INTERACTIVE VISUALIZATION OF SATISFIABILITY SOLVING (satviz.unl.edu)

Mary D. Burke, Daniel Geschwender, Keegan Lunn, Margaret Krause, Berthe Y. Choueiry & Matthew Dwyer
Computer Science & Engineering • University of Nebraska-Lincoln

1. Propositional Satisfiability (SAT)

SAT Problem

Given: A propositional Satisfiability (SAT) sentence, e.g.,

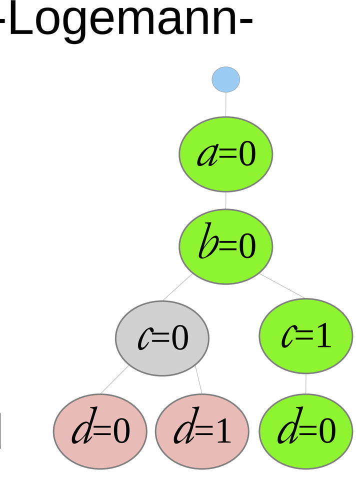
$$(\bar{a} \vee \bar{c} \vee \bar{d}) \wedge (b \vee c \vee \bar{d}) \wedge (a \vee \bar{b} \vee d) \wedge (a \vee c \vee d) \wedge (\bar{a} \vee b \vee \bar{c})$$

clause — *literals* — *terms or Boolean variables*

Question: Find an assignment for the Boolean variables such that the sentence holds, e.g., $a=0, b=0, c=1, d=0$

Solving SAT

- SAT is NP-complete, solved with search
- MiniSat[1] is a SAT solver based on the Davis-Putnam-Logemann-Loveland (DPLL) backtracking algorithm.
- DPLL explores combinations of values for the Boolean variables in a depth-first manner by expanding partial assignments that are consistent with the clauses of the sentence.
- When a partial solution cannot be expanded without violating one of the clauses, a conflict is detected and backtracking is occurs.



MiniSat uses heuristics & inference techniques to enhance the performance of DPLL, including

- Boolean Constraint Propagation (BCP).
- Conflict clause learning.
- Conflict-directed backtracking.

2. Our Project

Motivation & Goal

- SAT and Constraint Processing (CP) are fundamental areas of Computer Science that address the same computational questions.
- Compare SAT & CP: formalisms, search, and inference mechanisms.

Approach

Constraint Processing

- Studied formalism, modeling, algorithms for search, backtracking, and constraint propagation.
- Built, from scratch, a CP solver with main fundamental mechanisms & conducted extensive empirical performance studies.

SAT Solving

- Studied Tseiten's encoding, propagation, conflict graph, clause learning, simplification at pre-processing, etc.
- Instrumented MiniSat to capture and animate its main operations.
- Built a visualization tool of MiniSat using Flare and FlashBuilder.

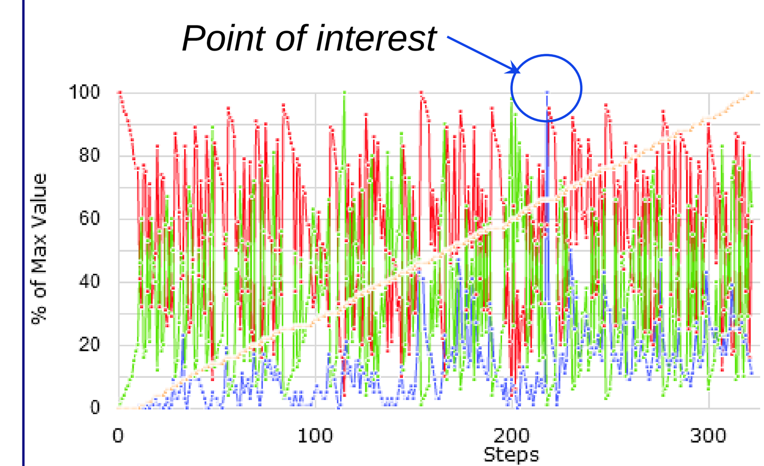
Outcomes

- A comparative synthesis of terminology, mechanisms in CP & SAT.
- A visualization tool of MiniSat as an instructional aid to teach Computer Science students about SAT & its fundamentals.

4. Visualization: The Chart

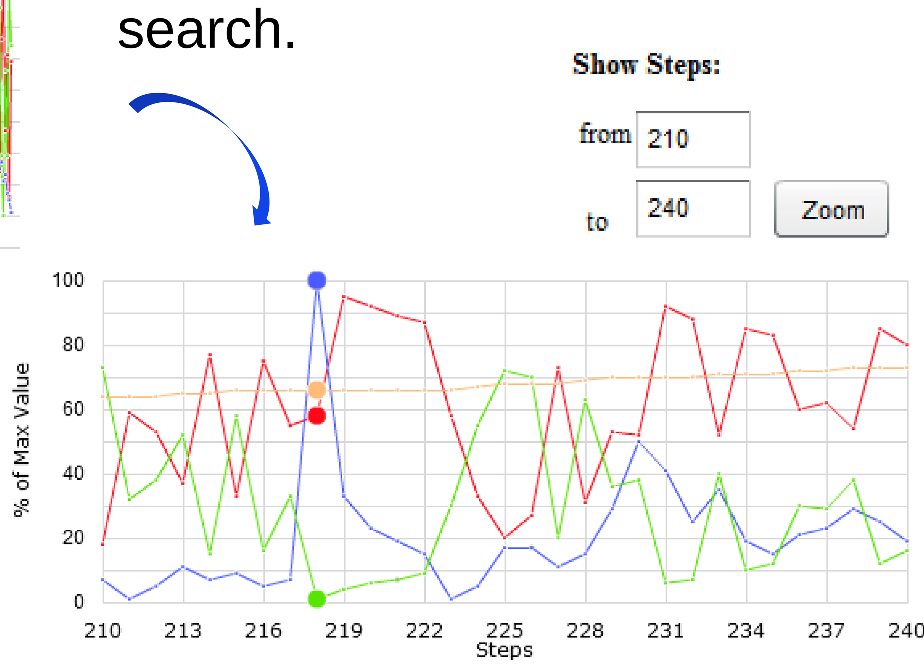
Chart selectively displays search statistics using check boxes

1. **Open Original Clauses** is the number of clauses given as input but not yet satisfied.
2. **Open Learnt Clauses** is the number clauses learned during search but not yet satisfied.
3. **Instantiated Literals** is the number of literals instantiated by decision or by propagation.
4. **Learnt Clauses** is the total number of clauses learnt during search.

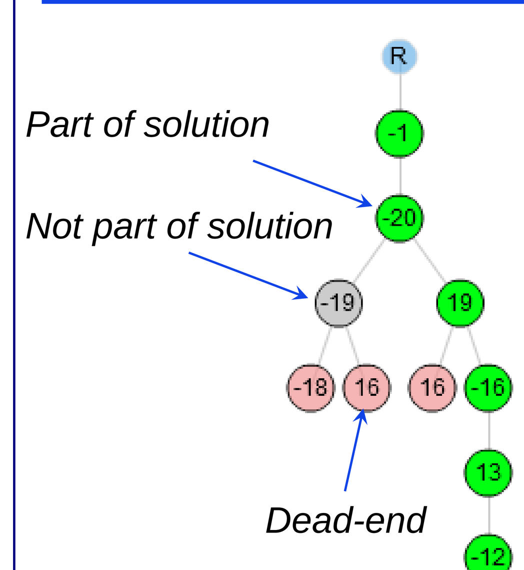


The chart shows trends during search.

Zooming allows users to explore points of interest.



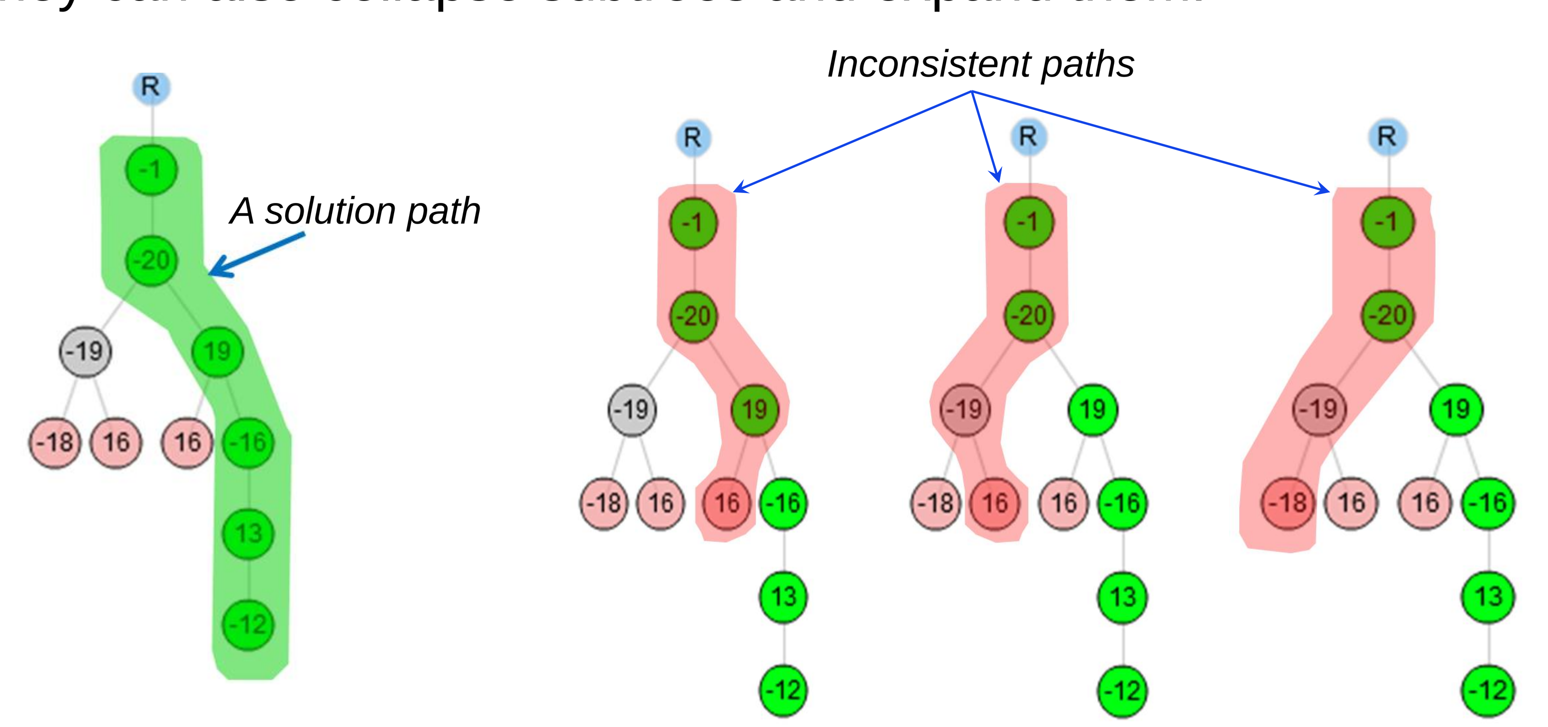
5. Visualization: The Search Tree



The search tree traces the assignments of decision variables. Each tree node

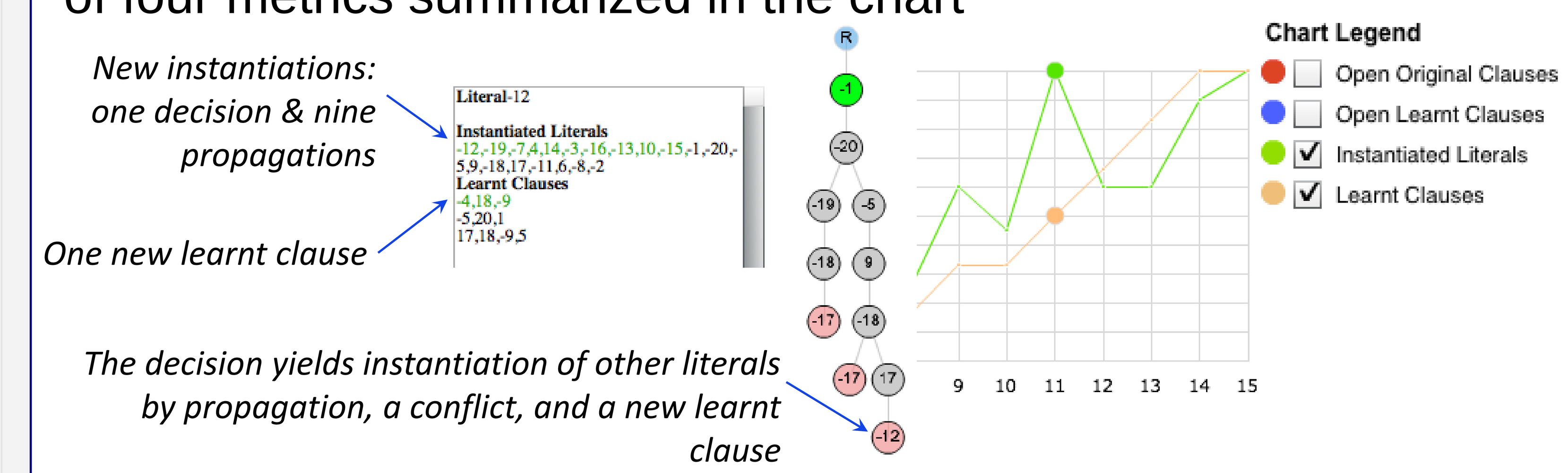
- Represents a MiniSat decision (assumption).
- Is labeled by the decision:
  "-20" indicates that the literal l_{20} is set to false $l_{20} \leftarrow 0$
  "13" indicates that the literal l_{13} is set to true $l_{13} \leftarrow 1$
- Is colored based on its status: a dead-end (pink), part of the solution (green), or not part of the solution (grey).

Users visually identify solution paths & inconsistent paths. They can also collapse subtrees and expand them.



6. Visualization: The Explanation Box

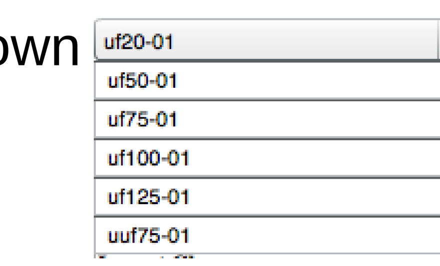
The user can examine in the Explanation Box the details of any of four metrics summarized in the chart



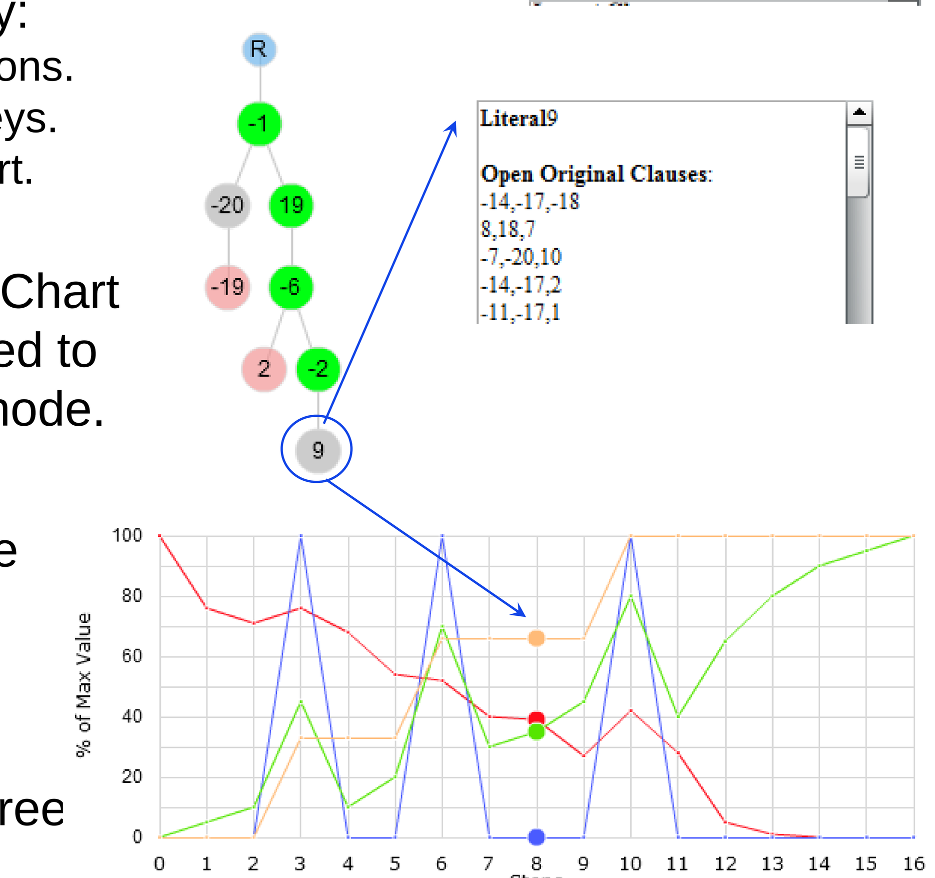
- The latest information is displayed first & in green to facilitate understanding
- Clauses are shown in CNF: $-4, 18, -9 \equiv \bar{l_4} \vee l_{18} \vee \bar{l_9}$
- When the user hovers over a node in the tree, the details of the corresponding metrics are listed in the Explanation Box.
- The contents of the Explanation Box change as the tree is being built to reflect the metrics details of the latest node generated in the tree.

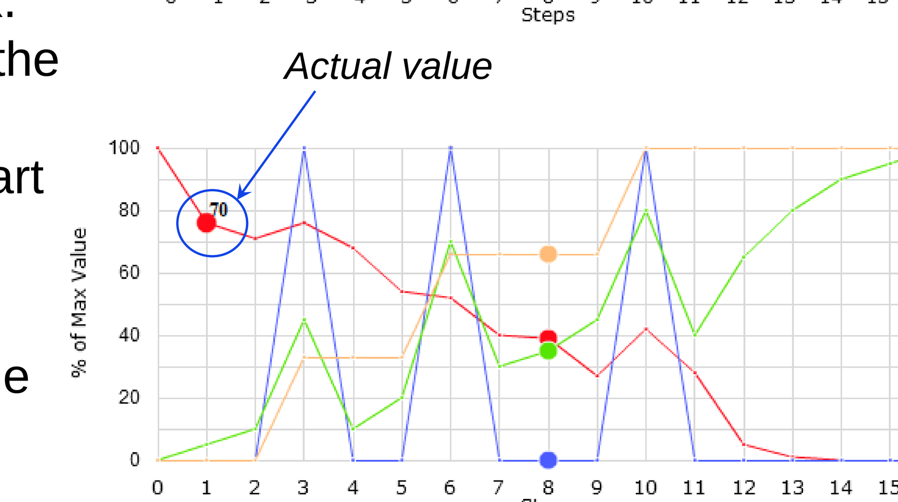
7. Driving the Visualization

A problem instance is selected from a drop-down menu. The prefixes 'uf'/'uuf' indicate that the instance is satisfiable/unsatisfiable.



- The user can build the tree by:
  - Activating the Play/Pause buttons.
  - Pressing the right/left arrow keys.
  - Clicking on a point on the Chart.
- As the tree is being built, the Chart & Explanation Box are updated to reflect the state of the latest node.
- The user can zoom in the tree using the mouse scroll-wheel or the CTRL+Click shortcut.
- Hovering over a node in the tree updates the Explanation Box. The corresponding node on the Chart blinks to indicate a relationship between the Chart and the tree.
- If the user hovers over a node the actual value is shown.





- When more than one metrics are selected, the chart's y-axis is shown as a percentage of the max values of each metric.
  When only one metric is selected the *y*-axis shows the true scale.

This research was supported by CREU-W of the CRA, Undergraduate Creative Activities and Research Experiences Program of the University of Nebraska-Lincoln, and National Science Foundation REU supplements for Grant #CNS-0720654 and Grant #RI-111795.