# Evaluating LTL Satisfiability Solvers

Viktor Schuppan
Supported by the Provincia Autonoma di Trento (project EMTELOS).

j.w.w. Luthfi Darmawan
Supported by the European Master's Program in Computational Logic (EMCL).

ATVA'11, Taipei, Taiwan, October 11–14, 2011

# LTL Satisfiability – (Our) Motivation



Verification gains momentum $\Rightarrow$ specifications become object of interest

Investigation of specifications

Property Simulation (e.g., RAT [PSC+06])

- Example traces, possibly with constraints
- Makes properties executable

Property Assurance (e.g., RAT [PSC+06])

- Possibilities, assertions

Sanity Checks (e.g., RAT [PSC+06]; [RV10]; [FKSFV08])

- Satisfiability, non-validity, non-redundancy

Boil down to LTL satisfiability.

(Note: checks beyond satisfiability are important, too.)

# LTL Satisfiability – (Our) Motivation



## Some specific triggers

Antichains for LTL satisfiability ([WDMR08])

- Claims advantage of **`ALASKA`** over **`NuSMV-BDD`**

LTL satisfiability solver comparison by Rozier and Vardi ([RV10])

- Focus on LTL satisfiability via explicit and BDD-based symbolic model checking, i.e., no SAT-based, temporal-resolution, tableaux-based tools.

Interest in Temporal Resolution and its potential for extraction of unsatisfiable cores ([Sch10])

No recent evaluation of LTL satisfiability solvers using a broad range of algorithms, a broad range of benchmarks, and comprehensive criteria available.

# Objective



Objective: compare performance of
off-the-shelf solvers for propositional LTL satisfiability.

This is a comparison of tools (as opposed to one of algorithms).

- Different features (e.g., preprocessing/simplification).
- Different maturity.
- Different programming languages.

# Outline



1. Introduction
2. LTL Solvers
3. Benchmarks
4. Methodology
5. Findings
6. Conclusions

# Selection of LTL Solvers



## Reduction to Model Checking

- Selected: **ALASKA**, **NuSMV-BDD**, **NuSMV-SBMC**
- Ruled out: explicit state model checkers [RV10], **Cadence SMV** (BDDs) [RV10], **SAL** [RV10], **VIS** (BDDs)
- Last hardware model checking competition [HWMCC10] focuses on safety

## Tableau-Based Algorithms

- Selected: **LWB**, **pltl**
- Ruled out: **TWB**, **LTL Tableau** [GKS09]

## Temporal Resolution

- Selected: **TRP++**, **TSPASS**
- Ruled out: **TeMP** [HKR+04,LH10]

# Selection of Benchmarks



Use families from previous comparisons [WDMR08,RV10,HS02].

- But: restrict number of instances in random category.

Add families not used for LTL satisfiability before: **acacia**, **amba**, **forobots**.

Create new families: **O1formula**, **O2formula**, **phltl**.

Scale up families.

Add variants that enforce non-trivial behavior.

To my knowledge this is the most comprehensive set of benchmarks in comparing propositional LTL satisfiability solvers to date.

# Benchmarks



| Family | Description | #Inst./uns. | Max. $\|\phi\|$ | Source |
|---|---|---|---|---|
| | Category **application** | | | |
| **acacia** | Arbiters and traffic light controllers | 71/- | 426 | [FJR09] |
| **alaska_lift** | Elevator specifications | 136/34 | 4450 | [WDMR08] |
| **alaska_szymanski** | Mutual exclusion protocol | 4/- | 183 | [WDMR08] |
| **anzu_amba** | Microcontroller buffer architecture | 51/- | 6173 | [BGJ+07a] |
| **anzu_genbuf** | Generalized buffer | 60/- | 5805 | [BGJ+07b] |
| **forobots** | Model of a robot with properties | 39/25 | 636 | [BDF09] |
| | Category **crafted** | | | |
| **rozier_counter** | 4 variants of a serial counter | 78/- | 751 | [RV10] |
| **rozier_pattern** | 8 scalable patterns to trigger difficulties in LTL to Büchi translators | 244/- | 7992 | [RV10] |
| **schup._O1/2form.** | Expon. behavior in some solvers | 54/42 | 6001 | |
| **schuppan_phltl** | Temporal variant of pigeonhole | 18/10 | 40501 | |
| | Category **random** | | | |
| **rozier_formulas** | Obtained by generating a syntax tree [DGV99] | 2000/57 | 185 | [RV10] |
| **trp** | Obtained by lifting propositional CNF into fixed temporal structure | 970/397 | 1422 | [HS02] |

# Setup, Flow



## Flow

Preliminary Stage

- Purpose: reduce number of configurations for **TRP++** and **TSPASS**
- 10 second time out

Main Stage

- All remaining configurations for all solvers
- 60 second time out

Graphical Evaluation

- Choose one winning configuration per solver based on highest score

## Setup

Hardware/Software

- Intel Xeon 3.0 GHz
- 4 GB memory
- Red Had Linux 5.4, 64 bit kernel 2.6.18
- Measure time, memory with **run** [BJ]

No shuffling of benchmarks

One run per instance and solver configuration

Memory out: 2 GB

# Scoring



Objective:

- Score by highest number of solved instances; break ties by lower time taken on solved instances. (Frequently used.)

Problem:

- Benchmark families with very different numbers of instances.
- Smallest family has 4 instances; largest has $> 2000$.

Solution:

- Arrange benchmark families in tree.
- Assign equal weight to the (immediate) children of each node.

Caveat:

- The weight of an instance may change between different scores, e.g., share of solved instances (all instances count) and run time on solved instances (only solved instances count).
- The weight of an instance may change between different solvers for the same score, e.g., run time on solved instances (only instances solved by particular solver count).

# Cactus Plots vs. Contour/Discrete Raw Data Plots



## Cactus Plots

- are standard;
- easily allow to identify the winner when ranking by highest number of solved instances with ties broken by time spent on solved instances;
- break the correlation between different solvers on the same instance.

## Contour/Discrete Raw Data Plots

- retain the correlation between different solvers on the same instance;
- easily allow to identify similar and complementary behavior;
- easily allow to see performance of a solver on subfamilies;
- easily allow to see difficulty of instances and subfamilies.

# Correctness



We found 1 or 2 bugs each in **ALASKA**, **NuSMV**, **TRP++**, and **TSPASS**.

- Kindly fixed quickly by tool authors.

We also found bugs in **LWB**.

- We contacted the developers.
- We received no response.
- 187 out of 7446 instances (known) buggy.
- 13 wrong results.
- Others abnormal termination.
- Hors concours.

# Winning Configurations per Tool



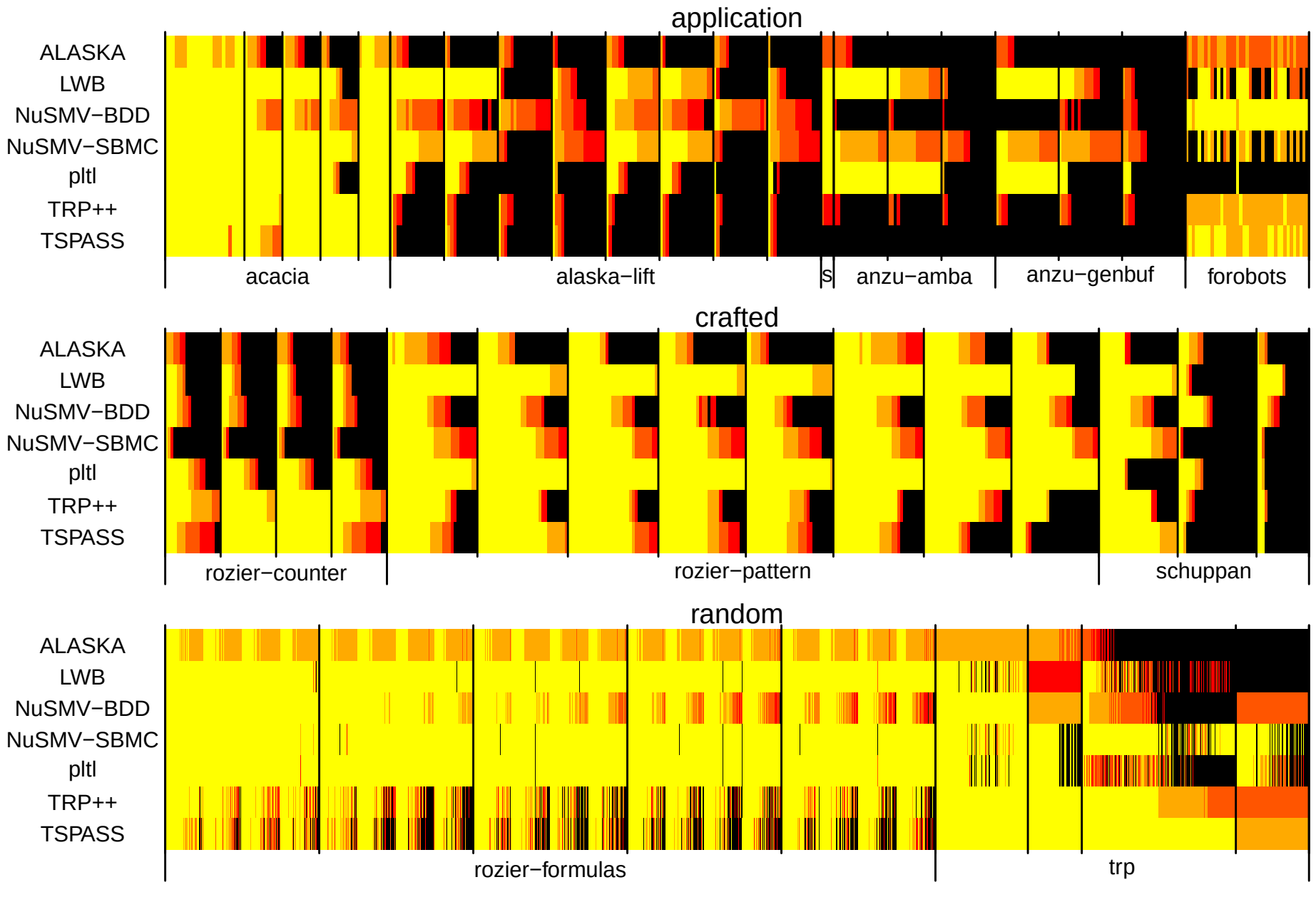
We select one winning configuration (column max) per tool.

| | model construction disabled (*sat* and *unsat* instances) | | | |
|---|---|---|---|---|
| tool | winning configuration | max | min | vbs |
| **ALASKA** | `noc_nos_nob` | 0.581 | 0.322 | 0.595 |
| **LWB** | `mod` | 0.740 | 0.656 | 0.800 |
| **NuSMV-BDD** | `dcx_fflt_dyn_elbwd` | 0.743 | 0.607 | 0.823 |
| **NuSMV-SBMC** | `nodcx_c` | 0.723 | 0.651 | 0.726 |
| **pltl** | `tree` | 0.694 | 0.687 | 0.702 |
| **TRP++** | `s_r_noal_bfs_nop_fsr` | 0.752 | 0.593 | 0.776 |
| **TSPASS** | `ext_nogrp_nosev_sub_nosls_rfmrr-_norbmrr_nomod_mor` | 0.667 | 0.479 | 0.670 |

Numbers: weighted share of solved instances. vbs: virtual best solver.

# Run Times (Contour/Discrete Raw Data Plots)

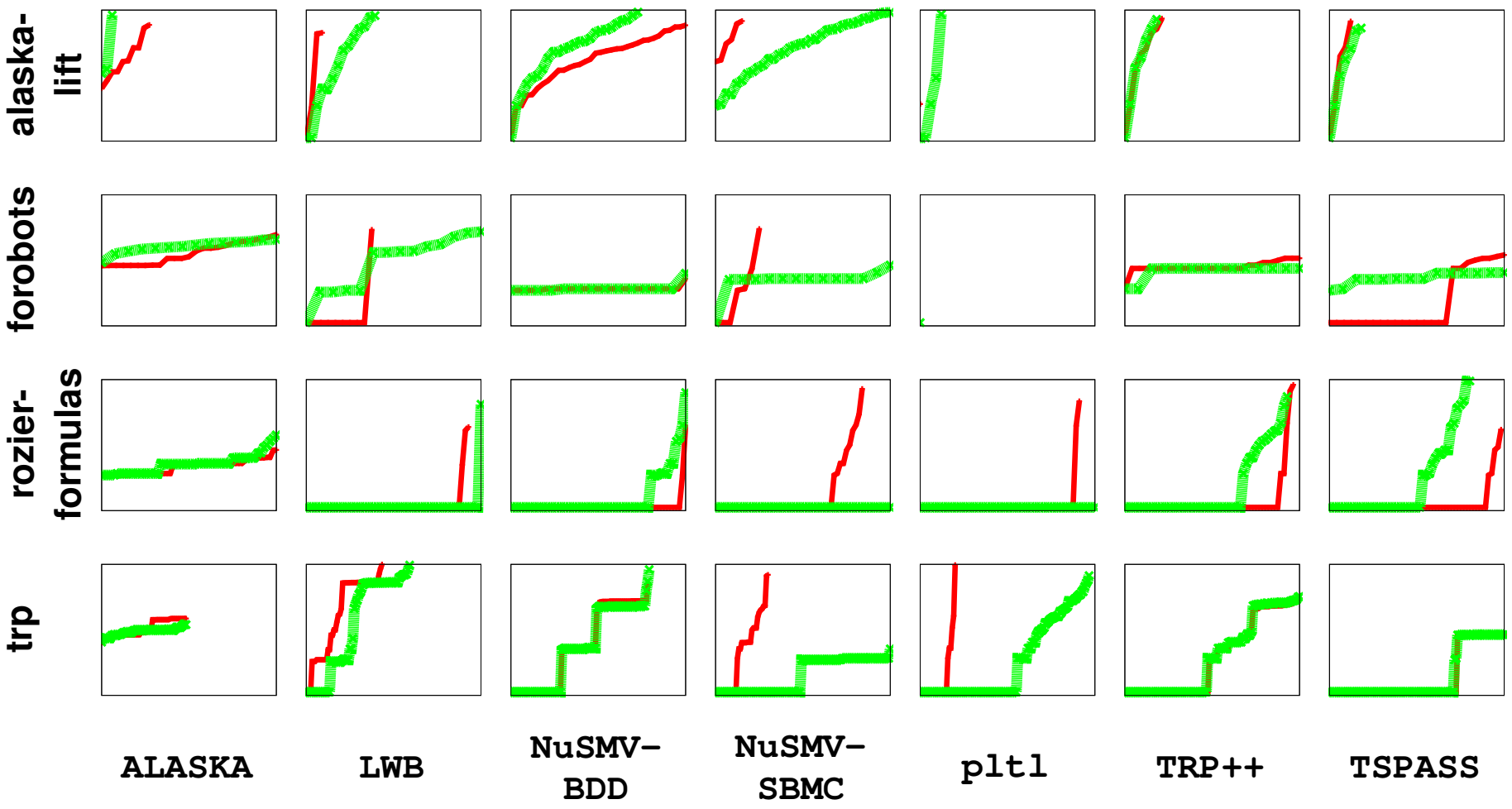





Author: V. Schuppan

# Run Times *Sat* vs. *Unsat* Instances






key: *sat* *unsat*

x-axes: number of solved instances

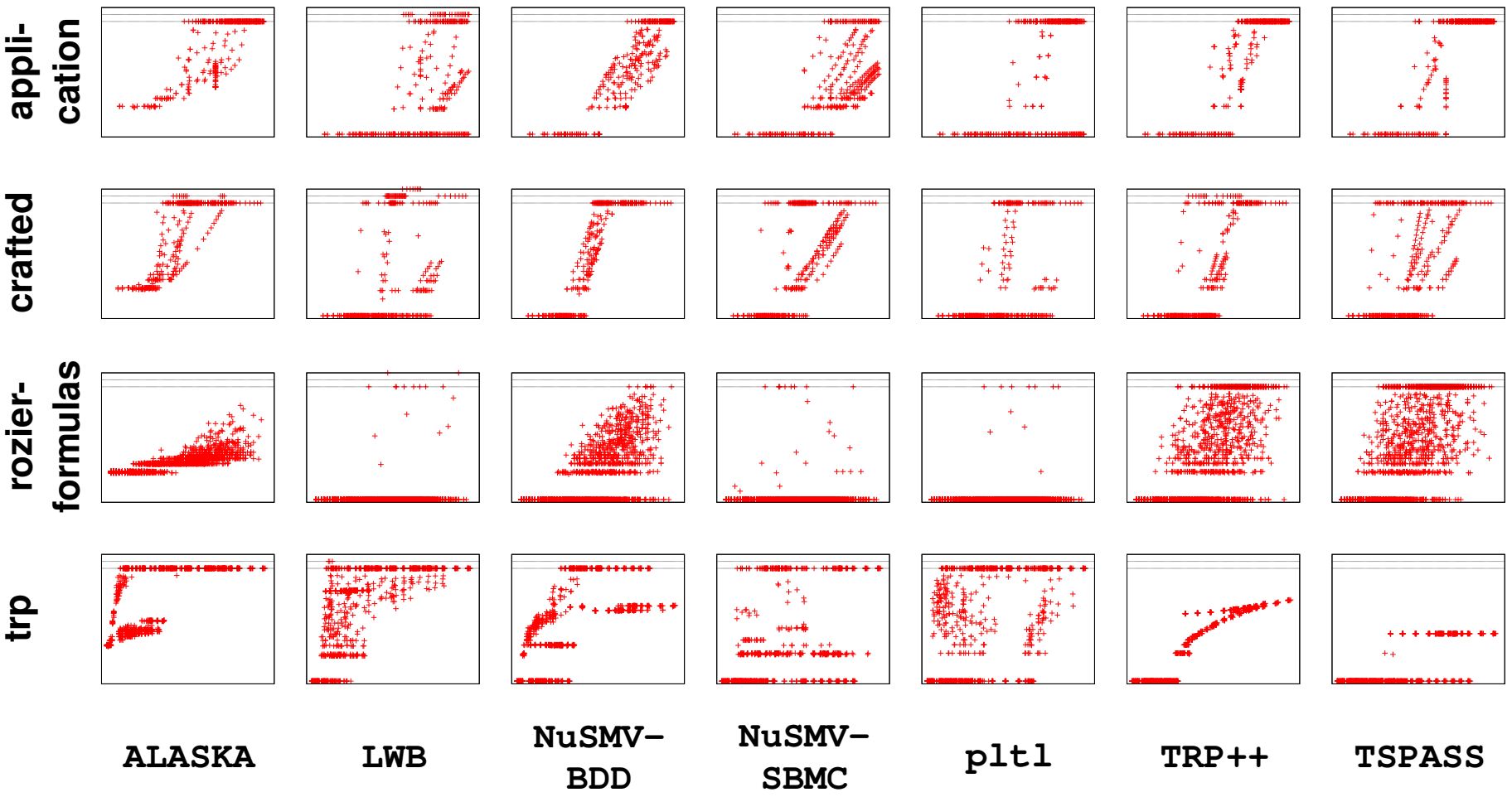
y-axes: run time

Instances selected to “equalize” features.

Author: V. Schuppan

# Run Times by Instance Size

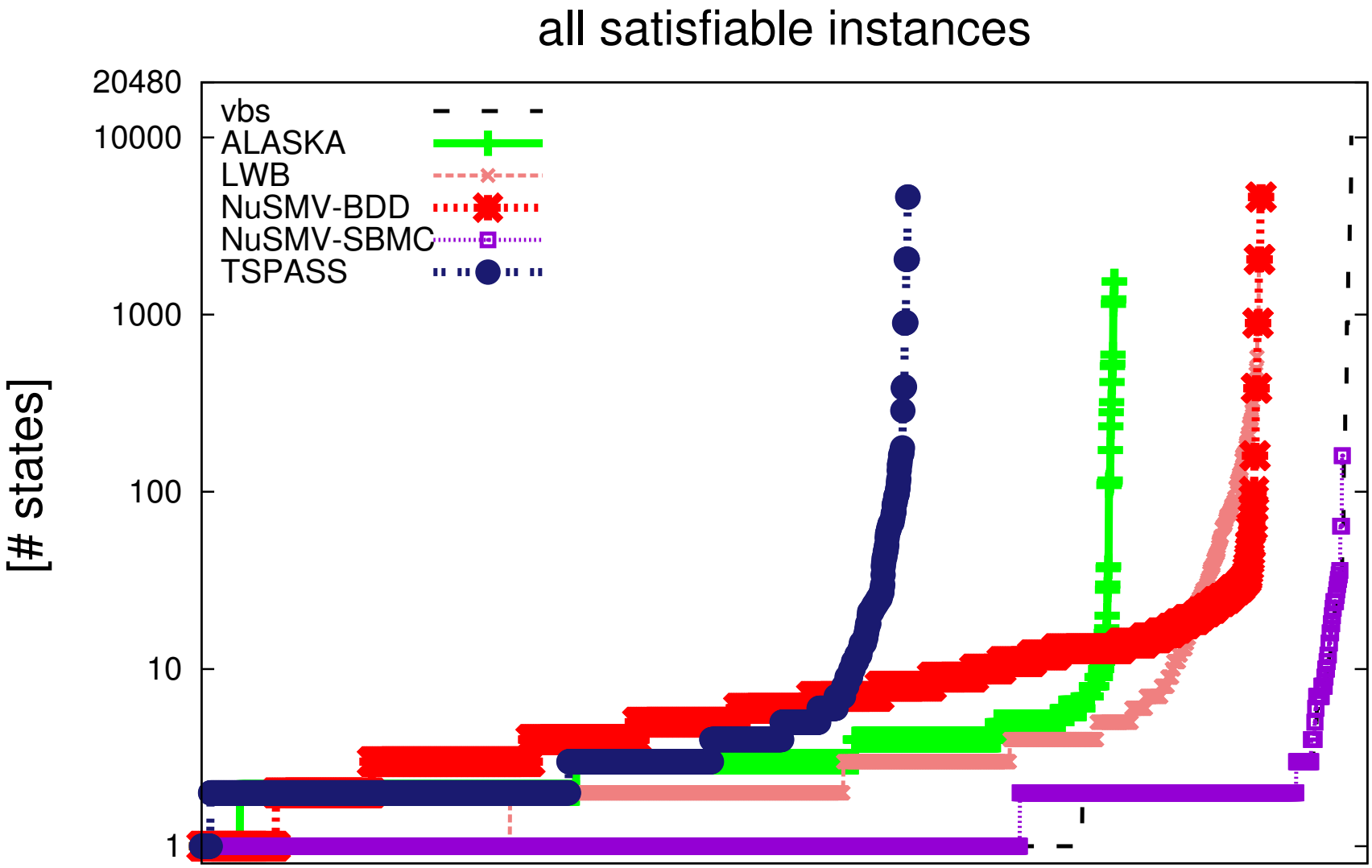





x-axes: instance size
y-axes: run time

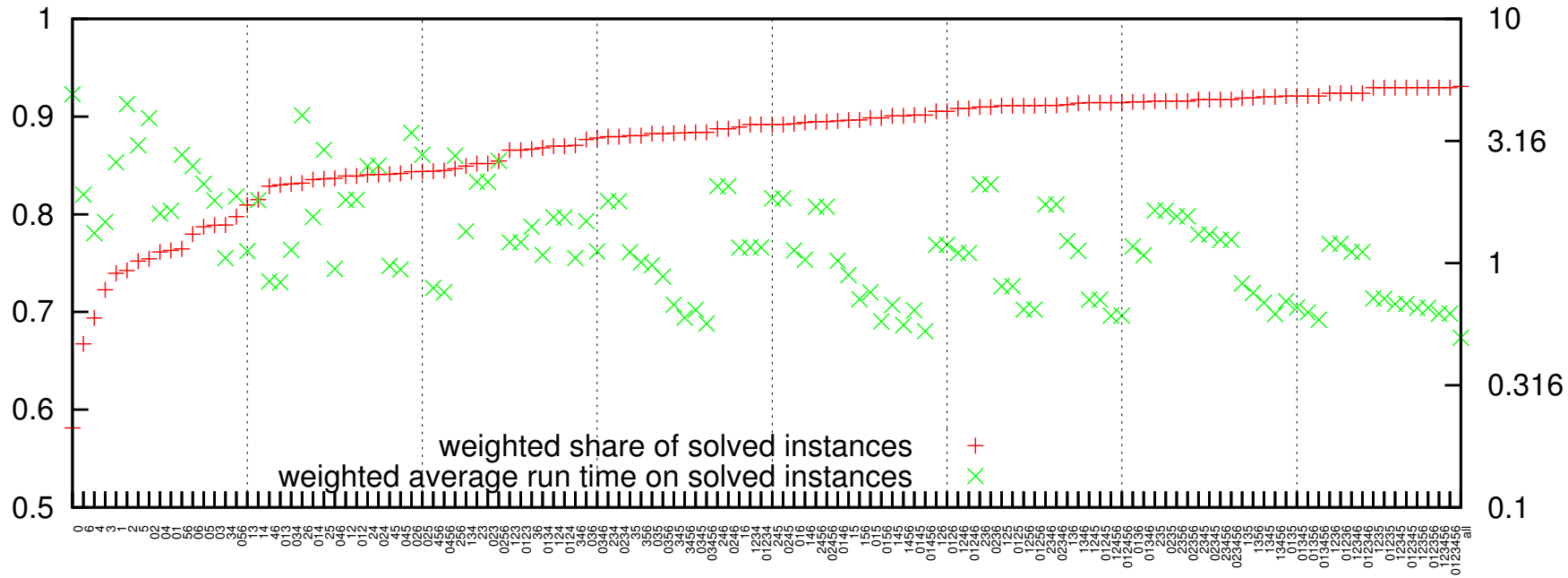
# Model Sizes




all satisfiable instances
20480
10000
1000
100
10
1
[# states]
vbs
ALASKA
LWB
NuSMV-BDD
NuSMV-SBMC
TSPASS

# Potential of a Portfolio Solver (Perfect Oracle)





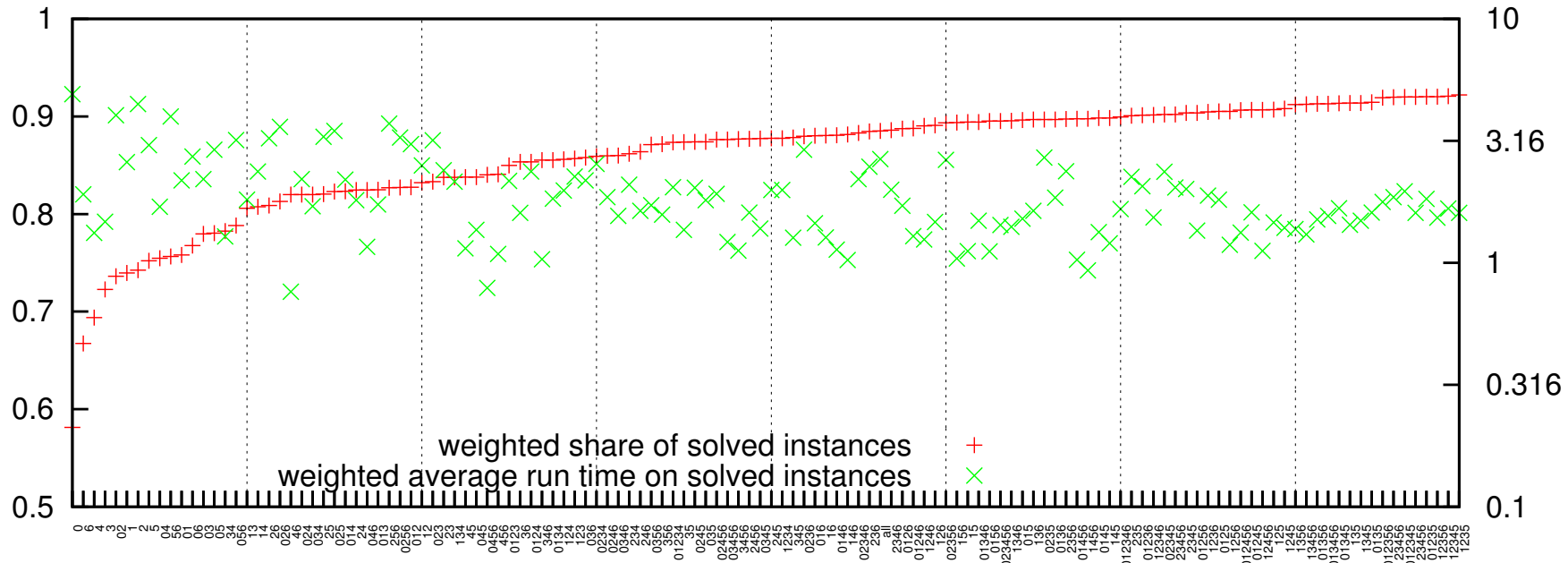

Mode: perfect oracle selects best portfolio member for any given instance. Best case scenario for portfolio solvers without communication between portfolio members. Unrealistic.

Left y-axis: weighted share of solved instances.

Right y-axis: weighted average run time on solved instances [seconds].

# Potential of a Portfolio Solver (Perfect Task Switcher)






Mode: task switching with no overhead and infinitely small time slices between portfolio members. Reference case scenario for portfolio solvers without communication between portfolio members. Should be beaten.

Left y-axis: weighted share of solved instances.

Right y-axis: weighted average run time on solved instances [seconds].

# Potential of a Portfolio Solver (Fast Presolver)



Even a simplistic portfolio solver can yield considerable benefits.

Pick 4 best 2-configuration portfolios.

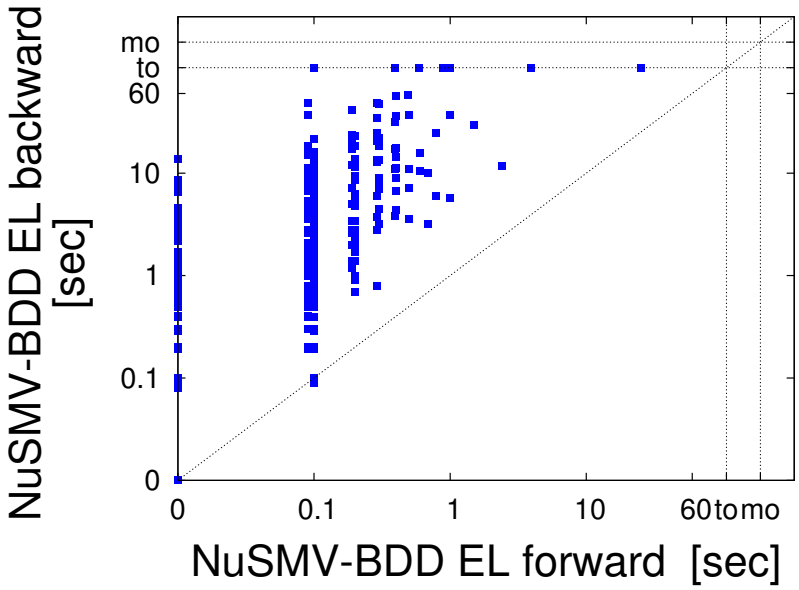
Run one as fast presolver [XHH+08] for a short time. If that fails, run the other solver for the remaining time.

| Results | 1st as fast presolver | | | | 2nd as fast presolver | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 second | | 2 seconds | | 1 second | | 2 seconds | |
| | share | time | share | time | share | time | share | time |
| (**LWB**, **TRP++**) | 0.880 | 1.09 | 0.885 | 1.30 | 0.841 | 1.26 | 0.850 | 1.45 |
| (**LWB**, **TSPASS**) | 0.868 | 0.88 | 0.874 | 1.10 | 0.850 | 1.20 | 0.858 | 1.48 |
| (**NuSMV-SBMC**, **TRP++**) | 0.823 | 1.03 | 0.841 | 1.18 | 0.860 | 0.97 | 0.862 | 1.31 |
| (**NuSMV-SBMC**, **TSPASS**) | 0.813 | 1.00 | 0.831 | 1.21 | 0.837 | 1.17 | 0.840 | 1.42 |

| Reference | 1st in isolation | | 2nd in isolation | | perfect oracle | | perf. task switcher | |
|---|---|---|---|---|---|---|---|---|
| | share | time | share | time | share | time | share | time |
| (**LWB**, **TRP++**) | 0.740 | 2.59 | 0.752 | 3.03 | 0.896 | 0.89 | 0.894 | 1.12 |
| (**LWB**, **TSPASS**) | 0.740 | 2.59 | 0.667 | 1.91 | 0.889 | 1.16 | 0.881 | 1.27 |
| (**NuSMV-SBMC**, **TRP++**) | 0.723 | 1.47 | 0.752 | 3.03 | 0.880 | 1.11 | 0.874 | 1.37 |
| (**NuSMV-SBMC**, **TSPASS**) | 0.723 | 1.47 | 0.667 | 1.91 | 0.867 | 1.41 | 0.853 | 1.60 |

# A Performance Advantage of ALASKA over NuSMV-BDD?






[WDMR08] claims an advantage of **ALASKA** over **NuSMV-BDD**.

A difference beyond representation (antichains vs. BDDs) was direction of fixed point computations. (Forward fixed point computation was not available in **NuSMV-BDD** when [WDMR08] was done.)

Using appropriate options in **NuSMV** and forward rather than backward fixed point computation **ALASKA** does not outperform **NuSMV-BDD**.

# The End



## Summary

Identification of reference solvers with options at instance level.

No solver dominates. Rather, complementary behavior.

We don't declare any single solver to be the winner.

A portfolio approach seems worth trying.

Benchmarks, data, more plots available:
`http://www.schuppan.de/viktor/atva11/`.

## Future Work

Check out participants of HWMCC'11.

Consider explicit state model checkers that handle the property on-the-fly.

Have a proper competition?

# Thanks to

... you for your attention,

... J.-F. Raskin and N. Maquet for help with **ALASKA** and hosting the first author for one week,

... R. Goré and F. Widmann for help with **pltl**,

... B. Konev and M. Ludwig for help with **TRP++** and **TSPASS**,

... C. Dixon for the **forobots** family,

... B. Jobstmann and G. Hofferek for help with the **amba** family,

... K. Rozier for feedback,

... A. Artale for supervising the second author's MSc thesis,

... the ES group at FBK, esp. A. Cimatti, A. Mariotti, and M. Roveri for discussion and support.

# Questions?

# References



**BDF09** A. Behdenna, C. Dixon, and M. Fisher. Deductive verification of simple foraging robotic behaviours. International Journal of Intelligent Computing and Cybernetics, 2009.

**BGJ+07a** R. Bloem, S. Galler, B. Jobstmann, N. Piterman, A. Pnueli, M. Weiglhofer. Automatic hardware synthesis from specifications: a case study. DATE'07.

**BGJ+07b** R. Bloem, S. Galler, B. Jobstmann, N. Piterman, A. Pnueli, M. Weiglhofer. Specify, compile, run: Hardware from PSL. COCV'07.

**FJR09** E. Filiot, N. Jin, J. Raskin. An antichain algorithm for LTL realizability. CAV'09.

**FKSFV08** D. Fisman, O. Kupferman, S. Sheinvald-Faragy, M. Vardi. A Framework for Inherent Vacuity. HVC'08.

**GKS09** V. Goranko, A. Kyrikov, D. Shkatov: Tableau Tool for Testing Satisfiability in LTL: Implementation and Experimental Analysis. M4M'09.

**HKR+04** U. Hustadt, B. Konev, A. Riazanov, A. Voronkov. TeMP: A Temporal Monodic Prover. IJCAR'04.

**HS02** U. Hustadt, R. Schmidt. Scientific benchmarking with temporal logic decision procedures. KR'02.

**HWMCC10** A. Biere, K. Claessen. Hardware Model Checking Competition (presentation, slides only). HWVW'10.

**LH10** M. Ludwig and U. Hustadt. Implementing a fair monodic temporal logic prover. AI Commun 23 (2-3) 2001: 69–96.

**PSC+06** I. Pill, S. Semprini, R. Cavada, M. Roveri, R. Bloem, A. Cimatti. Formal analysis of hardware requirements. DAC'06.

**RV10** K. Rozier and M. Vardi. LTL satisfiability checking. STTT, 12(2):123–137, 2010.

**Sch10** V. Schuppan. Towards a Notion of Unsatisfiable and Unrealizable Cores for LTL. Science of Computer Programming, 2010. In press.

**WDMR08** M. De Wulf, L. Doyen, N. Maquet, and J.-F. Raskin. Antichains: Alternative algorithms for LTL satisfibility and model-checking. TACAS'08.

**XHH+08** L. Xu, F. Hutter, H. Hoos, and K. Leyton-Brown. SATzilla: Portfolio-based Algorithm Selection for SAT". J. Artif. Intell. Res. (JAIR) 32 (2008): 565–606.