# Instance-Based Selection of Policies for SAT Solvers

Mladen Nikolić, Filip Marić and Predrag Janičić
Automated Reasoning Group
Faculty of Mathematics, University of Belgrade, Serbia
SAT 2009, Swansea UK

July 1, 2009.

# Overview

1. Problem Description
2. Methodology
3. Experiments and Evaluation
   - Training Phase
   - Analysis
   - Exploitation
4. Related Work
5. Conclusions and Further Work

# Overview

1. Problem Description
2. Methodology
3. Experiments and Evaluation
   - Training Phase
   - Analysis
   - Exploitation
4. Related Work
5. Conclusions and Further Work

# SAT Solvers and Their Policies

- DPLL-based, conflict driven SAT solvers are considered
- Heuristic policies (variable selection, restarts, forgetting. . . ) control SAT solver behaviour
- Strategies are fixed combinations of policies
- Formulae from the applications are typically clustered in the families

# SAT Solvers and Their Policies

- DPLL-based, conflict driven SAT solvers are considered
- Heuristic policies (variable selection, restarts, forgetting. . . ) control SAT solver behaviour
- Strategies are fixed combinations of policies
- Formulae from the applications are typically clustered in the families
- Main idea: find a family of formulae most similar to the formula being solved, and use the strategy that has performed good for formulae of that family?

# Hypotheses

(H1) Formulae of the same family share some syntactical properties that can be used for automated formula classification

(H2) For each family of formulae there is only a small number of solving strategies that are appropriate — that show better performance on formulae belonging to that family then all other available strategies

(H3) For syntactically similar formulae, the best strategies are also similar in some sense

## Goals

- Analyze a relationship between the syntax of a formulae and their appropriate solving strategies
- Develop a methodology for intelligent choosing of solvers' strategies, based on the syntax of the input formula, and use it to improve the efficiency of a SAT solver

# Overview

1. Problem Description
2. Methodology
3. Experiments and Evaluation
   - Training Phase
   - Analysis
   - Exploitation
4. Related Work
5. Conclusions and Further Work

# Overview of the Methodology

1. Training
   - Solving formulae from a large corpus for all given strategies
   - Determining which strategies are best for different families of formulae
2. Analysis
   - Verification of the hypotheses (H1-H3)
3. Exploitation
   - Classification of an unknown input formula to some of the families
   - Solving the input formula using the strategy best for the chosen family

# Corpora of Formulae

- Corpus SAT 2002
  - 1964 formulae
  - 39 families
- Corpus SAT 2007
  - 906 formulae

# SAT Solver and Strategies Used

SAT solver: ArgoSAT

**Variable selection**

$\mathrm{VS}_{\mathit{random}}$, $\mathrm{VS}_{\mathit{VSIDS}}^{1.0,\,1.0/0.95,\,\mathit{freq}}$, $\mathrm{VS}_{\mathit{random}}^{0.05} \circ \mathrm{VS}_{\mathit{VSIDS}}^{1.0,\,1.0/0.95,\,\mathit{freq}}$

**Polarity selection**

$\mathrm{PS}_{\mathit{pos}}$, $\mathrm{PS}_{\mathit{neg}}$, $\mathrm{PS}_{\mathit{random}}^{0.5}$, $\mathrm{PS}_{\mathit{polarity\_caching}}^{\mathit{neg}}$, $\mathrm{PS}_{\mathit{polarity\_caching}}^{\mathit{freq}}$

**Restart**

$\mathrm{RS}_{\mathit{no\_restart}}$, $\mathrm{RS}_{\mathit{minisat}}^{100,1.5}$, $\mathrm{RS}_{\mathit{luby}}^{512}$, $\mathrm{RS}_{\mathit{picosat}}^{100,1.5}$

# Classification of Propositional Formulae

- Represent formulae by vectors of numerical features
- Perform classification over such vectors using standard classification techniques

# Choice of Features

Formulae are represented by 33 *syntactical* features:

- The number of clauses $c$ and variables $v$ in the input formula, and their ratio $\frac{c}{v}$,
- Node degree statistics for variable nodes in variable-clause graph: mean, variation coefficient, minimum, maximum and entropy,
- Fraction of binary clauses, ternary clauses, and Horn clauses,
- ...

# Classification Method

Formulae are classified by *k* Nearest Neighbours method with the following distance function

$$d(f_1, f_2) = \sum_{x \in features} \frac{|f_1(x) - f_2(x)|}{\sqrt{|f_1(x) f_2(x)|} + 1}$$

# Overview

1. Problem Description
2. Methodology
3. Experiments and Evaluation
   - Training Phase
   - Analysis
   - Exploitation
4. Related Work
5. Conclusions and Further Work

# Training Details

- Cutoff time for solving used is 10 minutes
- Clause and variable shuffling
- Total number of SAT solver calls:
  1964 formulae × 2 shuffled variants × 60 strategies $=$ 235.680
- A cluster computer with 32 processors was used
- Total computing time on the cluster was around a month.

# Evaluation of Formulae Classification (H1)

*(H1) Formulae can be automatically classified into families based on their syntax*

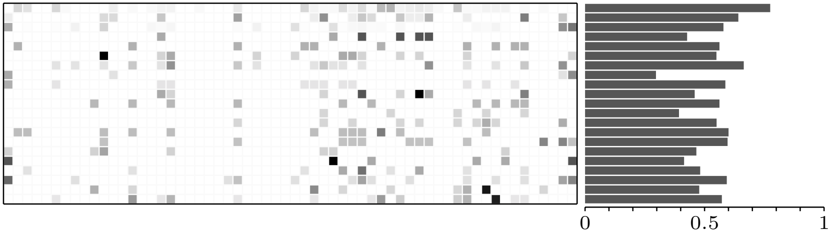
- Corpus SAT 2002
- Average feature computation and classification time for a formula was 0.4s
- The best precision was achieved by 1 Nearest Neighbour
- The leave one out procedure was used
- Precision was 92.5%

# Analysis of Appropriate Strategies (H2)

*(H2) For each family of formulae there is only a small number of appropriate solving strategies*

- For each family and each strategy the percentage of formulae for which that strategy is better than the others was considered
- The normalized entropies for families were calculated

# Analysis of Appropriate Strategies (H2)



# Similarity of Formulae and Their Best Strategies (H3)

*(H3) For syntactically similar formulae, the best strategies are also similar*

- For each 2 formulae form the corpus do:
  - Calculate distance between them
  - Calculate edit distance between their best strategies
- The Pearson correlation coefficient between these distances is 0.51

# Strategy Selection Procedures

Best fixed - for each formula use the same strategy that is the best on the training corpus

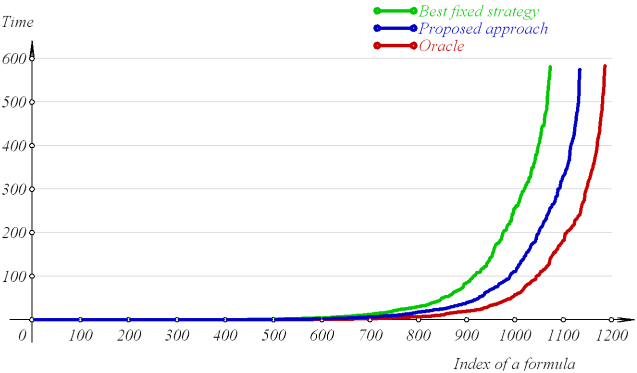
Proposed - classify the formula and use the strategy best for the chosen family

Oracle - for each formula use the strategy that is the best for it

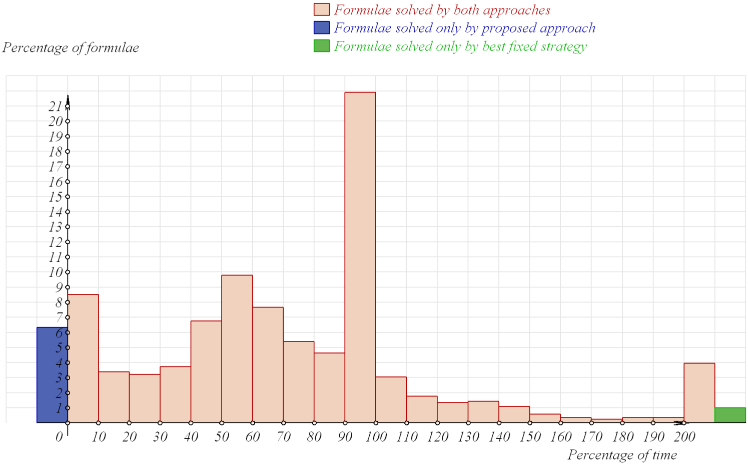
# Evaluation of Strategy Selection Procedures on SAT2002

| Procedure | No. solved | Median time |
|---|---|---|
| Best fixed | 1073 | 207.4s |
| Proposed | 1135 | 92.6s |
| Oracle | 1187 | 45.8 |

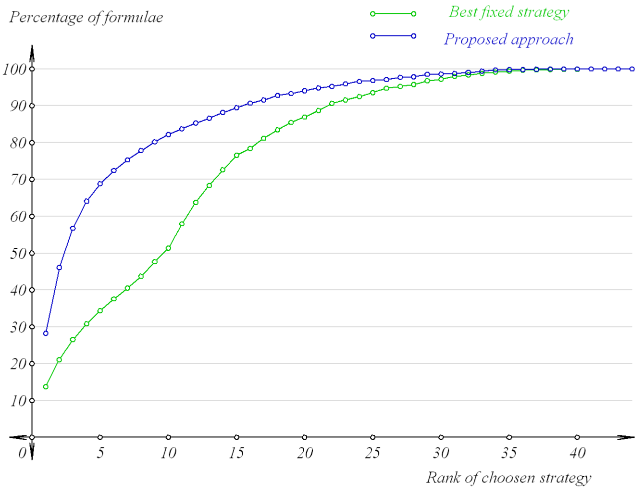
# Solving Times for Different Strategy Selection Procedures



# Speedup Histogram for the Proposed Procedure



# CDF of Ranks of Chosen Strategies



# Evaluation on SAT2007.

ArgoSmArt = ArgoSAT + proposed strategy selection procedure

| System | No. solved | 20-th perc. time |
|---|---|---|
| ArgoSAT | 219 | 311.6s |
| ArgoSmArt | 239 | 249.5s |

# Overview

1. Problem Description
2. Methodology
3. Experiments and Evaluation
   - Training Phase
   - Analysis
   - Exploitation
4. Related Work
5. Conclusions and Further Work

# Related Work

- L. Xu, F. Hutter, H. Hoos, K. Leyton-Brown. SATzilla: Portfolio-Based Algorithm Selection for SAT. *Journal of Artificial Intelligence Research*, 2008.
- S. Haim, T. Walsh. Online Estimation of SAT Solving Runtime. *SAT 2008*.
- H. Samulowitz, R. Memisevic. Learning to Solve QBF. *AAAI 2007*.
- ...

# Overview

1. Problem Description
2. Methodology
3. Experiments and Evaluation
   - Training Phase
   - Analysis
   - Exploitation
4. Related Work
5. Conclusions and Further Work

# Conclusions and Further Work

- Intelligent choosing of solvers' strategies, based on the syntax of the input formula, can significantly improve efficiency of a SAT solver
- Further work:
  - Further analysis of gathered data
  - Stochastic parameter optimization
  - On-line control of SAT solver