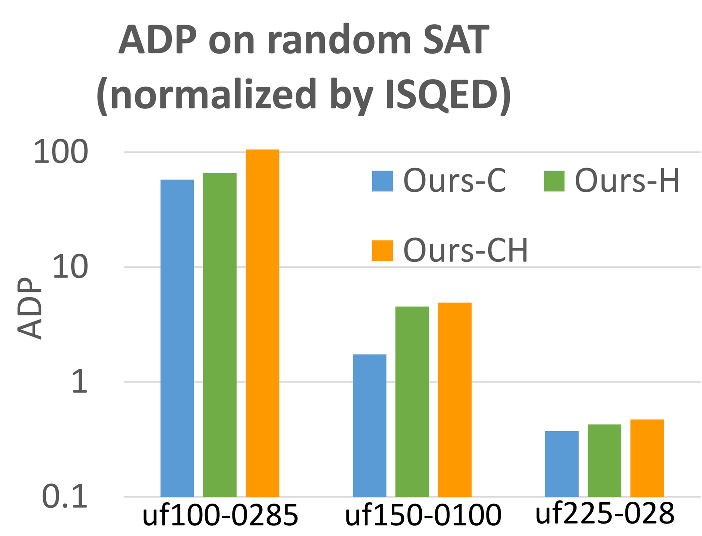


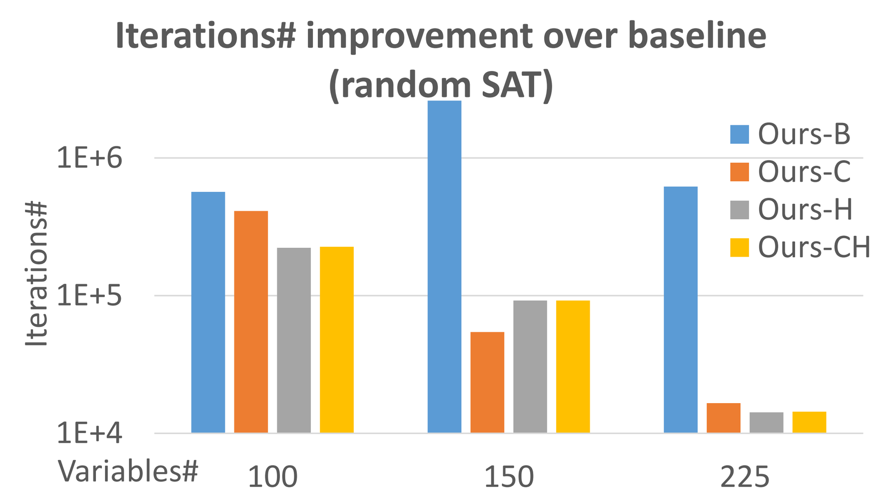
Experiments & Results

❑ Experiment A: Synthesis and simulation results of AmoebaSAT with random & flat graph colouring benchmarks from SATLIB

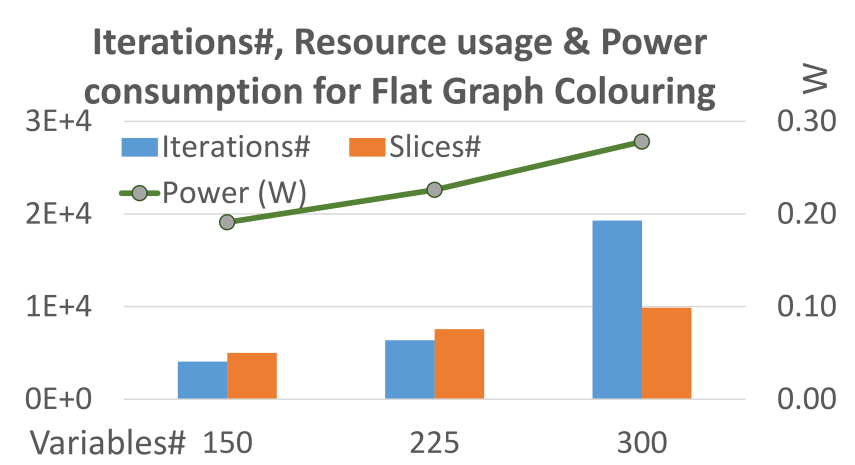
○ Setup: Zynq xc7z030ffv676-3, VivadoHLS 2016.3
- Ours-B: Baseline, implemented original AmoebaSAT
- Ours-C: COLLAPSE version
- Ours-H: HyperCONTRA version
- Ours-CH: employs both COLLAPSE & HyperCONTRA

○ Results



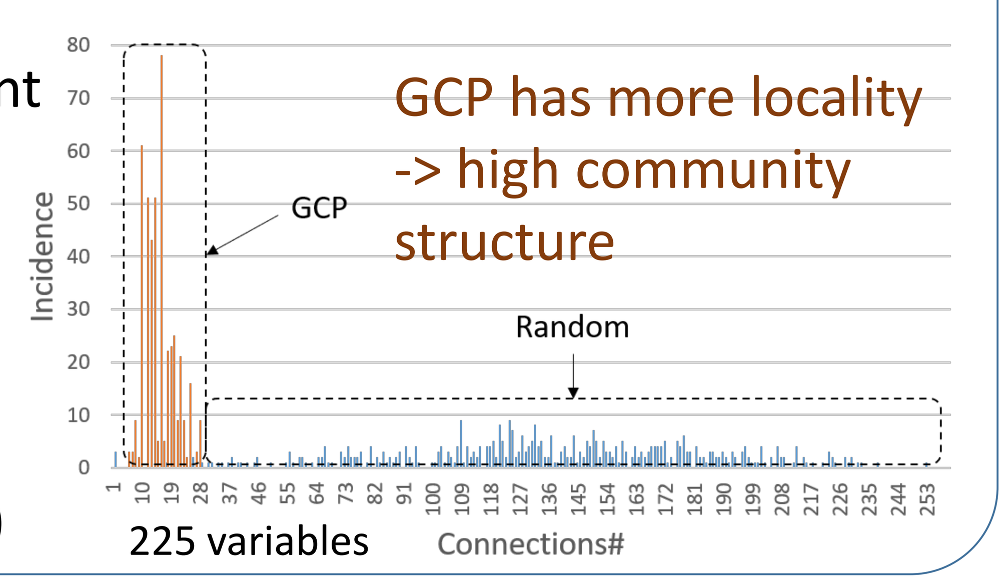


- ➢ Better ADP than state-of-the-art on bigger instances (ISQED: 2.12-2.67x, WalkSAT: 3.24x)
- ➢ Extensions significantly reduce iterations#
- ➢ Scalable with the instance size



❑ Experiment B: Important insights on IoT-based SAT instances
- Unit's connections#: dependent units# for rules checking
- Measured a histogram of connections# and their incidence

=> Leverage these insights into HW configurations (future work)



Publications

1. A. H. N. Nguyen, et al., "FPGA-Based Hardware/Software Co-Design ofa Bio-Inspired SAT Solver," in IEEE Access, vol. 8, pp. 49053-49065,Mar. 2020.
2. A. H. N. Nguyen, et al., "Amoeba-Inspired Hardware SAT Solverwith Effective Feedback Control," Proc. of Int'l Conference on Field-Programmable Technology, pp.241-246, 2019.
3. A. H. N. Nguyen et al., "FPGA-Based Amoeba-Inspired SAT Solver forCyber-Physical Systems," Proc. of WIP of Int'l Conference on Cyber-Physical Systems, pp.316-317, 2019.
4. Y. Nakayama et al., "Real-Time Routing for Wireless Relay Fronthaulwith Vehicle-Mounted Radio Units," Proc. of IEEE Vehicular TechnologyConference-Spring, 2020.