Instance-based selection of strategies for SAT solvers

Mladen Nikolić, Filip Marić and Predrag Janičić

January 31, 2009

Introduction

SAT Solvers and Their Strategies

- DPLL-based SAT solvers, their applications and importance
- Strategies of SAT solvers
  - Variable selection strategies
  - Polarity selection strategies
  - Restart strategies
  - Forget strategies
- Choosing SAT solver strategies
- Families of propositional formulae

Sketch of the Methodology

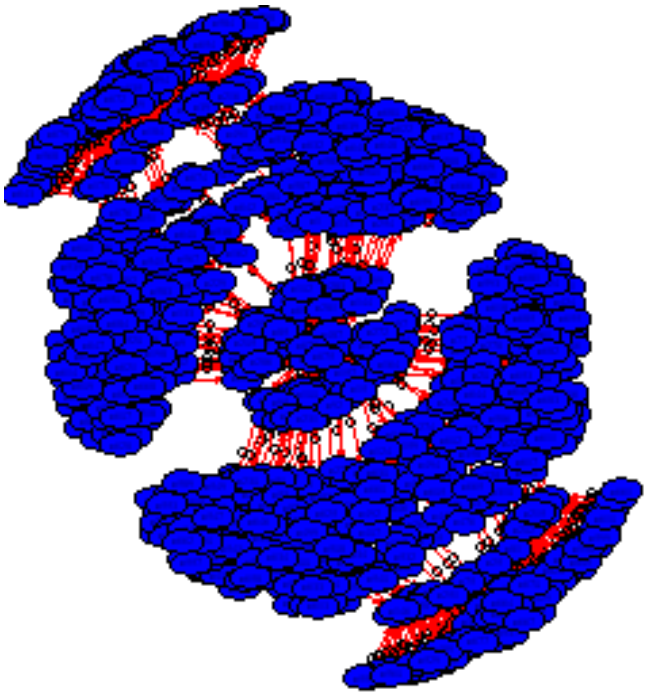
A methodology of choosing suitable strategies is formulated for an arbitrary given solver. It consists of two steps:

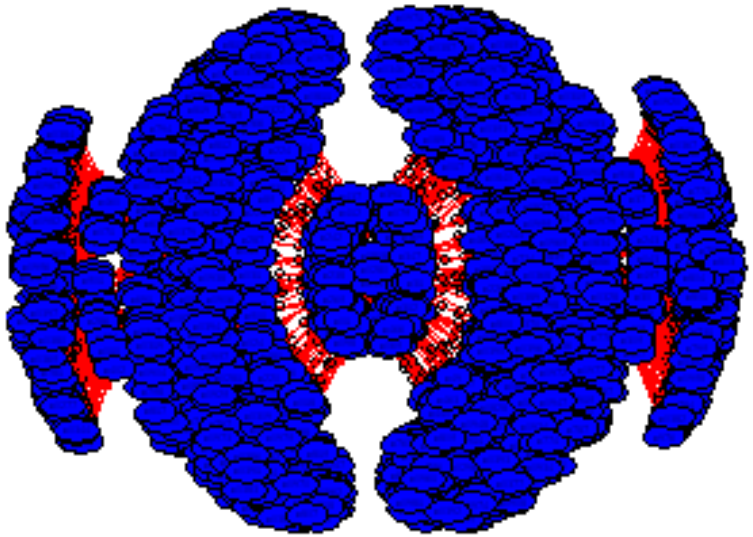
1. Systematical solving of a corpus of formulae for all combinations of strategies
2. Intelligent choosing of suitable strategies with respect to characteristics of the formula being solved

Premise 1

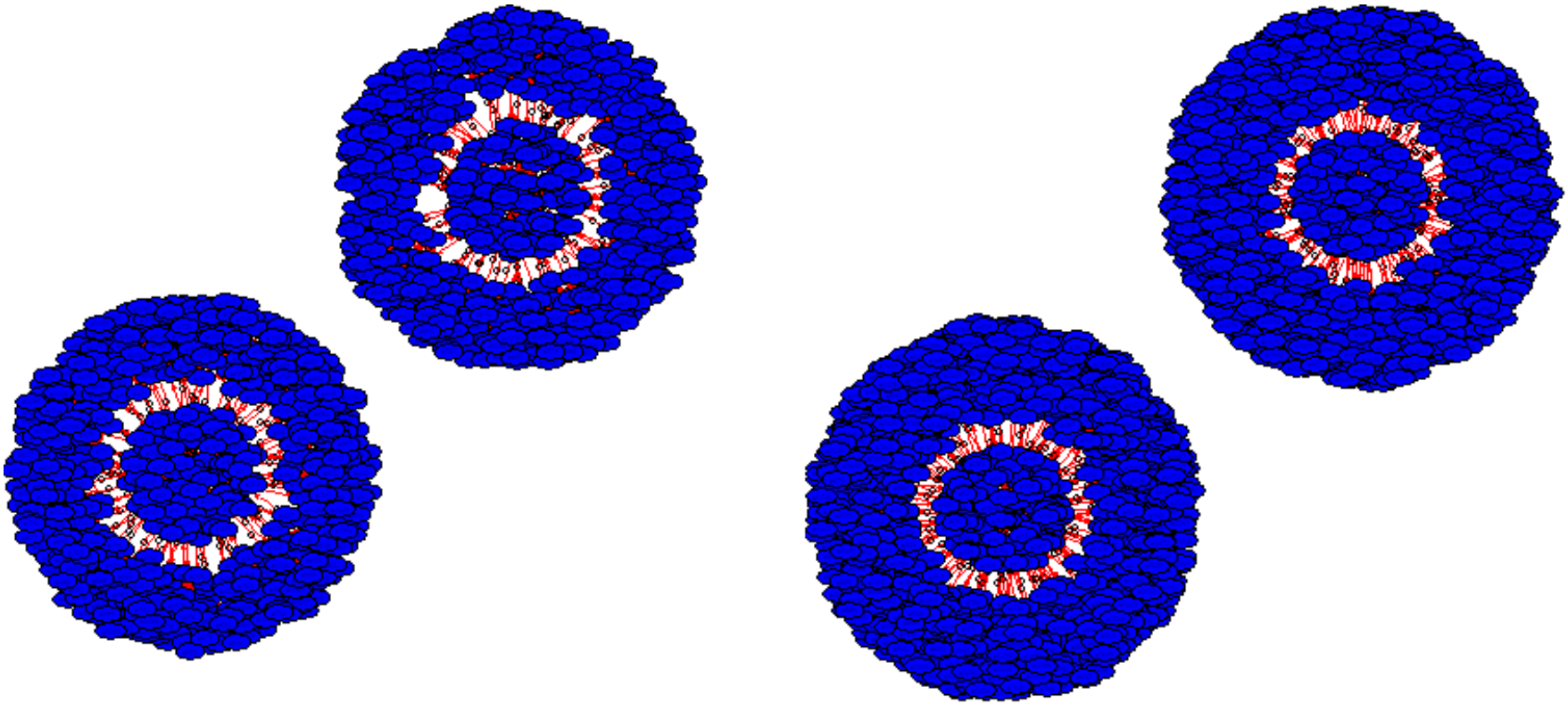
Formulae from the same family share some syntactical properties that can be used for automated formula classification

Family Bart form Corpus SAT 2002

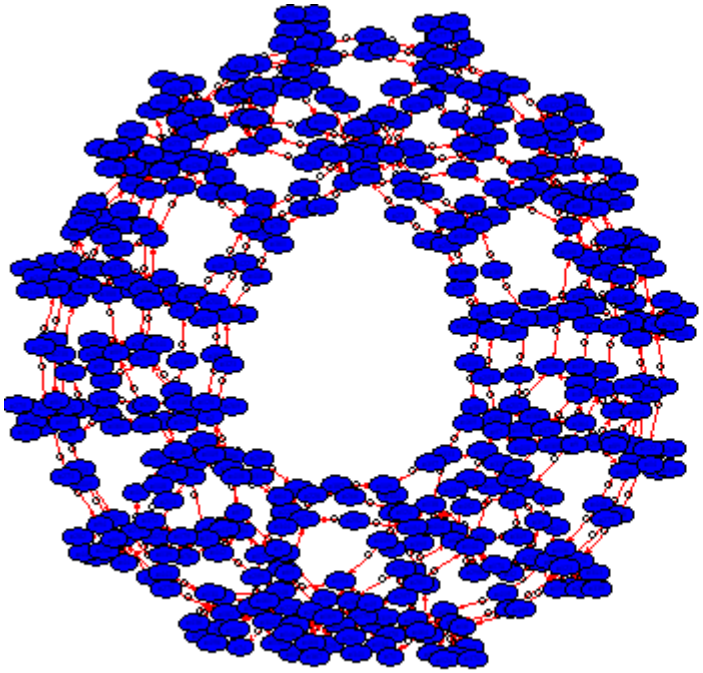


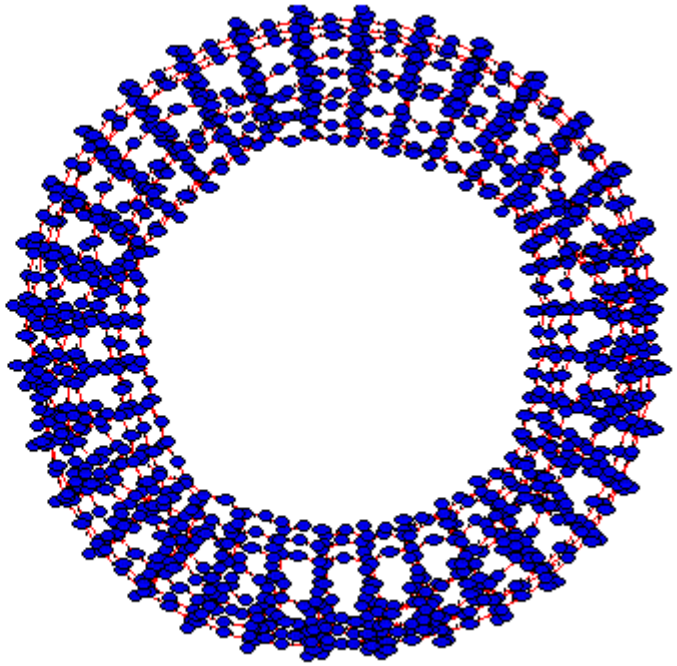


Family Homer from Corpus SAT 2002



Family Rope Bench from Corpus SAT 2002





Premises 2 and 3

- For families of formulae some combinations of strategies show dominantly best results
- For syntactically similar formulae the best combinations of strategies are also similar

Main Message

The main message of this work is that intelligent choosing of solvers' strategies, based on the syntax of the input formula, can significantly improve efficiency of a SAT solver.

Important Aspects of the Methodolgy

Choice of Strategies and Their Admissible Parameter Values

Variable selection

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

Polarity selection

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

Restart strategies

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

Forget strategies

$\mathrm{FS}^{1/3,\,1.0/0.99}_{minisat}$

Choice of Method for Classification of Propositional Formulae

Formulae are classified by kNN method using following distance function:

- $d(\mathcal{P}_1, \mathcal{P}_2) \sum_{x \in features} \frac{|f_1(x) - f_2(x)|}{\sqrt{|f_1(x) f_2(x)|} + 1}$

Choice of Relevant Features of Propositional Formulae

Formulae are represented by a chosen set of syntactical properties:

- 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,
- ...

Choice of Corpus of Formulae for Training and Evaluation

- Corpus SAT 2002
  - 1964 formulae
  - Around 60 families
- Corpus SAT 2007
  - 906 formulae
  - Around 30 families
- Only 12 shared formulae, and significantly different sets of families

Description of Methodology

Training Phase

1. Solving formulae for all admissible combinations of strategies
2. Determining which combinations of strategies are best on average for different families of formulae
3. Making profiles for all the formulae

Exploitation Phase

Parametar: k

1. Profile building for input formula
2. Classification of input formula to some of given families using kNN method
3. Solving input formula using strategies best for the chosen family

Evaluation of Methodology

Training Phase

- ArgoSAT solver was used
- Cutoff time for solving used is 10 minutes
- 60 combinations of strategies
- Clause and variable shuffling
- Number of times that SAT solver was run — 235.680
- A cluster computer with 32 processors was used
- Average profile building time — 0.39s

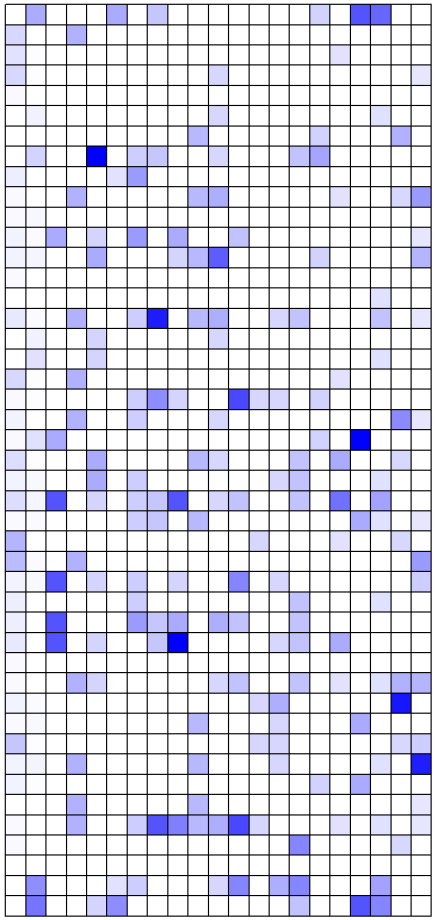
Evaluation of Automated Formulae Classification (P1)

- Corpus SAT 2002
- The leave one out procedure was used
- The best performance was achieved for $k = 1$ and the distance d
- Precision: 92.5%
- Average per family precision: 79.2%
- Classification time when the profile is known — < 0.1s

Analysis of Dominant Combinations of Strategies (P2)

- For each family and each combination we calculated percentage of number of formulae for which that combination is better than the others
- Restriction: at least 10 solved formulae in family
- 15 combinations are not the best for any formula

Analysis of Dominant Combinations of Strategies (P2)



Similarity of Formulae and Their Best Combinations of Strategies (P3)

Computing similarity correlation:

- For each 2 formulae form the corpus do:
  a) Calculate distance d between them (syntactical)
  b) Calculate distance d_c between their best combinations of strategies (semantical)

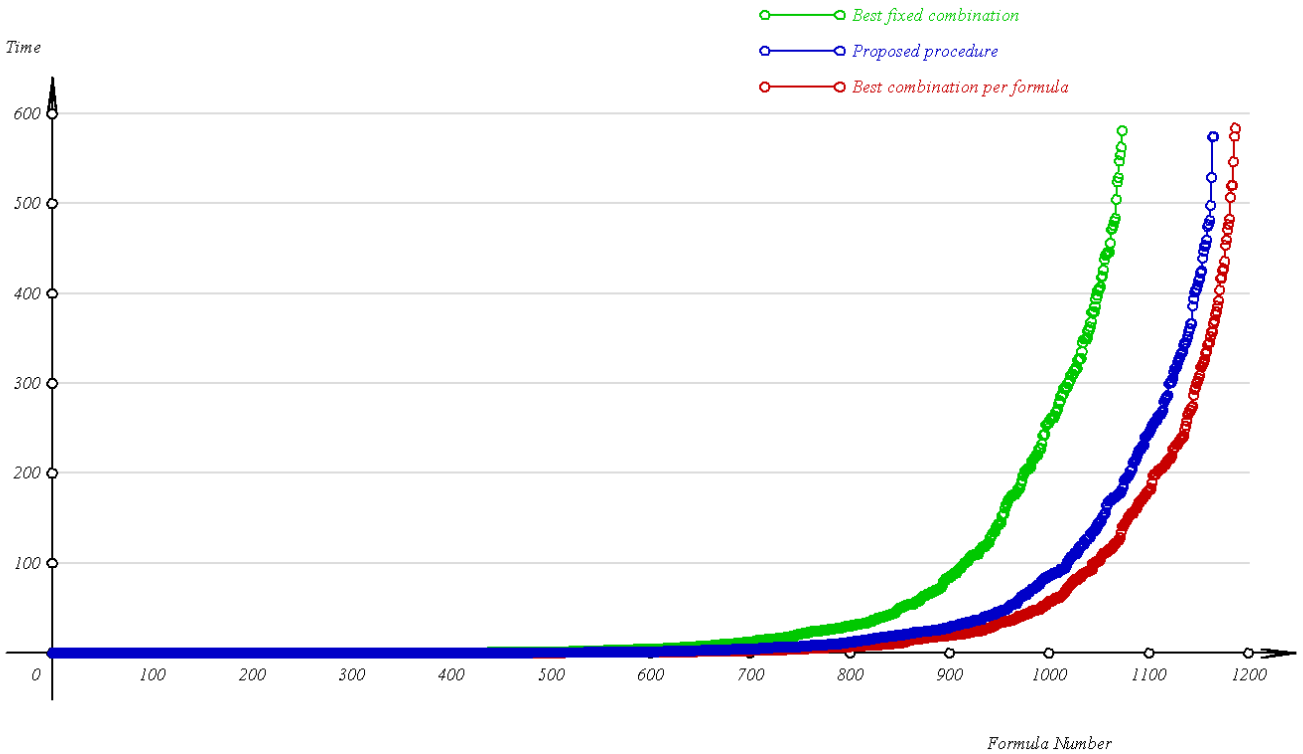
$$d_c(s_1 s_2 s_3, t_1 t_2 t_3) = \sum_{i=1}^{3} c(s_i, t_i)$$

- Pearson correlation coefficient is 0.51
- For syntactically similar formulae restart strategy varies the least, and variable selection strategy the most.

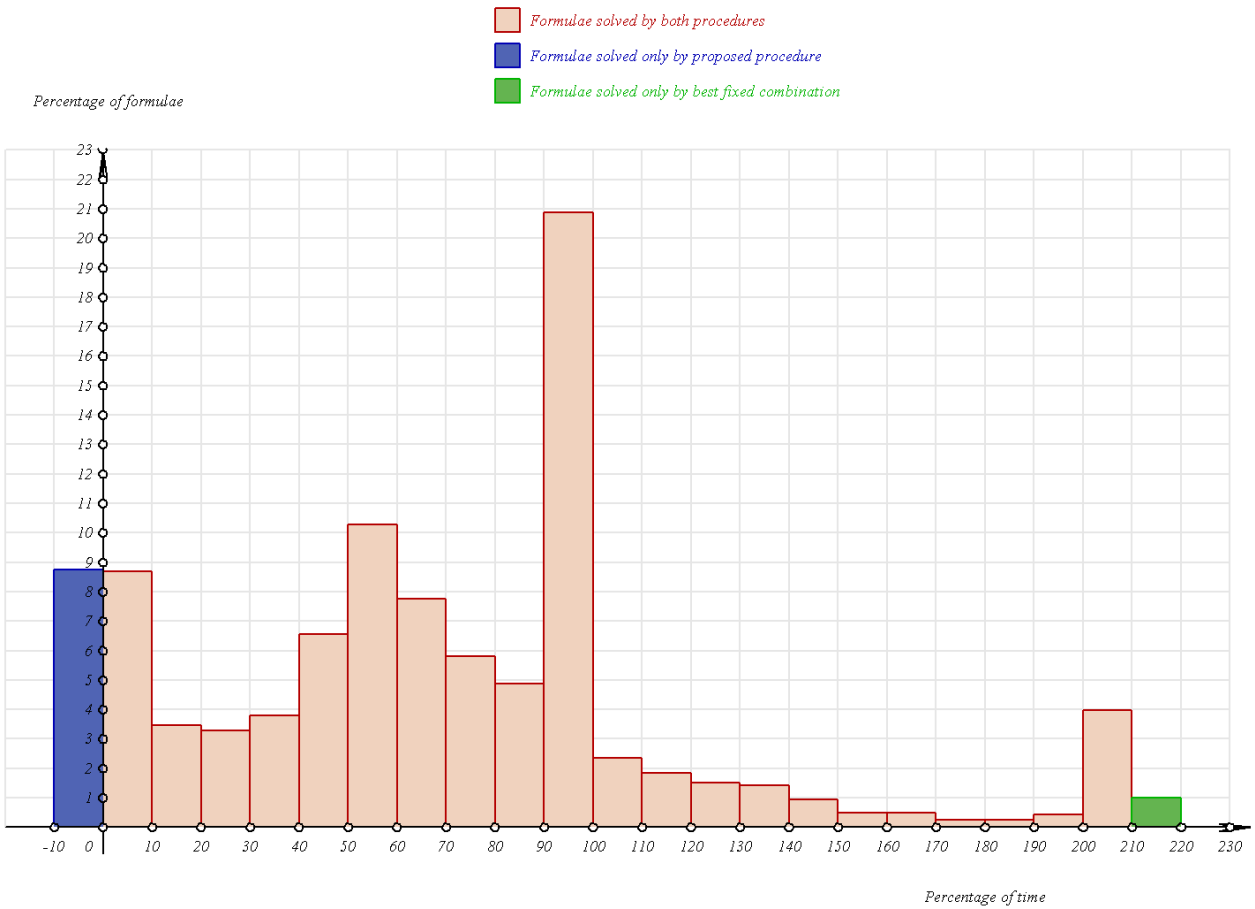
Evaluation of Strategy Selection Procedure

Procedure	No. solved	Median time
Best fixed	1073	207.45s
Proposed	1165	70.64s
Best per formula	1187	46.08

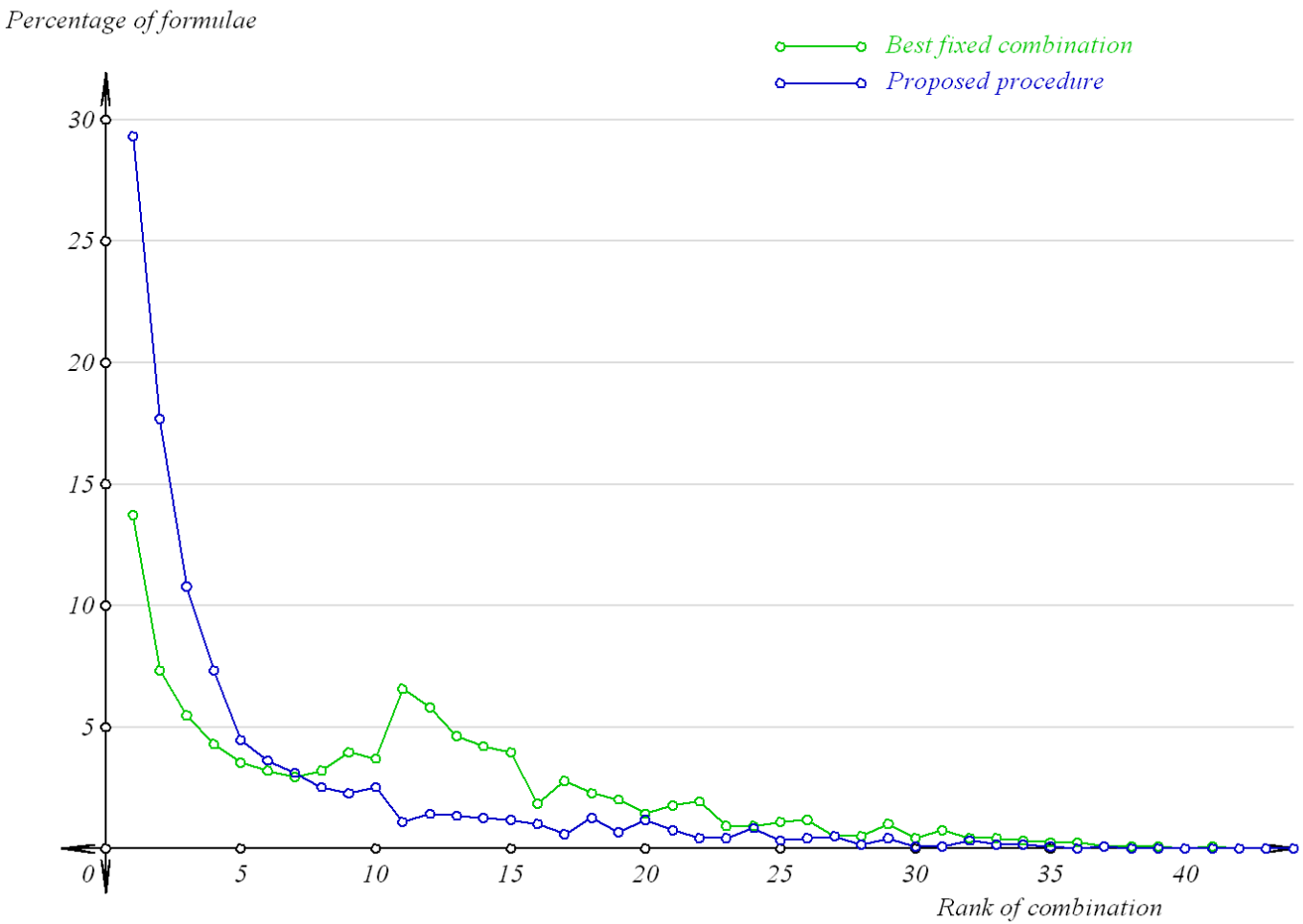
Solving Times



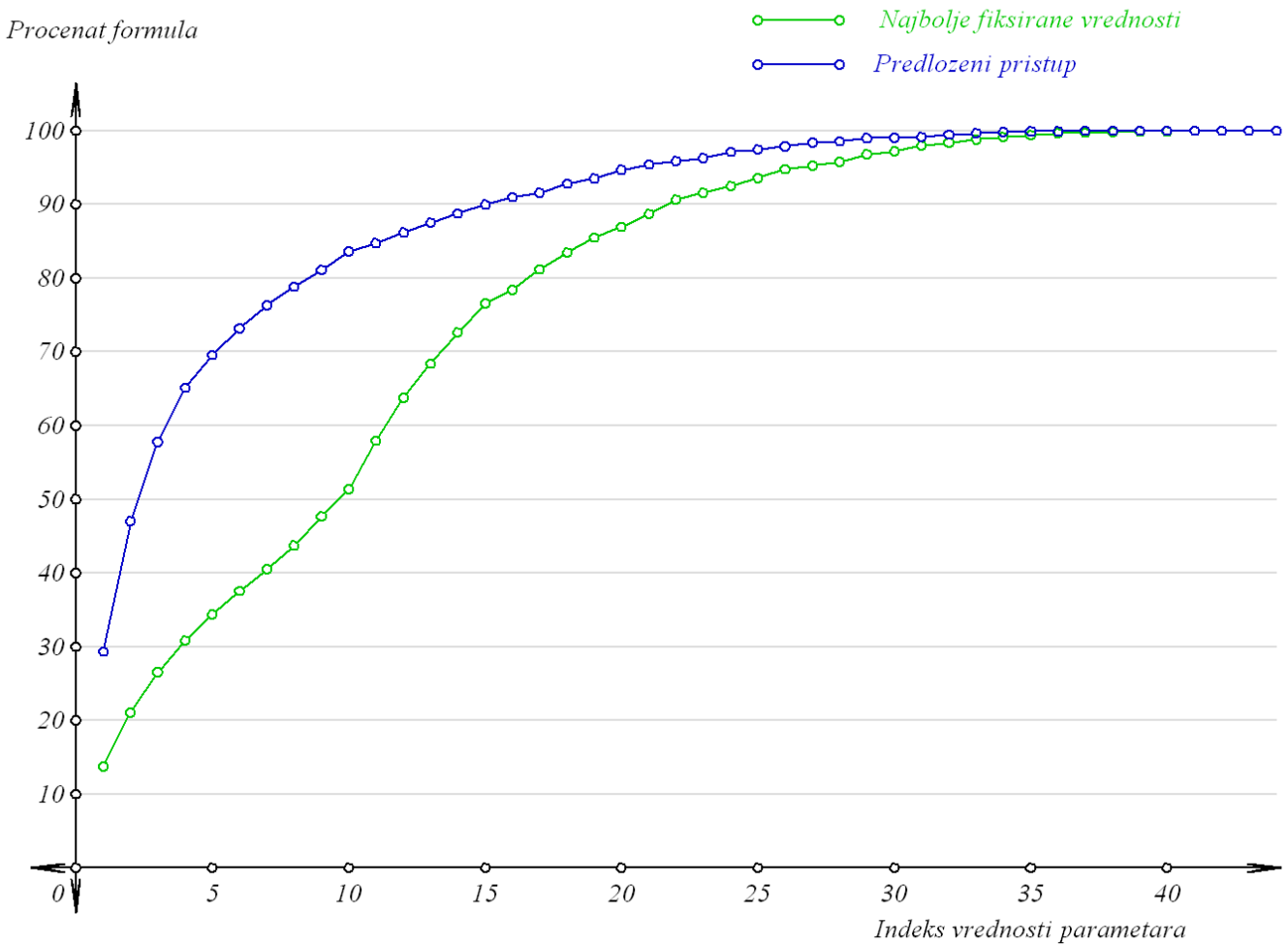
Speedup Histogram



Distribution of Ranks of Chosen Combinations



CDF of Ranks of Chosen Combinations



Exploitation phase: SAT 2007.

System	No. solved	20-th perc. time
ArgoSAT	219	314.16s
ArgoSmArT	253	173.92s

Further work

Further work

- Further analysis of gathered data
- Stochastic parameter optimization
- Testing of stability of best combinations in phase transition region for 3-SAT
- Learning to control SAT solver by reinforcement learning

Conclusions

Conclusions

- A syntactical similarity between the formulae of the same family exists, and it can be used for automatic recognition of family the formula belongs to
- There are no unique dominant combinations of strategies, but there exist small sets of such combinations
- Correlation between formula similarity and similarity of their best combinations of strategies is significant
- Intelligent choosing of solvers' strategies, based on the syntax of the input formula, can significantly improve efficiency of a SAT solver
- Improvements achieved by using this methodology are present even a on different corpus