Towards a Complexity-theoretic Understanding of Restarts in SAT solvers

Chunxiao (Ian) Li[1], Noah Fleming[2], Marc Vinyals[3], Toniann Pitassi[2] and Vijay Ganesh[1]

[1] University of Waterloo, Canada

[2] University of Toronto, Canada

[3] Technion, Israel

PART 1

Context and Motivation

Why should we care about restarts?

- Empirical reasons:
  - Solvers with restarts outperform solvers without restarts
  - Natural for search procedures

- Theoretical reasons:
  - CDCL with non-deterministic branching and restarts (after every conflict) is p-equivalent to general resolution [Pipatsrisawat and Darwiche *2011, Atserias et al. 2011*]
  - The question of whether CDCL without restarts is p-equivalent to resolution has been open for two decades
  - The problem is interesting!

What is restart?

- History of restarts
  - Restarts have been studied extensively in the context of search and optimization problems. [Shylo 2016]
    - Escape local minima
- Restarts in DPLL:
  - Upon invocation, erase the trail (partial assignment)
- Restarts in CDCL solvers:
  - Upon invocation, erase the trail while keeping some other information
    - Learnt clauses
    - Activities in VSIDS branching
    - Phase-saving values.
- Are restarts “really” useful for SAT solvers? How do we justify it empirically? And how do we prove it theoretically?

Previous work on the power of restarts

- Empirical perspective:
    - Heavy-tailed explanation – DPLL
        - “Heavy-Tailed Phenomena in Satisability and Constraint Satisfaction Problems” [Gomes and Selman 2000]
    - Restarts compact assignment trail
        - “ManySAT: a Parallel SAT solver” [HJS 2008]
        - “Machine Learning-based Restart Policy for CDCL SAT Solvers” [LOMTLG 2018]

- Theoretical perspective:
    - Pool resolution [Van Gelder 2005] and regWRTI [BHJ 2008]
    - Common consensus: CDCL solvers without restarts are weaker than general resolution

Various configurations of CDCL solvers

Backjumping

Variable selection

And a few more…

CDCL SAT solver

Conflict analysis

Restarts

Clause deletion

Value selection

PART 2

Results

Our results on restarts

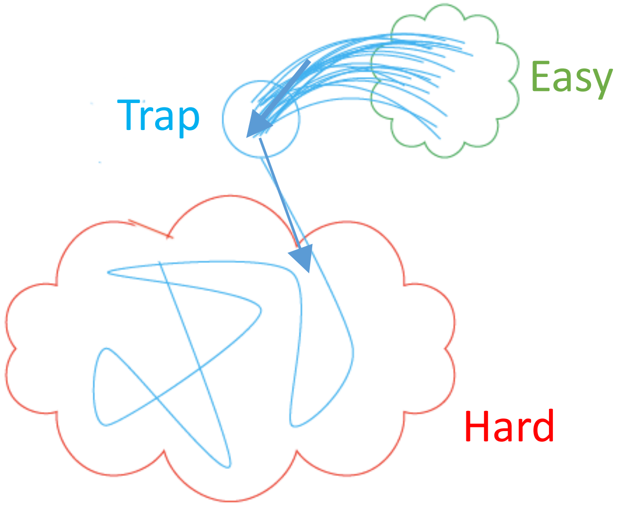
- A total of 4 separation and 2 equivalence theorems [LFVPG SAT 2020]
- Separation theorem: drunk CDCL with and without restarts
  - Drunk CDCL = backtracking + non-deterministic variable selection + random value selection + clause learning
  - For satisfiable formulas (Ladder), drunk CDCL without restarts takes exponential time, while drunk CDCL + restarts takes polynomial time w.h.p.
- Separation theorem: VSIDS with and without restarts
  - backjumping + VSIDS variable selection + phase-saving value selection
  - For unsatisfiable formulas (Pitfall), CDCL + VSIDS without restarts takes exponential time, while CDCL + VSIDS + restarts takes polynomial time w.h.p.

Our approach to study the power of restarts

	Previous theoretical approach	Our approach
Type of formulas	Unsatisfiable	Unsatisfiable + satisfiable
Type of heuristics	Non-deterministic	Weakened variable selection Weakened value selection Backtracking/backjumping

- Why weakened heuristics?
  - The power of restarts is subtle:
    - Subtle interplay between solver heuristics and the power of restarts
    - The power of restarts becomes more apparent when certain heuristics are weaker than non-deterministic

Proof methodology – Ladder and Pitfall formulas



- The pitfall formulas have three components:
  - Hard formula for resolution
  - Trap – Tricks the solver into focusing on the hard formula
  - Easy formula – a small backdoor
    - (weak backdoor in the satisfiable case, and strong backdoor for unsatisfiable formulas)
- Lower bound argument:
  - Without restarts, w.h.p. the solver will fall into the trap, and needs to refute the hard formula before escaping
- Upper bound argument:
  - Solvers with restarts can exploit the small backdoor
    - Finding the backdoor variables for the strong backdoor
    - Finding the desired assignment to the backdoor variables for the weak backdoor

Separation result: drunk CDCL

- Model:
  - Backtracking: undo the most recent decision on the trail after learning a conflict
  - Non-deterministic variable selection: non-deterministically returns an unassigned variable upon invocation.
  - Random value selection: returns a truth value uniformly at random
- New formula: Ladder_n
  - Satisfiable formula
  - log(n) size weak backdoor
  - All but one assignment to the weak backdoor variables implies getting trapped
    - No restarts: Hard to assign the backdoor variables correctly with random value selection, branching on other variables also implies the trap w.h.p.
    - Restarts: Keep querying the backdoor variables until assigning them correctly

Separation result: VSIDS

- Model
  - Backjumping: after learning a learnt clause, undo assignments with decision level higher than the literal with the second highest decision level in the learnt clause.
  - VSIDS variable selection: returns the variable with highest activity, with random tie breaking. We consider a version of restarts that resets activities
  - Phase-saving value selection: returns “true” if the input variable x was assigned “true” when the last time x was on the trail, else return “false”. If a variable has not been assigned, then return “false”.
- Formula [Vinyals 2020]:
  - Unsatisfiable formula
  - Constant size strong backdoor
    - No restarts: w.h.p. first conflict bumps activities of variables in the hard formula [Vinyals 2020]
    - Restarts: restart to reset the activities, and use random tie breaking to exploit the constant size backdoor

Other results

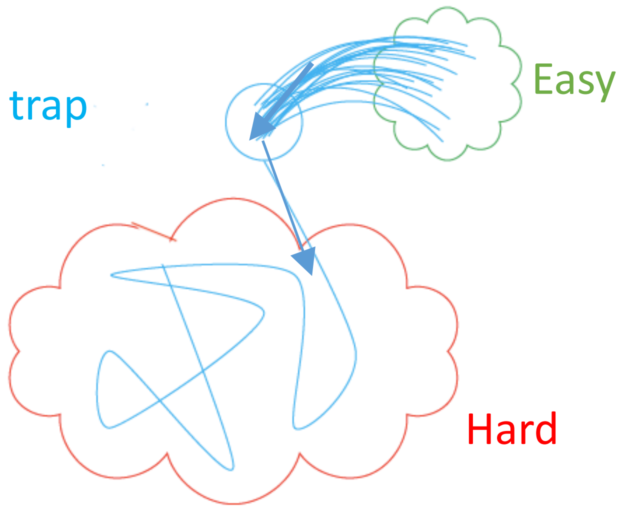
- Equivalence result: static CDCL
  - For satisfiable and unsatisfiable formulas
  - backjumping + static variable selection + static value selection
- Equivalence result: non-deterministic DPLL
  - For unsatisfiable formulas
  - backtracking + non-deterministic variable selection + non-deterministic value selection
- Separation result: drunk DPLL
  - For satisfiable formulas
  - backtracking + non-deterministic variable selection + random value selection
- Separation result: weak decision learning scheme CDCL
  - For unsatisfiable formulas
  - backjumping + non-deterministic variable selection + non-deterministic value selection

PART 3

Key Insights and Takeaways

Key insights and conclusions

- Heuristics that are weaker than non-deterministic ones
- The power of restarts is subtle:
    - Subtle interplay between solver heuristics and the power of restarts
    - The power of restarts becomes more apparent when certain heuristics are weaker than non-deterministic
- Satisfiable vs unsatisfiable formulas
- Pitfall formulas



Open questions

- Whether CDCL solver without restarts is p-equivalent to general resolution remains open
- Backtracking vs backjumping

Thank you!