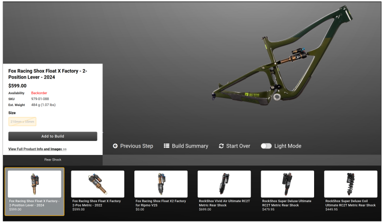
Evaluating State-of-the-Art #SAT Solvers on Industrial Configuration Spaces

Chico Sundermann[1], Tobias Heß[1], Michael Nieke[2], Paul M. Bittner[1], Jeffrey M. Young[4], Thomas Thüm[1], Ina Schaefer[2,3]

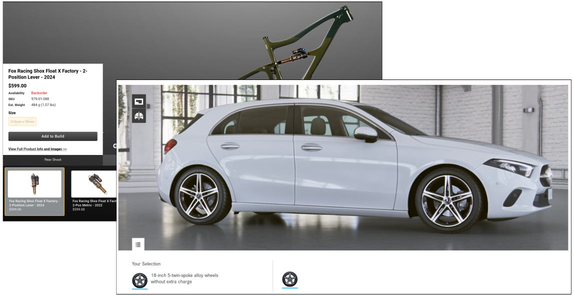
SE'24 | 27.02.2024

1) universität ulm
2) Technische Universität Braunschweig
3) KIT Karlsruher Institut für Technologie
4) Oregon State University

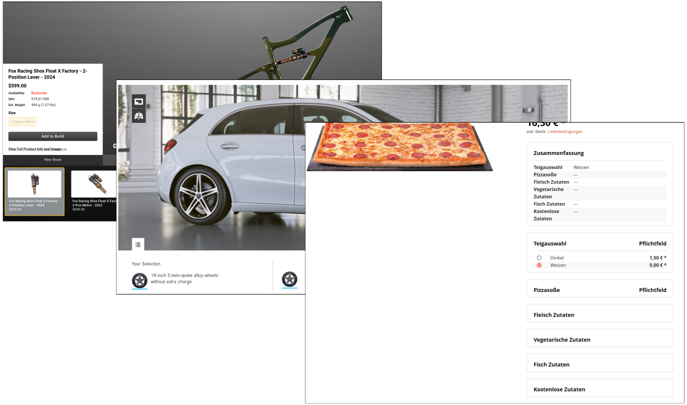
Configurable Systems



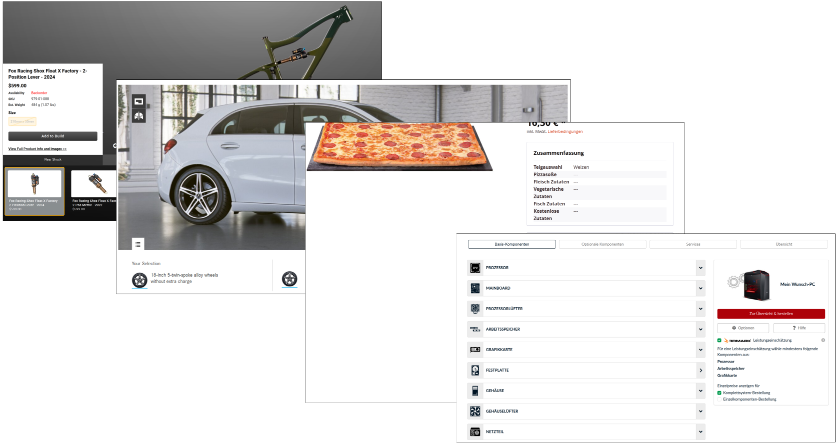
Configurable Systems



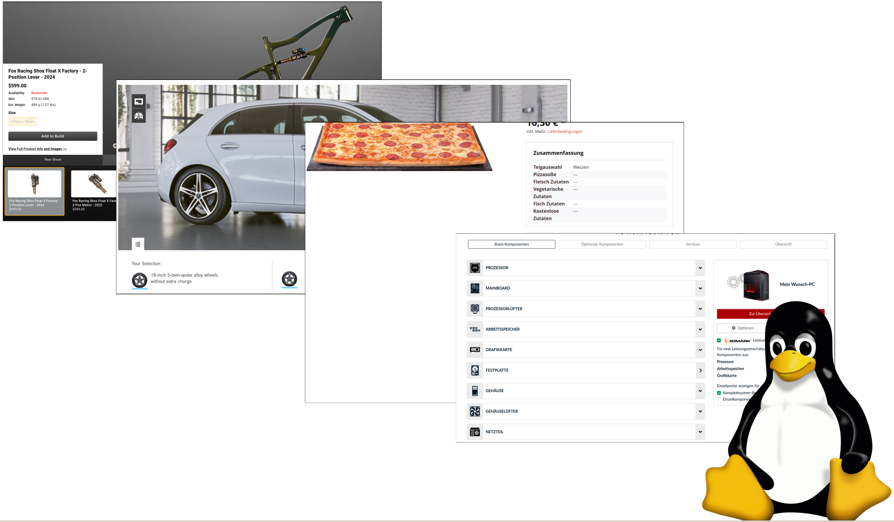
Configurable Systems



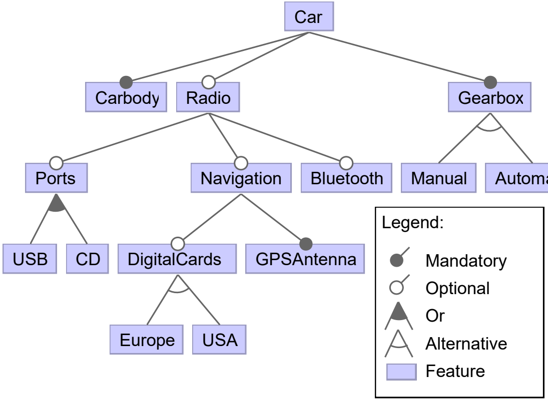
Configurable Systems



Configurable Systems

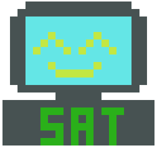


Feature Models



Navigation ⇒ USB

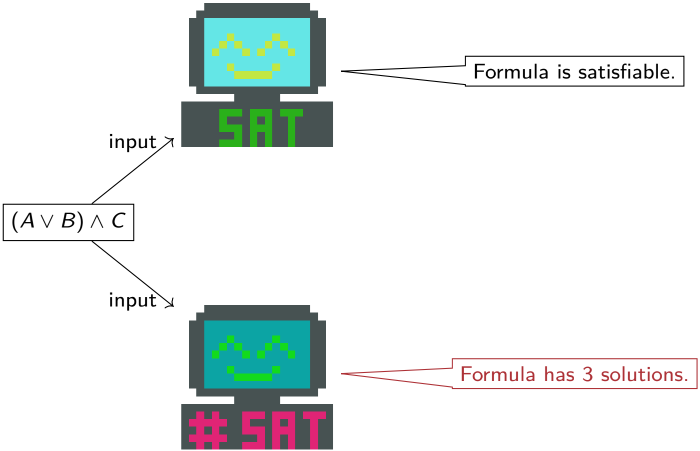
What is SAT?



$(A \vee B) \wedge C$ input

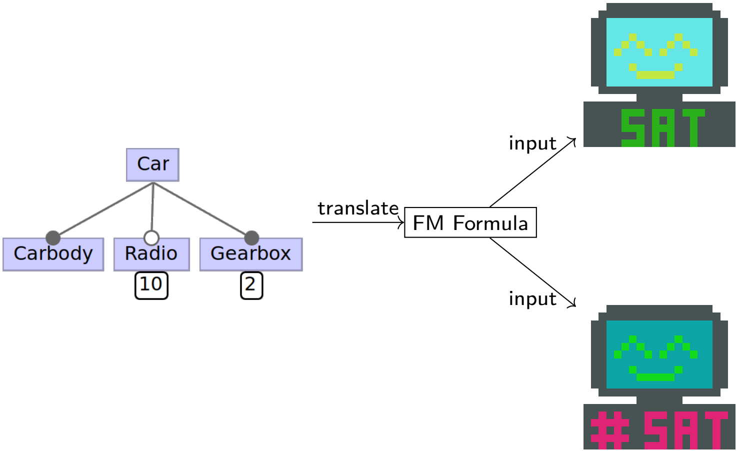
Formula is satisfiable.

What is #SAT?

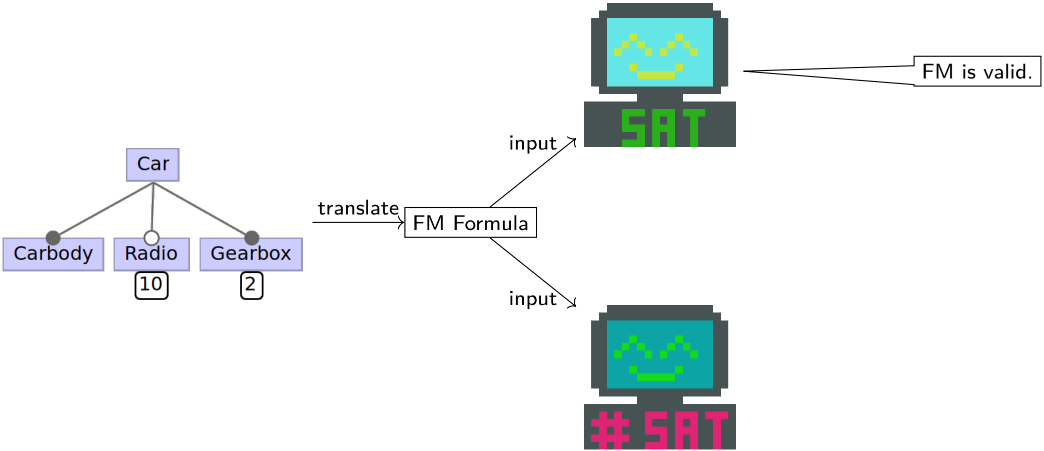


$(A \vee B) \wedge C$
input
input
SAT
Formula is satisfiable.
#SAT
Formula has 3 solutions.

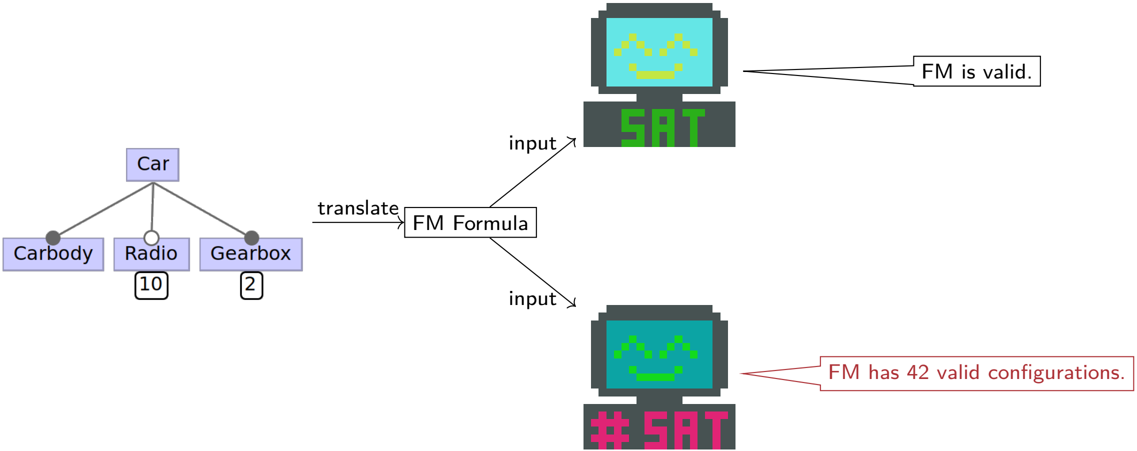
What is #SAT?


Car
Carbody
Radio
Gearbox
10
2
translate
FM Formula
input
input
SAT
SAT

What is #SAT?


Car
Carbody
Radio
Gearbox
10
2
translate
FM Formula
input
input
SAT
#SAT
FM is valid.

What is #SAT?



Car
Carbody
Radio
Gearbox
10
2
translate
FM Formula
input
input
SAT
FM is valid.
#SAT
FM has 42 valid configurations.

Variability Factor

Homogeneity

Uniform Random Sampling

Applications of #SAT Solvers on Feature Models

Chico Sundermann
University of Ulm, Germany

Michael Nieke
TU Braunschweig, Germany

Paul Maximilian Bittner
University of Ulm, Germany

Tobias Heß
University of Ulm, Germany

Thomas Thüm
University of Ulm, Germany

Ina Schaefer
TU Braunschweig, Germany

Configuration Relevance

Feature Prioritization

Rating Errors

CTC Restrictiveness

Variability Reduction

Optimize Configuring

Subset Variability

- Variability Factor
- Void Feature Moddel
- Degree of Orthogonality
- Cost Savings
- Maintainability Prediction
- Configuration Relevance
- Rating Errors
- Variability Reduction
- Subset Variability

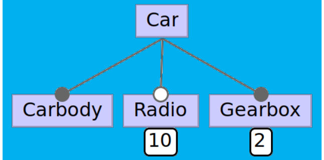
- Homogeneity
- Payoff Threshold
- Degree of Reuse
- Core, Dead & False-Optional
- Atomic Set Candidates
- Feature Prioritization
- CTC Restrictiveness
- Optimize Configuring

- Uniform Random Sampling
- Atomic Sets
- Rate Interactions
- Configuration Derivation

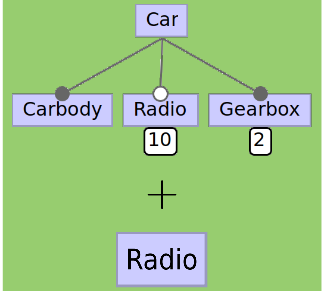
- Variability Factor
- Void Feature Moddel
- Degree of Orthogonality
- Cost Savings
- Maintainability Prediction
- Configuration Relevance
- Rating Errors
- Variability Reduction
- Subset Variability

- Homogeneity
- Payoff Threshold
- Degree of Reuse
- Core, Dead & False-Optional
- Atomic Set Candidates
- Feature Prioritization
- CTC Restrictiveness
- Optimize Configuring

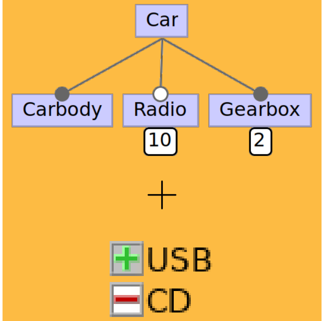
- Uniform Random Sampling
- Atomic Sets
- Rate Interactions
- Configuration Derivation



Cardinality of
Feature Models

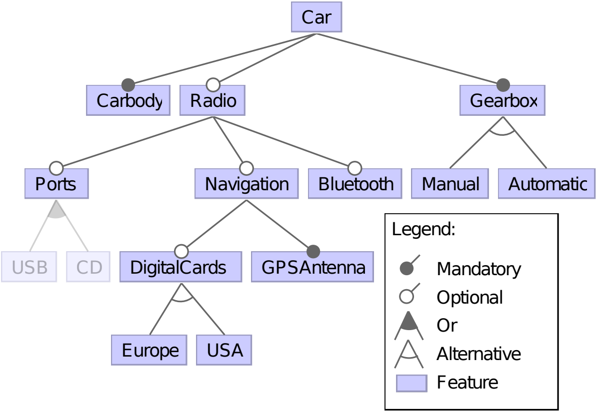


Cardinality of
Features



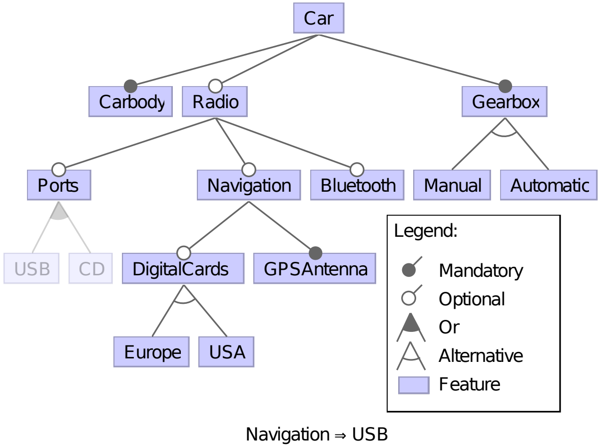
Cardinality of
Partial Configurations

Feature Prioritization

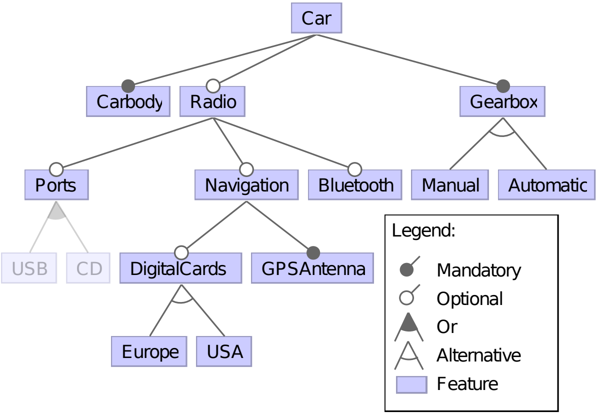


Navigation ⇒ USB

Feature Prioritization



Which feature to develop first?



Navigation ⇒ USB

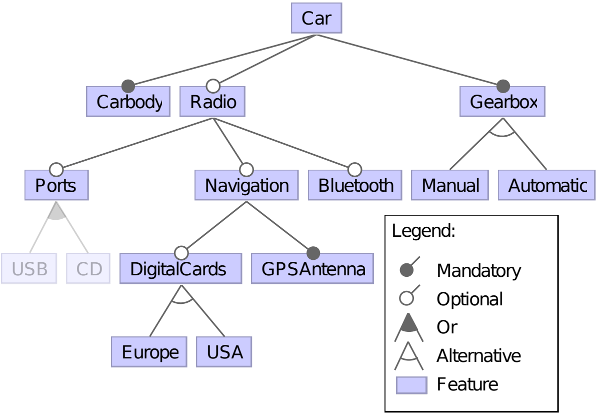
Feature Prioritization

Which feature to develop first?

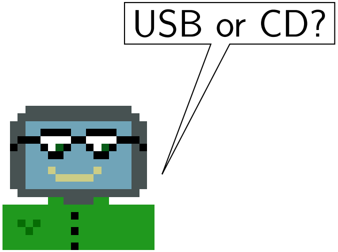
↓

Prioritize based on cardinality

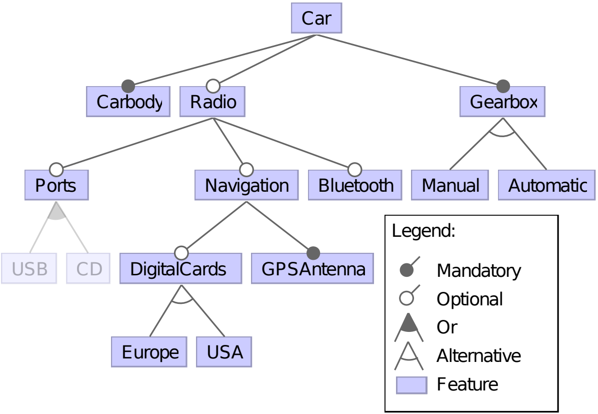
Feature Prioritization



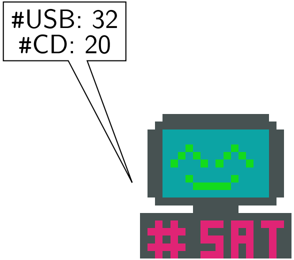
Navigation ⇒ USB



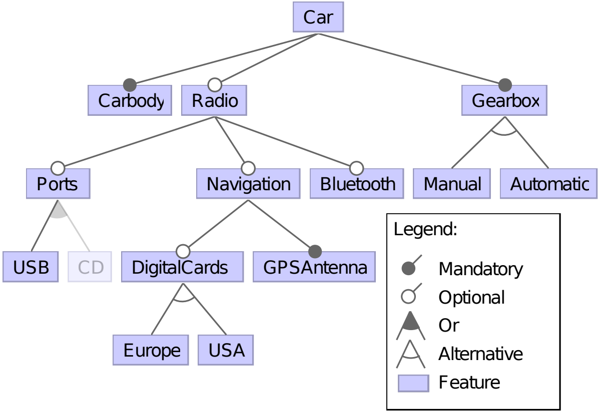
Feature Prioritization



Navigation ⇒ USB



Feature Prioritization



Navigation ⇒ USB

Prioritize USB
#USB > #CD

Experiment Design

- 17 Solvers (15 exact, 2 approximate)
  - 7 DPLL Solvers (Single query)
  - 8 Knowledge Compilers (3 d-DNNF, 3 BDD, 2 other)

DPLL

PicoSAT Relsat

SharpCDCL Cachet SharpSAT

countAntom Ganak

Knowledge Compilation

c2d d4 dShaarp

Minic2d CNF2EADT

CNF2OBDD BuDDy Cudd

Experiment Design

- 17 Solvers (15 exact, 2 approximate)
  - 7 DPLL Solvers (Single query)
  - 8 Knowledge Compilers (3 d-DNNF, 3 BDD, 2 other)
- 15 Subject Systems
  - 6 Evolutions
  - 373 Feature Models

Systems	#Vers.	#Feat.	#Const.
BerkeleyDB	1	76	20
axTLS	1	96	14
uClibc	1	313	56
uClinux-base	1	380	3,455
Automotive04	50	127–531	0–623
Automotive03	5	149–588	0–1,184
BusyBox	37	439–631	463–691
FinancialServices	10	557–771	1,001–1,148
Embtoolkit	1	1,179	323
CDL	116	1,178–1,408	816–956
uClinux-dist.	1	1,580	197
Automotive05	136	246–1,663	0-11,632
Automotive01	1	2,513	2,833
Linux	1	6,467	3,545
Automotive02	4	14,010–18,616	666–1,369

,

Experiment Design

- 17 Solvers (15 exact, 2 approximate)
  - 7 DPLL Solvers (Single query)
  - 8 Knowledge Compilers (3 d-DNNF, 3 BDD, 2 other)
- 15 Subject Systems
  - 6 Evolutions
  - 373 Feature Models
- Objectives
  - Scalability #SAT solvers
  - Recommendations: Solvers/Techniques
  - Future work

Insights

...

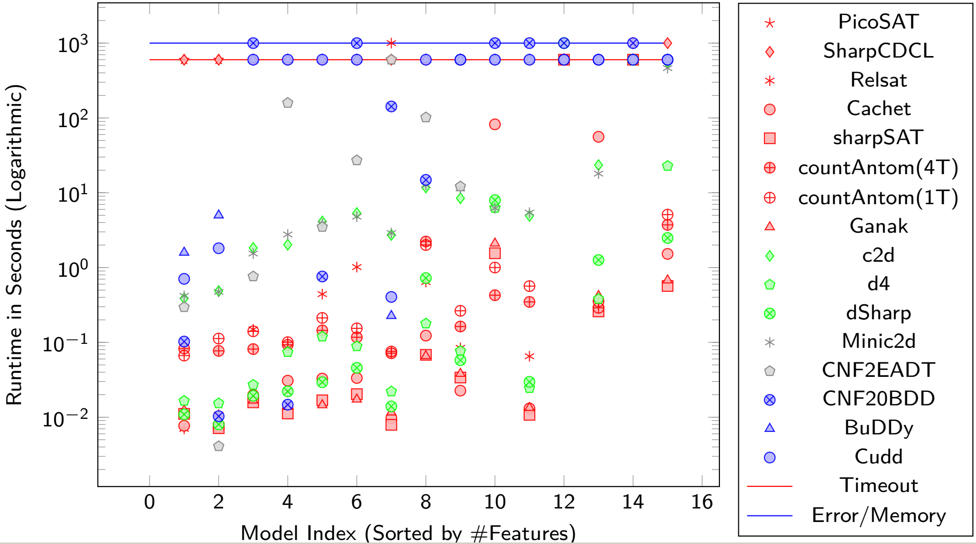
Limits

...

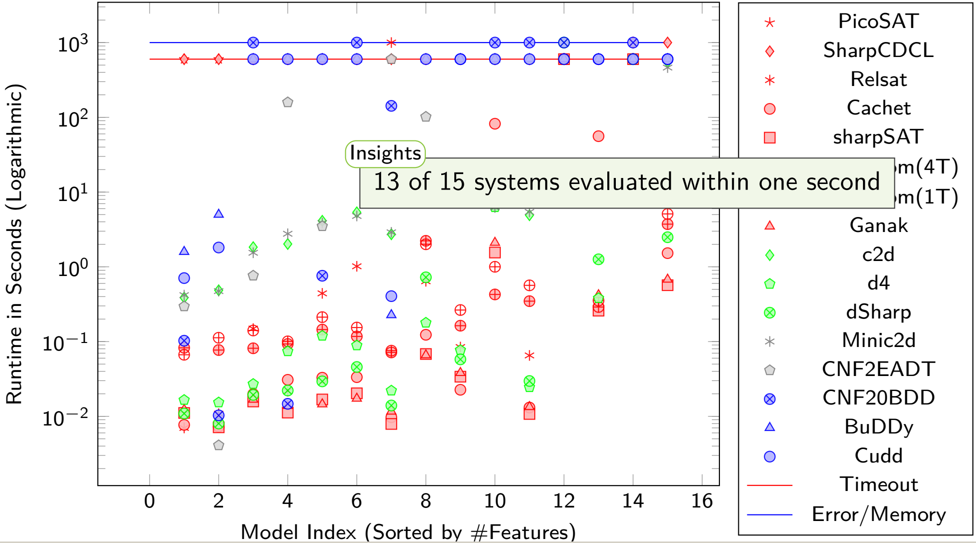
Future Work

...

Scalability #SAT



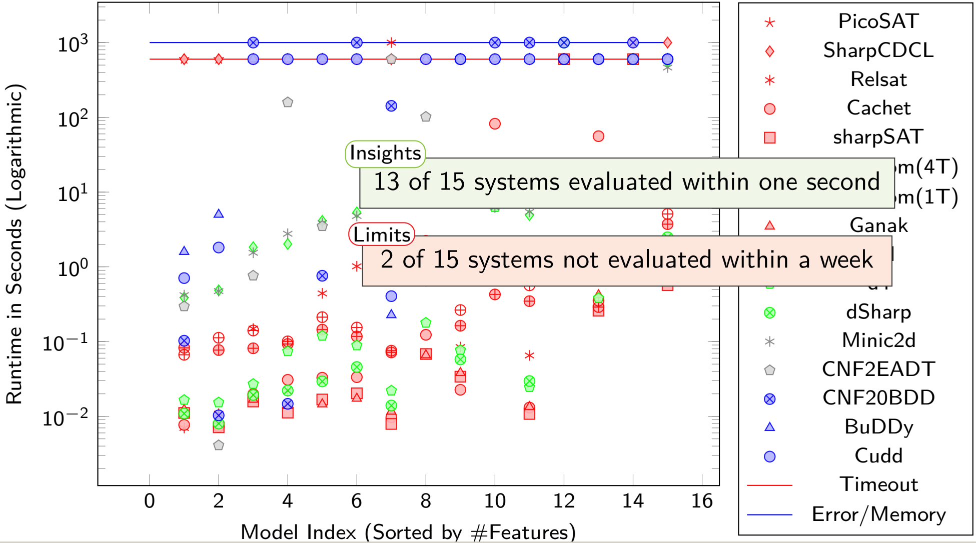
Scalability #SAT



Insights

13 of 15 systems evaluated within one second

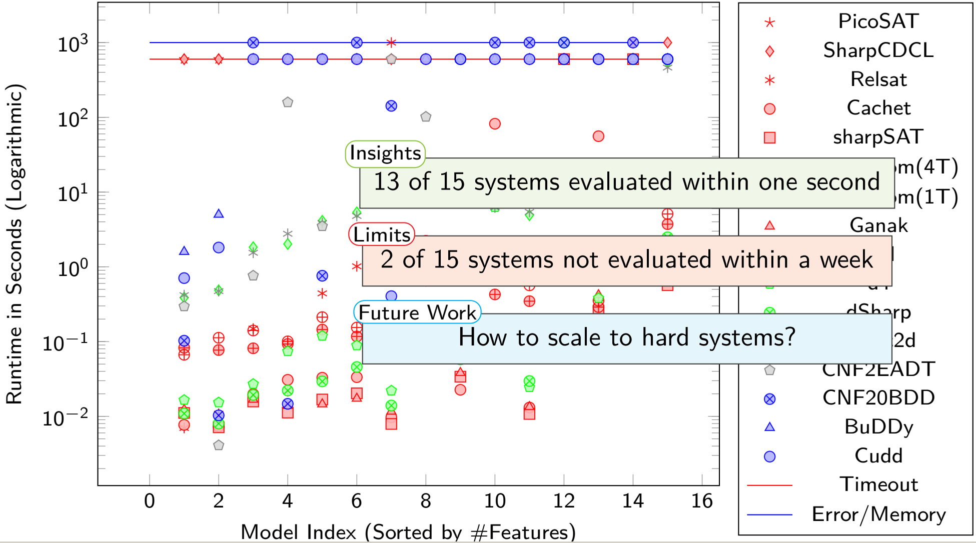
Scalability #SAT



Insights
13 of 15 systems evaluated within one second

Limits
2 of 15 systems not evaluated within a week

Scalability #SAT



Insights

13 of 15 systems evaluated within one second

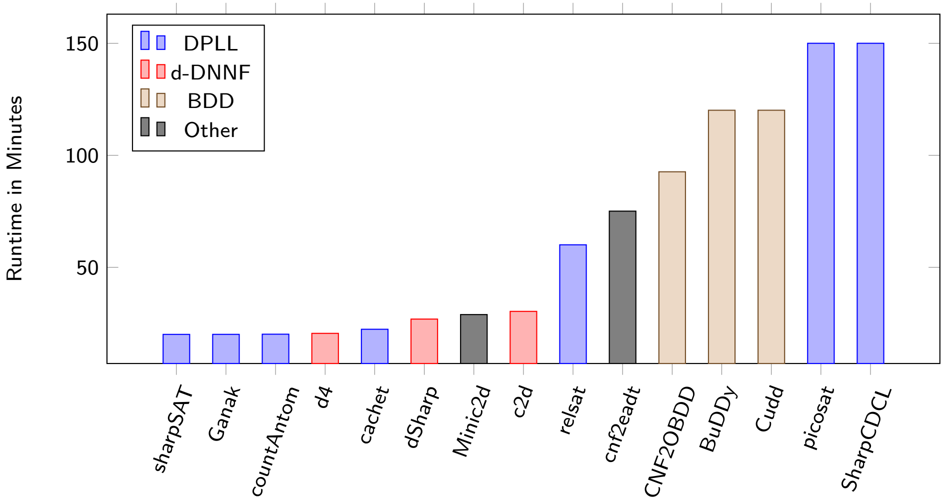
Limits

2 of 15 systems not evaluated within a week

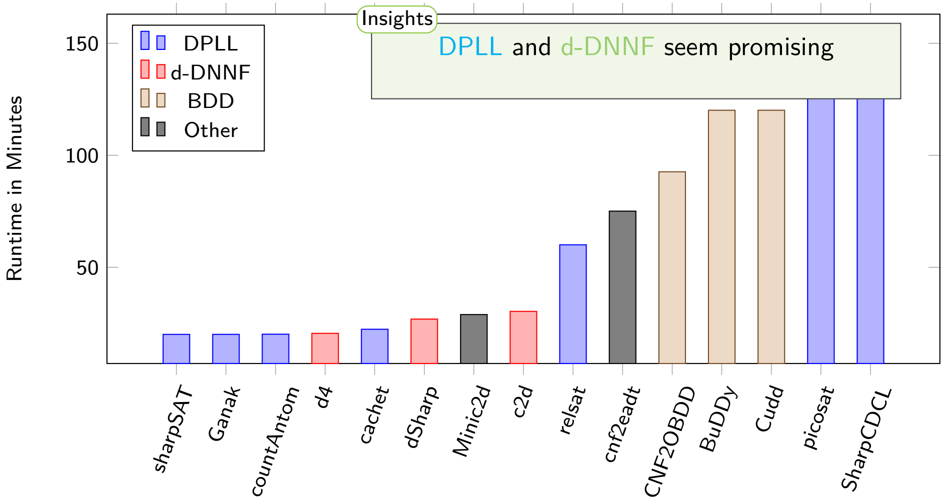
Future Work

How to scale to hard systems?

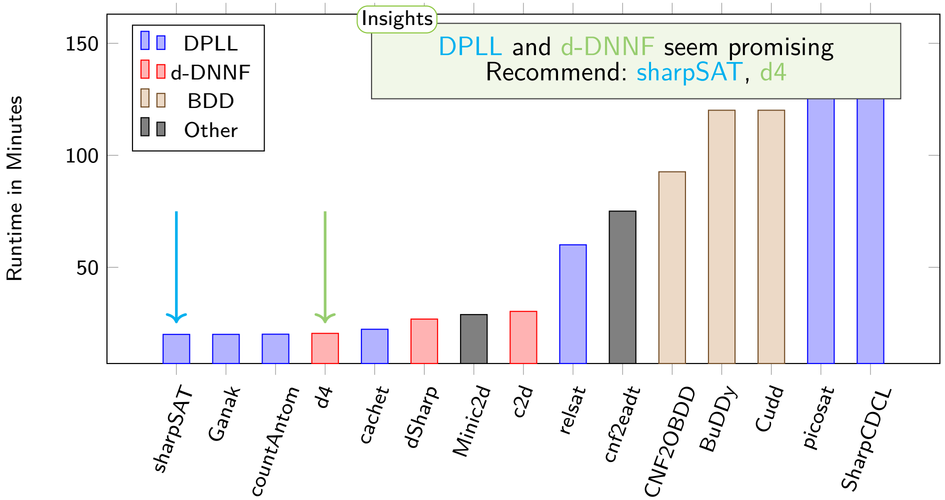
Performance Solvers



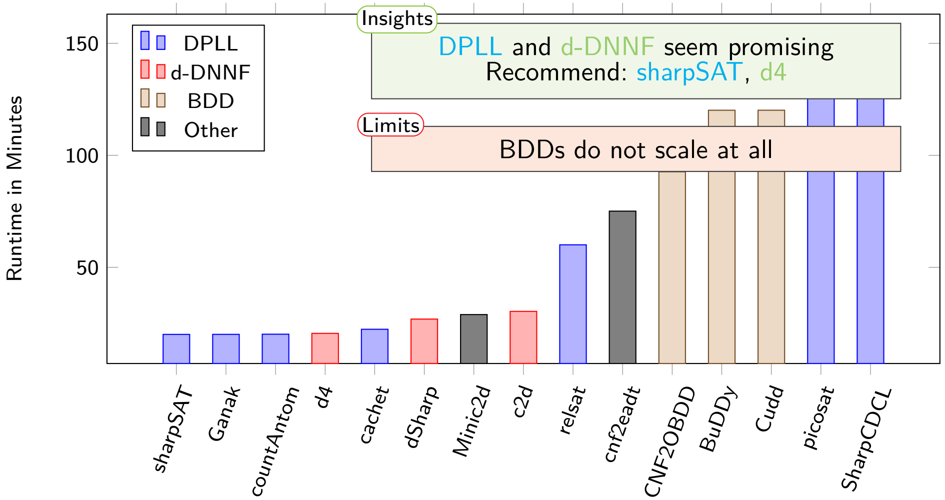
Performance Solvers



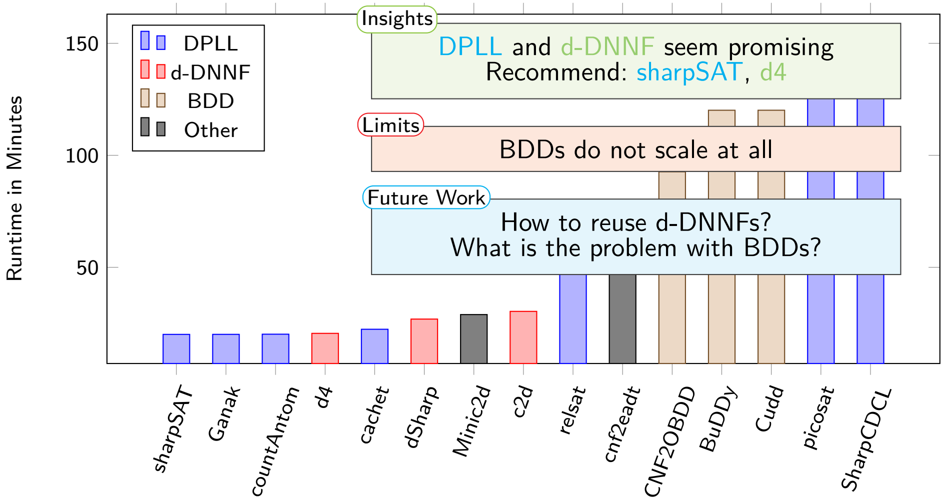
Performance Solvers



Performance Solvers



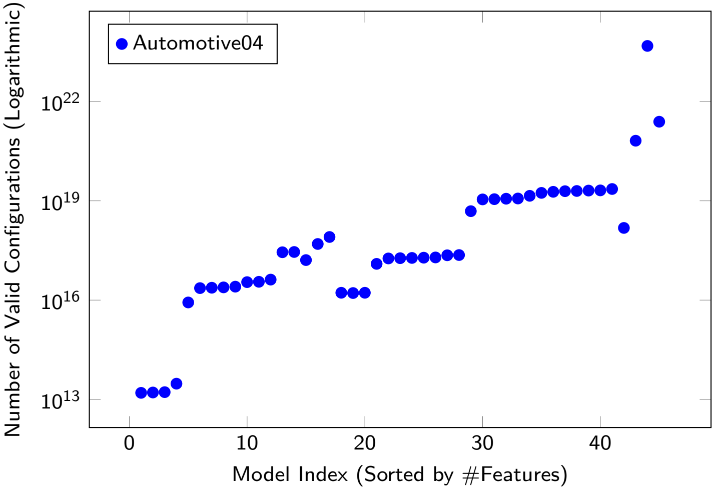
Performance Solvers



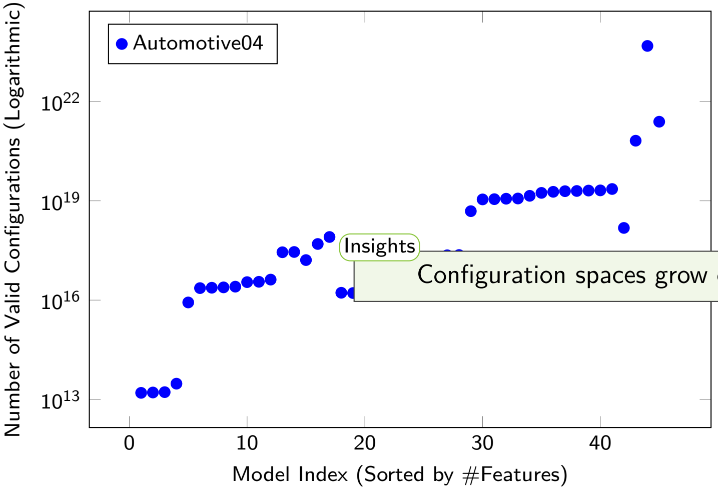
How many configurations do we have in practice?

Subject Systems	Features	Number of Valid Configurations
BerkeleyDB	76	$4.1 \cdot 10^{9}$
axTLS	96	$8.3 \cdot 10^{11}$
uClibc	313	$1.7 \cdot 10^{40}$
uClinux-base	380	$2.6 \cdot 10^{22}$
Automotive04	531	$2.5 \cdot 10^{21}$
Automotive03	588	$2.5 \cdot 10^{31}$
BusyBox	631	$2.1 \cdot 10^{201}$
FinancialServices	771	$9.7 \cdot 10^{13}$
Embtoolkit	1,179	$5.1 \cdot 10^{96}$
CDL (116 Models)	1,178–1,408	$2.6 \cdot 10^{118} - 3.0 \cdot 10^{136}$
uClinux-distribution	1,580	$4.1 \cdot 10^{409}$
Automotive05	1,663	unknown
Automotive01	2,513	$5.4 \cdot 10^{217}$
Linux	6,467	unknown
Automotive02	18,616	$1.7 \cdot 10^{1534}$

Evolution: Sizes



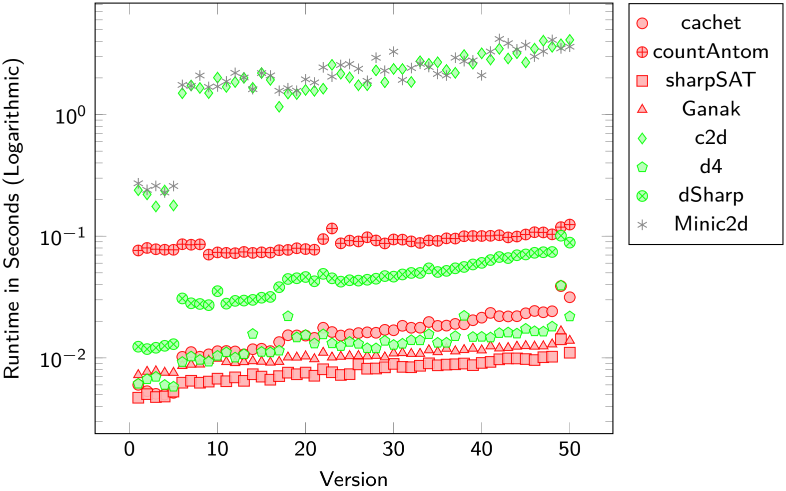
Evolution: Sizes



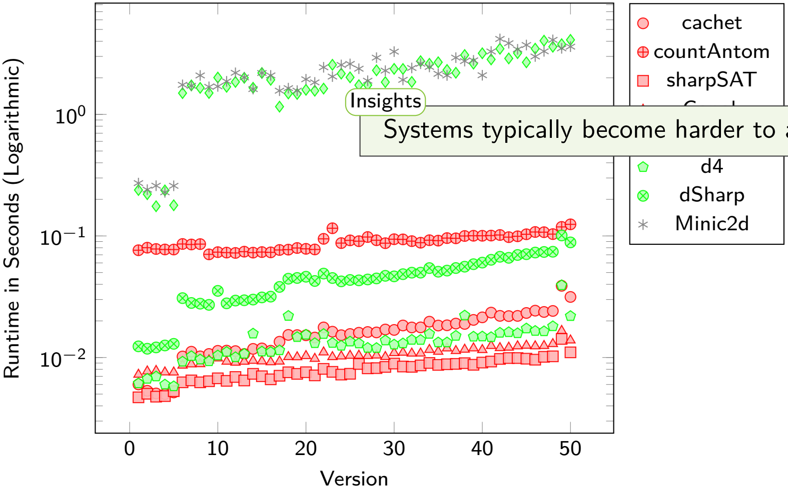
Insights

Configuration spaces grow over time

Evolution: Performance



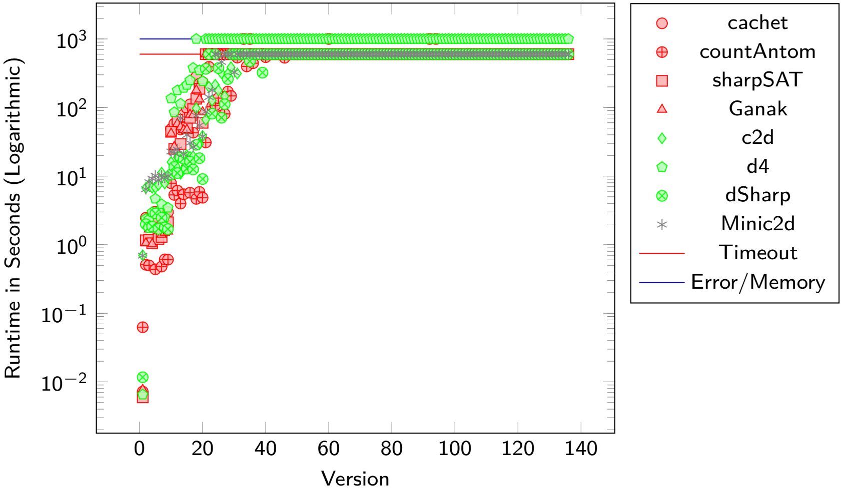
Evolution: Performance



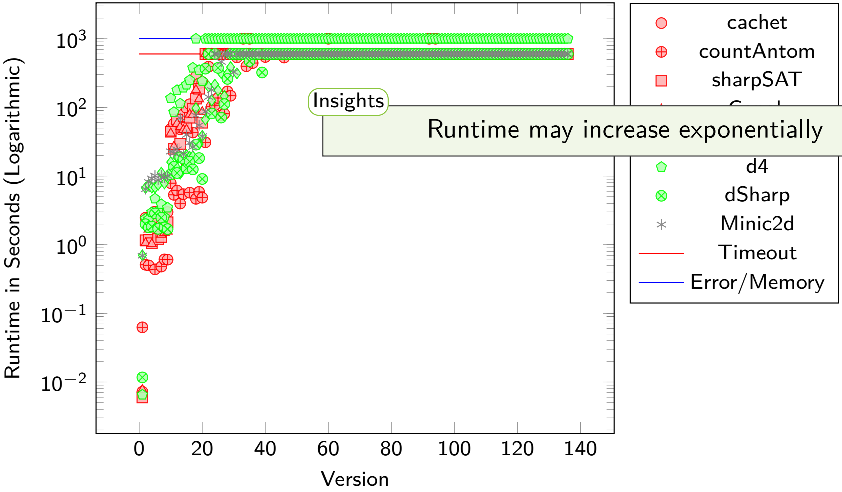
Insights

Systems typically become harder to analyze

Evolution: Performance



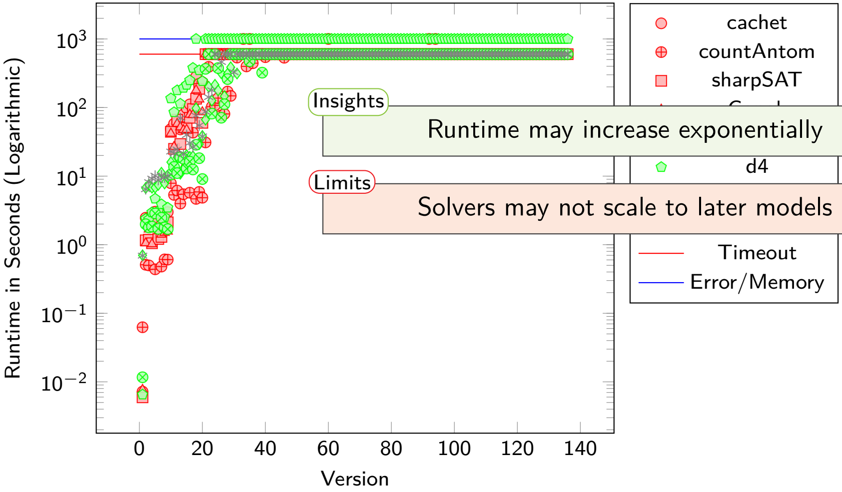
Evolution: Performance



Insights

Runtime may increase exponentially

Evolution: Performance



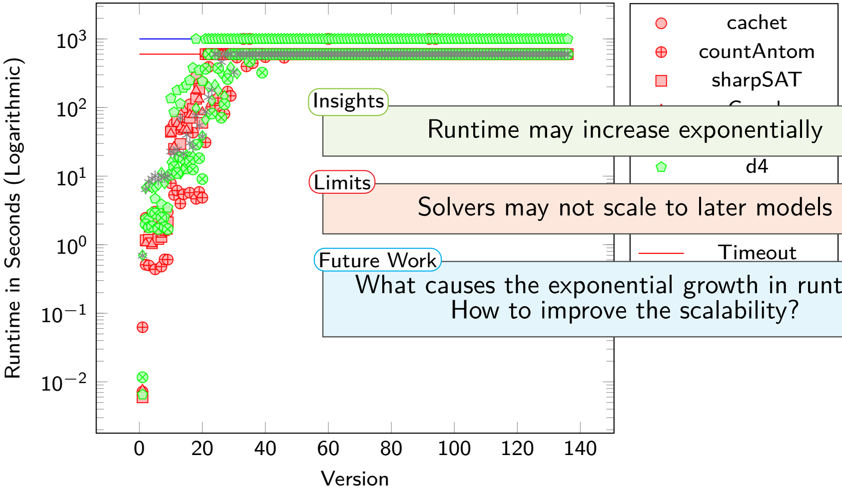
Insights

Runtime may increase exponentially

Limits

Solvers may not scale to later models

Evolution: Performance



Insights

Runtime may increase exponentially

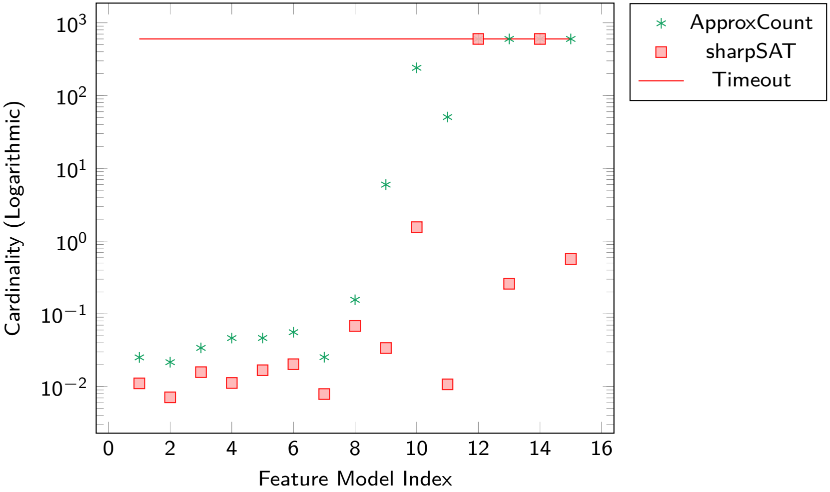
Limits

Solvers may not scale to later models

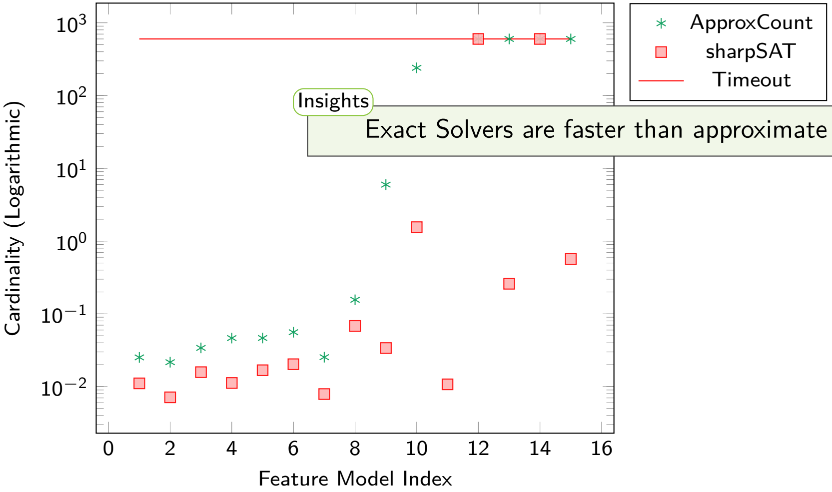
Future Work

What causes the exponential growth in runtime?
How to improve the scalability?

Approximate #SAT Solvers



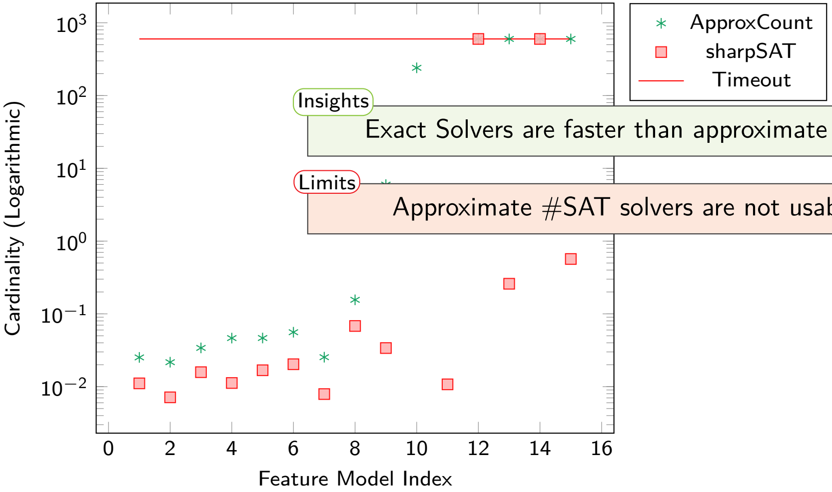
Approximate #SAT Solvers



Insights

Exact Solvers are faster than approximate ones

Approximate #SAT Solvers



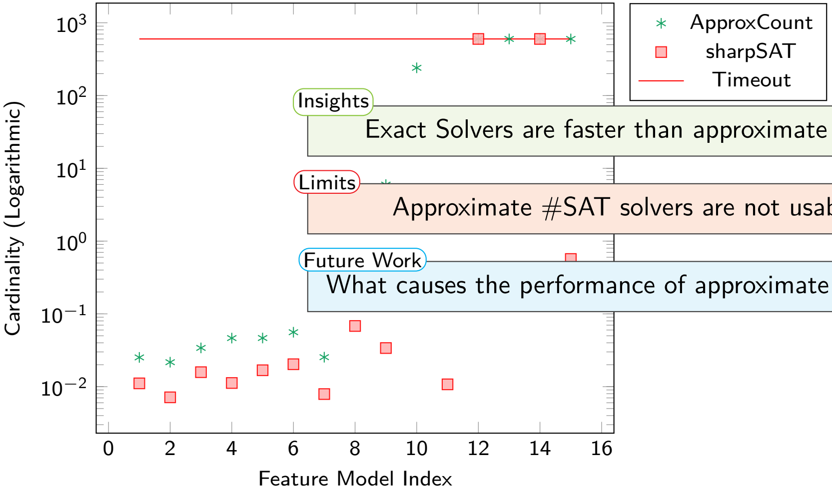
Insights

Exact Solvers are faster than approximate ones

Limits

Approximate #SAT solvers are not usable

Approximate #SAT Solvers



Insights
Exact Solvers are faster than approximate ones

Limits
Approximate #SAT solvers are not usable

Future Work
What causes the performance of approximate solvers?

Evaluating State-of-the-Art #SAT Solvers on Industrial Configuration Spaces

Insights

Majority of systems can be analyzed with #SAT solvers
DPLL: ✓d-DNNF: ✓

Evaluating State-of-the-Art #SAT Solvers on Industrial Configuration Spaces

Insights

Majority of systems can be analyzed with #SAT solvers
DPLL: ✓ d-DNNF: ✓ BDD: ✗

Evaluating State-of-the-Art #SAT Solvers on Industrial Configuration Spaces

Insights

Majority of systems can be analyzed with #SAT solvers
DPLL: ✓d-DNNF: ✓BDD: ✗

Evaluating State-of-the-Art #SAT Solvers on Industrial Configuration Spaces

Insights

Majority of systems can be analyzed with #SAT solvers
DPLL: ✓ d-DNNF: ✓ BDD: ✗

Limits

Two systems could not be analyzed
Solvers may not scale to later models in evolution

Evaluating State-of-the-Art #SAT Solvers on Industrial Configuration Spaces

Insights

Majority of systems can be analyzed with #SAT solvers
DPLL: ✓d-DNNF: ✓BDD: ✗

Limits

Two systems could not be analyzed
Solvers may not scale to later models in evolution

Future Work

How to scale to hard systems?
What causes exponential growth in runtime during evolution?

Start Backup Slides

Nice, a question that can be somewhat answered with backup :)

Correlation Performances

Tabelle: Spearman Correlation between Structural Metrics and Runtime of #SAT Solvers

Metric	Coefficient Fastest	Coefficient Range
Number of Literals	0.89 (very strong)	0.82 (very strong)–0.89 (very strong)
Number of Clauses	0.87 (very strong)	0.80 (very strong)–0.87 (very strong)
Number of Features	0.85 (very strong)	0.73 (strong) –0.94 (very strong)
Number of Leaf Features	0.82 (very strong)	0.68 (strong) –0.92 (very strong)
Number of Constraints	0.81 (very strong)	0.67 (strong)–0.84 (very strong)
Cyclomatic Complexity	0.77 (strong)	0.64 (strong)–0.79 (strong)
Tree Depth	0.34 (weak)	0.29 (weak)–0.39 (weak)
Connectivity Density	0.20 (weak)	0.11 (very weak)–0.57 (moderate)
Ratio of Variability	0.11 (very weak)	-0.01 (very weak)–0.27 (weak)
Number of Top Features	0.06 (very weak)	-0.01 (very weak)–0.17 (very weak)
Flexibility of Configuration	0.07 (very weak)	-0.01 (very weak)–0.18 (very weak)