Artificial Intelligence

Logic 3: Local Search

Jendrik Seipp

Linköping University

based on slides by Thomas Keller and Malte Helmert (University of Basel)

Local Search: GSAT

Local Search for SAT

- Apart from systematic search, there are also successful local search methods for SAT.
- These are usually not complete and in particular cannot prove unsatisfiability for a formula.
- They are often still interesting because they can find models for hard problems.
- However, all in all, DPLL-based methods have been more successful in recent years.

Local Search for SAT: Ideas

local search methods directly applicable to SAT:

- candidates: (complete) assignments
- solutions: satisfying assignments
- search neighborhood: change assignment of one variable
- heuristic: depends on algorithm; e.g., #unsatisfied clauses

GSAT (Greedy SAT): Pseudo-Code

auxiliary functions:

- violated(Δ, I): number of clauses in Δ not satisfied by I
- flip(I, v): assignment that results from I
  when changing the valuation of proposition v

function GSAT(Δ):

repeat *max-tries* **times**:
  $I :=$ a random assignment
  repeat *max-flips* **times**:
    if $I \models \Delta$:
      return I
    $V_{\text{greedy}} :=$ the set of variables v occurring in Δ
      for which violated$(\Delta, \text{flip}(I, v))$ is minimal
    randomly select $v \in V_{\text{greedy}}$
    $I := \text{flip}(I, v)$
return no solution found

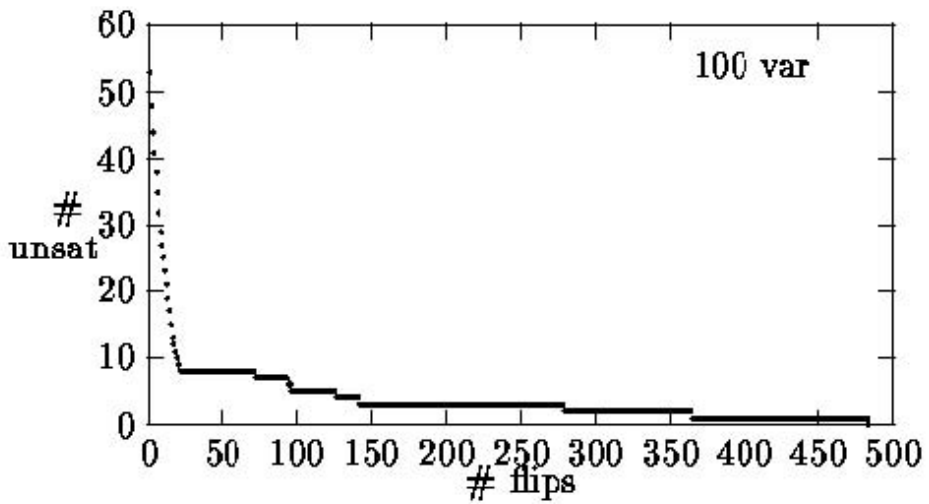
Whiteboard: $(A \vee \neg B) \wedge (A \vee C) \wedge (\neg A)$ with $I_0 = \{A \mapsto \mathbf{T}, B \mapsto \mathbf{T}, C \mapsto \mathbf{F}\}$

GSAT: Discussion

GSAT has the usual ingredients of local search methods:

- hill climbing
- randomness (although relatively little!)
- restarts

empirically, much time is spent on plateaus:



Local Search: Walksat

Walksat: Pseudo-Code

$\text{lost}(\Delta, I, v)$: #clauses in Δ satisfied by I, but not by $\text{flip}(I, v)$

function Walksat(Δ):

repeat *max-tries* **times**:
    I := a random assignment
    repeat *max-flips* **times**:
        if $I \models \Delta$:
            return I
        C := randomly chosen unsatisfied clause in Δ
        if there is a variable v in C with $\text{lost}(\Delta, I, v) = 0$:
            V_{choices} := all such variables in C
        else with probability p_{noise}:
            V_{choices} := all variables occurring in C
        else:
            V_{choices} := variables v in C that minimize $\text{lost}(\Delta, I, v)$
        randomly select $v \in V_{\text{choices}}$
        $I := \text{flip}(I, v)$
return no solution found

Walksat vs. GSAT

Comparison GSAT vs. Walksat:

- much more randomness in Walksat
  because of random choice of considered clause
- “counter-intuitive” steps that temporarily increase
  the number of unsatisfied clauses are possible in Walksat

$\rightsquigarrow$ smaller risk of getting stuck in local minima