The SAT'03 evaluation of QBF solvers

Challenges in the QBF arena

Daniel Le Berre and Laurent Simon and Armando Tacchella

Why evaluating QBF solvers?

- Several QBF solvers available, becoming more and more complex (and efficient)
- QBF problems available (QBFLIB) from various area: planning, formal verification, knowledge reasoning, etc.
- No global comparison yet (local comparison on papers)
- A good way to attract people to QBF?

The evaluation at a glance

- 11 solvers (10 complete, 1 incomplete)
- 442 benchmarks submitted
- 1278 benchmarks from www.qbflib.org
- 8 Linux boxes (PIV with 1GB RAM)
- SatEx framework
- about 1600 hours of CPU time (>60 days)

At the end of the day(s)...

- Relative strength of solvers and state-of-the-art in solving techniques
- Classification of benchmarks according to the state-of-the-art solvers
- First (ever) quantitative assessment of QBF potential for applications
- Challenges for QBF researchers

Benchmarks overview

- A total of 1720 benchmarks into 134 series have been run during the evaluation.
- Among these, 856 were generated according to some probabilistic model and 864 were real-world instances.
- 442 instances were submitted to the evaluation, and 1278 others were selected from QBFLIB.

Benchmarks: format

- Prenex Clausal Normal Form
- Any QBF can be translated into PCNF
- QDimacs format
  - Backward compatible with Dimacs
  - 2 years old now (QBF'01, Siena)
  - Rintanen's format still widely accepted

Submitted Benchmarks

Scholl/Becker 64 benchmarks (23 proved true, 16 proved false and 25 still unsolved) in 8 series, encode equivalence checking for partial implementations problems.

Guoqiang Pan 378 benchmarks (50% true, 50% false) in 18 series, encode the satisfiability of modal K formulas into QBF. The original benchmarks have been proposed during the TANCS'98 comparison of theorem provers for modal logics.

Series		Satisfi ability		Formulae Data				
name	count	T%	F%	variables	clauses	sets	exist	forall
k_branch_n	21	100%	0%	5679.05	28927.43	27.00	5614.67	64.38
k_branch_p	21	0%	100%	5679.05	28928.43	27.00	5614.67	64.38
k_d4_n	21	100%	0%	1000.90	3542.38	30.90	952.00	48.90
k_d4_p	21	0%	0%	694.62	2050.33	30.90	656.71	37.90
k_dum_n	21	100%	0%	575.86	1479.67	32.90	546.05	29.81
k_dum_p	21	0%	100%	486.52	1266.81	26.14	462.38	24.14
k_grz_n	21	100%	0%	539.76	1973.00	17.00	520.57	19.19
k_grz_p	21	0%	100%	520.10	1908.38	17.00	500.05	20.05
k_lin_n	21	100%	0%	2276.19	27013.90	8.81	2261.90	14.29
k_lin_p	21	0%	100%	586.24	3220.10	7.00	576.00	10.24
k_path_n	21	100%	0%	802.29	2229.76	27.00	765.29	37.00
k_path_p	21	0%	100%	736.52	2047.67	25.00	702.57	33.95
k_ph_n	21	100%	0%	3211.67	70922.62	4.86	3203.38	8.29
k_ph_p	21	0%	100%	3517.00	79384.52	4.90	3508.62	8.38
k_poly_n	21	100%	0%	911.14	2055.95	72.05	842.10	69.05
k_poly_p	21	0%	100%	909.86	2053.05	71.95	840.90	68.95
k_t4p_n	21	100%	0%	1603.76	5029.76	55.00	1528.81	74.95
k_t4p_p	21	0%	100%	908.81	2815.00	35.00	863.81	45.00
C432	8	50%	50%	594.50	1525.00	6.50	588.00	6.50
C499	8	50%	50%	881.50	2528.50	7.00	875.00	6.50
C5315	8	50%	50%	5509.00	14818.25	10.00	5459.50	49.00
C6288	8	50%	50%	4702.25	13798.75	6.00	4651.25	50.50
C880	8	50%	50%	1009.00	2571.50	7.50	994.50	14.00
comp	8	50%	50%	299.75	815.25	4.00	297.25	2.50
term1	8	100%	0%	1090.75	3760.25	8.00	1084.75	5.75
z4ml	8	100%	0%	63.75	193.00	3.50	62.25	1.50

Table 1: Data of submitted benchmarks.

Benchmarks from QBFLIB

Ayari 72 benchmarks (35 true, 37 false) in 8 series. A family of circuit-related problems.

Castellini 169 benchmarks (57 true, 112 false) in 5 series. Various QBF-based encodings of the bomb-in-the-toilet planning problem.

Letz 14 benchmarks (5 true, 9 false). tree problems.

Benchmarks from QBFLIB (2)

Narizzano/Random 836 benchmarks in 80 series. Several QBFs randomly generated with model A proposed by Gent and Walsh.

Narizzano/Robot 120 benchmarks in 4 series. QBF-based encoding of the "robot-on-the-grid" planning problem.

Rintanen 67 benchmarks (44 false, 23 false) in 9 series consisting of planning, hand made and random problems.

Series		Satisfi ability		Formulae Data				
name	count	T%	F%	variables	clauses	sets	exist	forall
Adder-s1	8	100%	0%	2775.00	3742.50	4.00	1216.50	667.50
Adder-s2	8	100%	0%	9071.00	9927.50	6.00	7623.50	556.50
Adder-u1	8	0%	100%	2791.00	3750.50	3.00	1556.50	343.50
Adder-u2	8	0%	100%	9071.00	9919.50	7.00	7725.50	454.50
DFlipFlop	10	0%	100%	62919.50	82655.30	3.00	62058.50	37.50
MutexP	7	100%	0%	8890.43	4172.71	2.00	3078.86	145.14
SzymanskiP	12	100%	0%	82461.75	90270.75	3.00	72526.08	1786.67
VonNeumann	11	0%	100%	441489.00	643479.00	3.00	438327.00	240.00
ToiletA-s	21	100%	0%	304.00	4341.81	3.00	297.43	6.57
ToiletA	56	0%	100%	219.25	5246.98	3.00	211.61	7.64
ToiletC-s	29	100%	0%	406.03	2997.93	3.00	402.97	3.07
ToiletC	56	0%	100%	214.79	986.48	3.00	211.61	3.18
ToiletG	7	100%	0%	49.57	204.29	3.00	46.43	3.14
Tree	14	35%	65%	52.86	34.14	39.29	26.43	26.43
Robots_D2	30	70%	0%	6422.00	21359.00	2.00	5172.50	27.50
Robots_D3	30	57%	30%	6422.00	21360.00	2.00	5172.50	27.50
Robots_D4	30	47%	53%	6422.00	21361.00	2.00	5172.50	27.50
Robots_D5	30	47%	53%	6422.00	21362.00	2.00	5172.50	27.50
Blocks-s	4	100%	0%	435.75	5098.00	3.00	431.00	4.75
Blocks	9	0%	100%	507.11	7570.44	3.00	501.78	5.33
Chain	12	100%	0%	1997.50	11173.75	3.00	1980.00	17.50
Impl	10	100%	0%	46.00	90.00	23.00	35.00	11.00
Logn	4	0%	100%	1318.00	61599.00	2.00	1317.50	0.50
R3cnf-s	13	100%	0%	150.00	383.08	5.15	135.00	15.00
R3cnf	7	0%	100%	150.00	381.43	4.71	135.00	15.00
Toilet-s	5	100%	0%	754.80	3434.60	3.00	751.80	3.00
Toilet	3	0%	100%	240.33	869.00	3.00	238.00	2.33

The solvers

OPENQBF by Gilles Audemard, Daniel Le Berre and Olivier Roussel.

QBFL by Florian Letombe.

QSAT by Jussi Rintanen.

QSOLVE/SSOLVE by Rainer Feldmann and Stefan Schamberger.

QUAFFLE by Lintao Zhang and Sharad Malik.

QUANTOR by Armin Biere.

The solvers (continued)

QUBE-BJ by Massimo Narizzano, Armando Tacchella and Enrico Giunchiglia.

QUBE-REL by Massimo Narizzano, Armando Tacchella and Enrico Giunchiglia.

SEMPROP from Reinhold Letz.

WALKQSAT by Andrew G. D. Rowley, Ian Gent, Holger Hoos and Kevin Smyth.

WATCHEDCSBJ Andrew G. D. Rowley and Ian Gent.

Solver features

Data structure WL stands for “watched literals”, a lazy data structure proposed for the Chaff architecture. CB stands for “counter based”, which is the traditional (non lazy) representation of the formula.

Heuristic QO stands for “Quantifier Order”, BO and MO are the Böhm’s and MOMS heuristics adapted to QBF, SRC stands for “one in the smallest reduced clause”.

Solver features (2)

Pure Literal rule extended for the QBF case: if an existential literal l is pure, propagate l, if a universal literal l is pure, propagate $\neg l$.

Trivial Truth A QBF is true if its deleting all universal literals from the clauses yields a satisfiable formula.

Trivial Falsity Π denotes that trivial falsity is detected through non tautological universal clauses, Σ denotes that the trivial falsity is detected through the inconsistency of the CNF made of existential clauses of the matrix.

Solver features (3)

No-good Learning the solver learns conflict-generated clauses (lemmas) during the search.

Good Learning the solver learns solutions (models) in the form of conjunctions of literals.

Unfolding universal quantifiers are unfolded using the rule $\forall x f(x) = f(true) \wedge f(false)$.

Inversion the solver applies the quantifier inversion rules. $\exists x \forall y f(x, y) \rightarrow \forall y \exists x f(x, y)$

Chronological/Lookahead solvers

Chronological BT	OPENQBF	QSAT	SSOLVE
data structure	CB	CB	CB
heuristic	QO	MO	BO
pure literals	yes	yes	yes
trivial truth	no	yes	yes
trivial falsity	Π	Π, Σ	Π, Σ
unfolding	no	partial	no
inversion	no	yes	yes

Incomplete solver

Incomplete	WALKQSAT
data structure	?
heuristic	Walksat
pure literals	yes
trivial truth	yes*
trivial false	no
conflict BJ	yes
solution BJ	yes

Backjumping solvers

NCB	QBFL	QUAFFLE	Q/BJ	Q/REL	SEMP	WCSBJ
data structure	CB+WL	CB	CB	CB	CB	WL
heuristic	BO	VS	BO	BO	SRC	?
pure literals	yes	no	yes	yes	yes	no
trivial truth	yes	no	yes	yes	yes*	yes
trivial false	Π, Σ	Π, Σ	Π	Π	Π	Π
nogood	yes*	yes	no	yes	yes	no
good	no	yes	no	yes	yes	no
conflict BJ	yes*	yes	yes	yes	yes	yes
solution BJ	no	yes	yes	yes	yes	yes

A remarquable solver: quantor

- Quote from Armin Biere:
  quantor: elimination of innermost universal quantifiers by copying. This is a very preliminary version, which was put together in less than 2 person weeks. A lot of optimizations are missing.
- It is the only solver from the evaluation not based on AND/OR search!

solver	Overall		Random			Non random		
	#series	#benchs	#series	#benchs	Remark	#series	#benchs	Remark
openqbf	110	725	65	404		45	321	
qbfl	118	709	75	415		43	294	
qsat	130	1152	82	701		48	451	SOTAC(22)
quaffle	99	812	60	389		39	423	SOTAC(9)
quantor	64	456	19	115		45	341	
qubebj	131	1383	82	842	SOTAC(4)	49	541	SOTAC(8)
quberel	133	1324	82	803		51	521	SOTAC(12)
semprop	133	1389	82	810		51	579	SOTAC(58)
ssolve	131	1293	82	840	SOTAC(5)	49	453	
walkqsat	117	823	80	600		37	223	
watchedcbsj	125	1004	80	602		45	402	SOTAC(2)

Table 3: Overall results of the evaluation

Strength of solvers

- Strong solvers (>130 series): QSAT,QUBE-BJ,QUBE-REL,SEMPROP,SSOLVE
- Medium solvers: OPENQBF,QBFL, QUAFFLE, WALKQSAT,WATCHEDCSBJ
- State Of The Art Contributor: a solver being the only one to solve a problem.
- Must be noted:
  - Quantor failed on random problems.
  - Quaffle failed on some non random problems because of input problems.

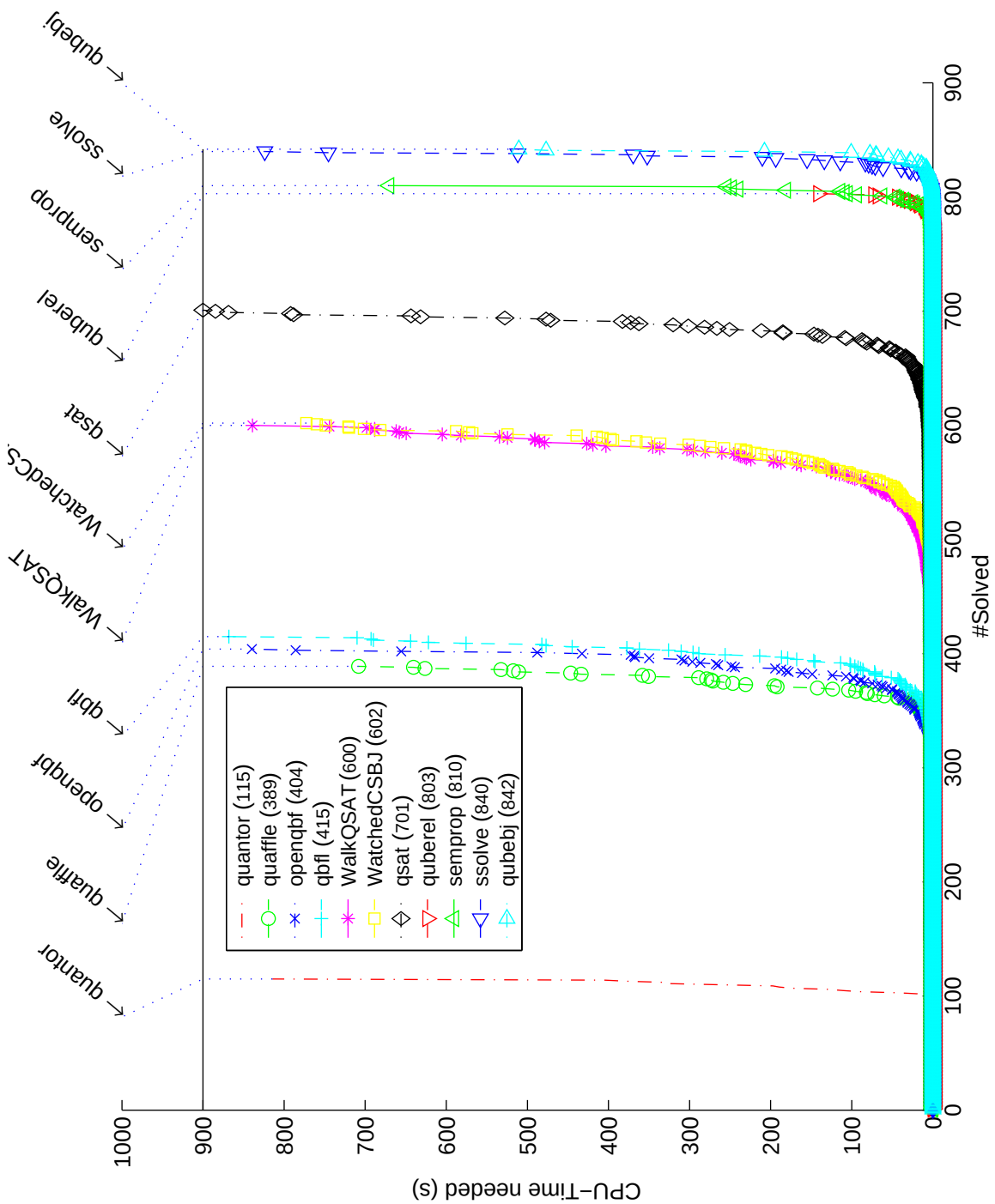


Figure 1: Number of instances solved vs. CPU time for random benchmarks

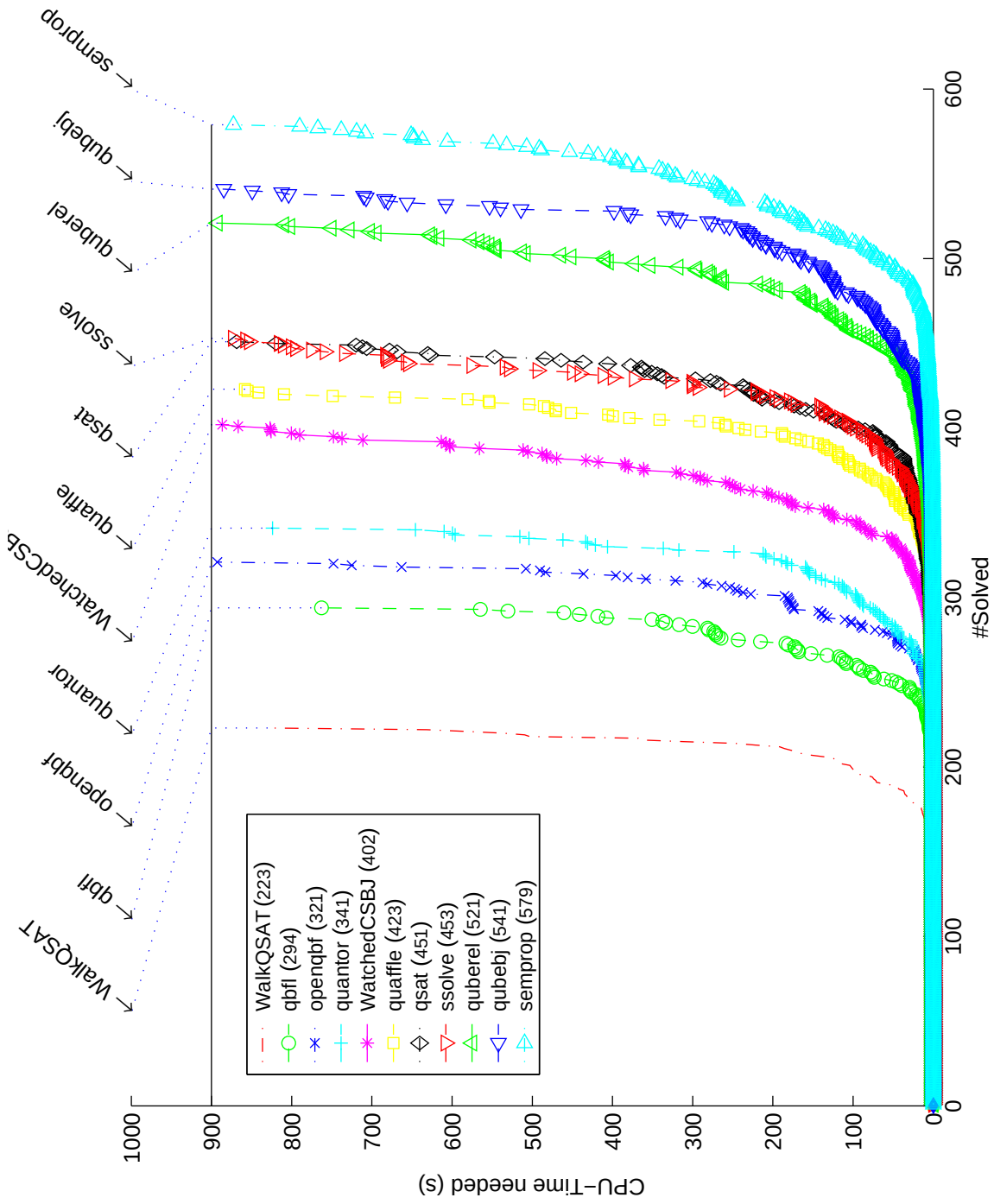
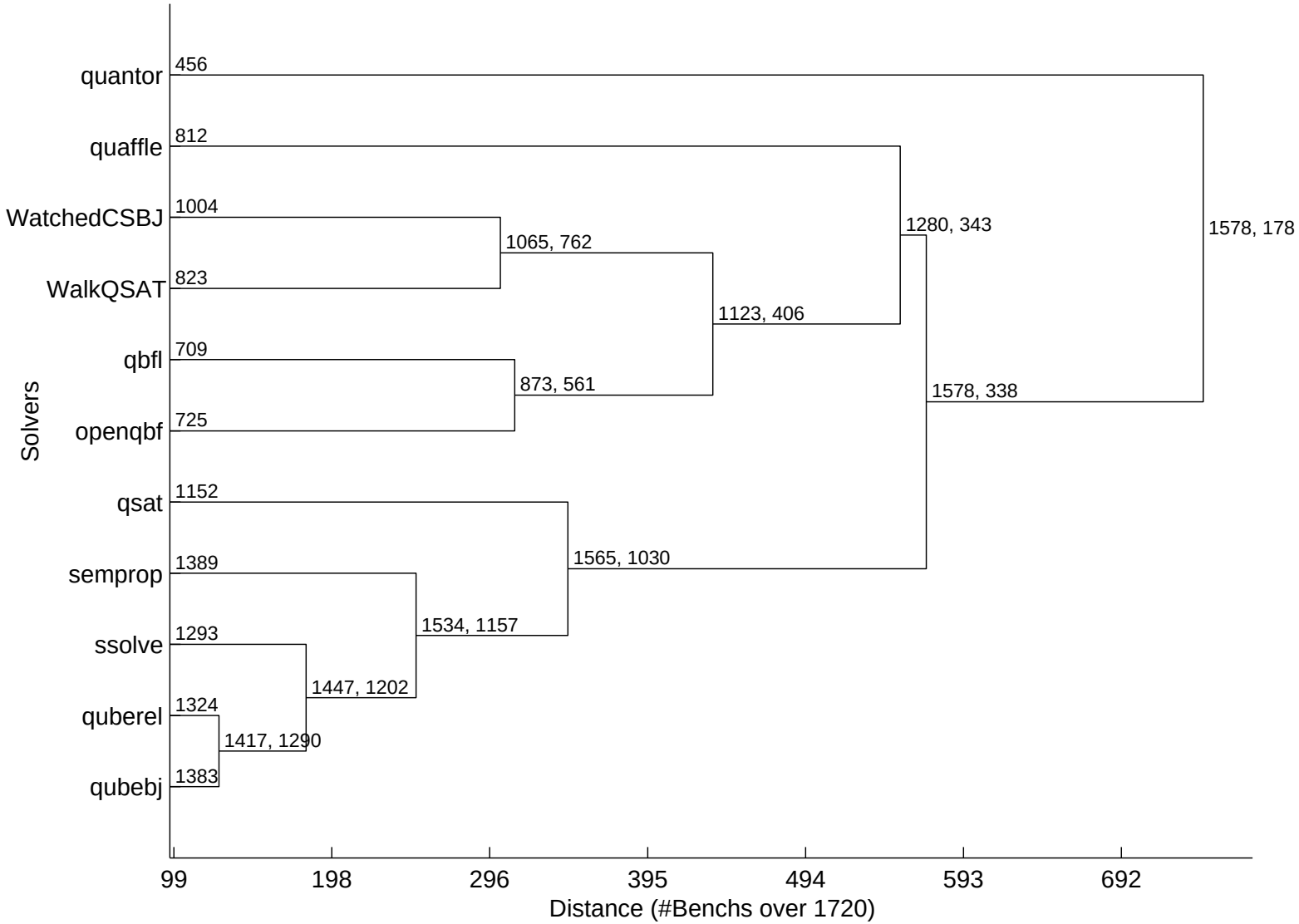


Figure 2: Number of instances solved vs. CPU time for non-random benchmarks


quantor
quaffle
WatchedCSBJ
WalkQSAT
qbfl
openqbf
qsat
semprop
ssolve
quberel
qubebj
Solvers
456
812
1004
823
709
725
1152
1389
1293
1324
1383
1065, 762
1123, 406
873, 561
1417, 1290
1447, 1202
1534, 1157
1565, 1030
1280, 343
1578, 338
1578, 178
99
198
296
395
494
593
692
Distance (#Benchs over 1720)

Benchmarks difficulty

- Easy: solved by all solvers (178)
- Hard: solved by no solver (142)
- Medium: solved by all “strong” solvers (862 = 1030-178)
- Smallest unsolved problem: SchollBecker (568v,1439c, 3 alt.)
- Biggest solved problem: Ayari (1.3Mv, 1.9Mc,3 alt.)

Benchmarks difficulty

Benchmark	#benchs	#Easy	#Medium	#Hard
Scholl/Becker	64	5	14	25
Guoqiang Pan	378	19	61	72
Ayari	72	0	15	25
Castellini	169	115	30	0
Letz	14	2	8	0
Narizzano/Random	836	16	663	7
Narizzano/Robot	120	0	36	12
Rintanen	67	21	25	0

Conclusion (personnal point of view)

- pure literals detection is important (see Quaffle)
  - unlike SAT, may trigger unit propagation
  - strong reason against watched literals
- strong solver have either a good lookahead (ssolve, qsat) or non chronological backtracking (both on solution and conflict)
- trivial truth looks important (all strong solvers implement it)
- is learning important?

Challenges 1/3 for benchmarks

Challenge 1 *Solve the 142 hard QBF benchmarks remained unsolved during the evaluation.*

Challenge 2 *Random generation model that respect the polynomial hierarchy.*

Challenge 3 *Proving on the field that QBF-based reasoning can beat SAT-based reasoning.*

Challenges 2/3: new techniques

Challenge 4 *Solution learning as enabling technology for new simplification techniques/heuristics.*

Challenge 5 *Conceive and experiment new simplification and intelligent backtracking techniques.*

Challenge 6 *Conceive and experiment new effective heuristics for QBF.*

Challenge 7 *Investigate the combination of technologies such as inversion of quantifiers, random sampling, partial expansion together with simplification and backtracking techniques.*

Challenges 3/3: new frontiers

Challenge 8 *Build an efficient QBF solver not based on AND/OR search.*

Challenge 9 *Propose new techniques for randomized/incomplete QBF solvers.*

Challenge 10 *Build special purpose solvers, i.e., solvers tuned specifically for k-QBFs with $k \in \{2, 3\}$*