XCSP³ Competition 2023 – Results –

https://www.cril.univ-artois.fr/XCSP23/

Gilles Audemard Christophe Lecoutre Emmanuel Lonca

CP'23

August 30, 2023



In a nutshell, the main ingredients of the complete tool chain we propose for handling combinatorial constrained problems are:

- PyCSP³: a Python library for modelling constrained problems see https://www.pycsp.org/
- XCSP³: an intermediate format used to represent problem instances while preserving structure of models see https://www.xcsp.org/

Why should you try $PyCSP^3/XCSP^3$?

- Mainstream Technologies Inside
- HiFi Compilation preserving the structure of models (in XCSP³)
- Quite compact and readable models
 - easy handling of data (from JSON): "one line can suffice"

nPlanes, times, costs = data

- natural expressions of constraints (tables, automatas, ...)
- Educational interest (but can be useful for industry too)
 - 25 Jupyter notebooks for introducing 25 popular constraints
 - 34 Jupyter notebooks for gently introducing, step by step, models of classical combinatorial problems
- Stable/Mature Technology: "any integer" problem can be modelled
- Many resources
 - parsers: (on GitHub): C++, Java, Rust, and (forthcoming) Python
 - solution checker
 - series of instances
 - forthcoming repository with several hundreds of models (and data)

Outline

1 2023 Competition

2 2023 Results

Purpose of Competitions

The goal of a competition is to:

- evaluate solvers in the same conditions
- help collecting publicly available benchmarks and data (results, traces, ...)
- help the community identify good ideas and strange results: the goal is to raise questions and get new ideas!

Competitions should not be misunderstood:

- The results are not an absolute truth: they depend on the benchmark selection, experimental conditions, ...
- A competition is not limited to a ranking: rankings are just an over-simplified view, but still relevant to motivate authors
- Competitions must be driven by the community: benchmark submission/selection advices, suggestions for improvements, ...

Perimeter of Constraints (mainly, XCSP³-core)

For the standard tracks:

- intension, extension
- regular, mdd
- allDifferent, allEqual, ordered, lex, precedence
- sum, count, nValues, cardinality
- maximum, minimum, element, channel
- noOverlap, cumulative, binPacking, knapsack
- circuit, instantiation
- slide

For the Mini-solver tracks:

- intension, extension
- allDifferent
- sum
- element

Tracks for the 2023 $\rm XCSP^3$ Competition

There are 4 Standard tracks and 2 Mini-solver tracks.

Problem	Goal	Exploration	Timeout
CSP	one solution	sequential	40 minutes
COP	best solution	sequential	40 minutes
Fast COP	best solution	sequential	4 minutes
Parallel COP	best solution	parallel	40 minutes

Table: Standard Tracks.

Problem	Goal	Exploration	Timeout
Mini CSP	one solution	sequential	40 minutes
Mini COP	best solution	sequential	40 minutes

Table: Mini-solver Tracks.

Computer Infrastructure



- the cluster was provided by CRIL and is composed of nodes with two quad-cores (Intel Xeon CPU E5-2637 v4 @ 3.50GHz, each equipped with 64 GiB RAM).
- Hyperthreading was disabled.
- Each solver was allocated a CPU and 64 GiB of RAM, independently from the tracks.
- Timeouts were set accordingly to the tracks through the tool runsolver:
 - sequential solvers in the fast COP track were allocated 4 min of CPU time and 12 min of Wall Clock time,
 - other sequential solvers were allocated 40 min of CPU time and 120 min of Wall Clock time,
 - parallel solvers were allocated 160 min of CPU time and 120 min of Wall Clock time.

Scoring/Ranking

The number of points won by a solver S is decided as follows:

- for CSP, this is the number of times *S* is able to solve an instance, i.e., to decide the satisfiability of an instance
- for COP, this is, roughly speaking, the number of times *S* gives the best known result, compared to its competitors. More specifically, for each instance *I*:
 - if *I* is unsatisfiable, 1 point is won by *S* if *S* indicates that the instance *I* is unsatisfiable, 0 otherwise,
 - if S provides a solution whose bound is less good than another one (found by another competiting solver), 0 point is won by S,
 - if S provides an optimal solution, while indicating that it is indeed the optimality, 1 point is won by S,
 - if *S* provides (a solution with) the best found bound among all competitors, this being possibly shared by some other solver(s), while indicating no information about optimality: 1 point (BB1) is won by *S* if no other solver proved that this bound was optimal, 0.5 (BB2) otherwise.

Selection of Benchmarks

Note that:

- the selection has been conducted by C. Lecoutre, which is why ACE is labeled *off-competition*
- three series (AircraftAssemblyLine, RuleMining and Break) have been submitted in response to the call

At the end:

- the selection of instances for the Standard tracks was composed of 200 CSP instances and 250 COP instances
- the selection of instances for the Mini-solver tracks was composed of 150 CSP instances and 155 COP instances

CSP Problems for the Main Track

CSP Problems	Global Constraints (including extension)
AnotherMagicSquare	allDifferent, sum
AntimagicSquare	allDifferent, sum, maximum, minimum
BinaryPuzzle	<pre>sum, regular, extension (*)</pre>
CalvinPuzzle	allDifferent, count, extension (*)
Coloring	
CoveringArray	allDifferent, channel, extension
Dominoes	allDifferent, extension
Fischer	
MagicSquare	allDifferent, mdd, sum
NonogramTernary	extension
NonTransitiveDice	ordered, sum, maximum
PegSolitaire	
Primes	sum
PythagoreanTriples	nValues
Slant	sum, count
Soccer	allDifferent, sum, extension
SquarePackingSuite	noOverlap, cumulative
WordDesignDNA	extension, lex, sum, mdd

COP Problems for the Main Track

COP Problems	Global Constraints (including extension)
AircraftAssemblyLine	cumulative, noOverlap
BeerJugs	extension
Benzenoide	count, extension (*), sum, lex, precedence
CarpetCutting	extension, element, cumulative, noOverlap
GBAC	cardinality, binPacking, maximum, extension
GeneralizedMKP	knapsack, sum
HCPizza	extension, sum
HSP	allDifferent, noOverlap, maximum, extension
KidneyExchange	allDifferent, element, binPacking, precedence
KMedian	allDifferent, element, ordered, minimum, sum
LargeScaleScheduling	cumulative, maximum
ProgressiveParty	element, channel, sum, allDifferent
PSP	sum, element, count
RIP	cumulative
RuleMining	allDifferent, count, extension (*)
Sonet	<pre>sum, lex, extension (*)</pre>
SRFLP	allDifferent, extension, sum
TSPTW	allDifferent, element, circuit

Outline

1 2023 Competition



Teams/Solvers (in alphabetic order)

- ACE (C. Lecoutre)
- BTD, miniBTD (M. Cherif, D. Habet, P. Jégou, H. Kanso, C. Terrioux)
- Choco (C. Prud'homme)
- CoSoCo (G. Audemard)
- Exchequer (M. Mariusz Lester)
- Fun-sCOP (T. Soh, D. Le Berre, H. Nabeshima, M. Banbara, N. Tamura)
- MiniCPBP (G. Pesant and A. Burlats)
- Mistral (E. Hebrard and M. Siala)
- Nacre (G. Glorian)
- Picat (N.-F. Zhou)
- RBO, miniRBO (M. Sami Cherif, D. Habet, C. Terrioux)
- Sat4j-CSP-PB (extension of Sat4j by T. Falque and R. Wallon)
- SeaPearl (M. Bourgeat, A. Navarro, L. Boisvert, T. Marty, L.-M. Rousseau, Q. Cappart)
- Toulbar2 (D. Allouche et al.)

Scanning the Results

Available Resources:

- Full set of XCSP^3 Instances
- PyCSP³ Models (and Data)
- Proceedings
- Dedicated website (Tables, Plots, Traces)

Late identification of some bugged situations (sorry, but these are the rules of the game)

Let us see all detailed results at https://www.cril.univ-artois.fr/XCSP23/

Scanning the Results

Available Resources:

- Full set of XCSP^3 Instances
- PyCSP³ Models (and Data)
- Proceedings
- Dedicated website (Tables, Plots, Traces)

Late identification of some bugged situations (sorry, but these are the rules of the game) $% \left(\left({{{\mathbf{x}}_{i}}} \right) \right)$

Let us see all detailed results at https://www.cril.univ-artois.fr/XCSP23/

Scanning the Results

Available Resources:

- Full set of XCSP^3 Instances
- PyCSP³ Models (and Data)
- Proceedings
- Dedicated website (Tables, Plots, Traces)

Late identification of some bugged situations (sorry, but these are the rules of the game) $% \left(\left({{{\mathbf{x}}_{i}}} \right) \right)$

Let us see all detailed results at https://www.cril.univ-artois.fr/XCSP23/

	Solver	#solved	#SAT	#UNSAT
1	Picat	120	105	15
2	Fun-sCOP-kissat	118	106	12
3	Choco	92	86	6

See https://www.cril.univ-artois.fr/XCSP23/competitions/csp/csp

COP (250 instances)

	Solver	Score	#Opt	#BB1	#BB2
1 2 3	Mistral Choco CoSoCo	106.0 92.0 88.5	40 37 35	60 48 47	12 14 13

See https://www.cril.univ-artois.fr/XCSP23/competitions/cop/cop

Without last year winner Picat neutralized on KidneyExchange (while remaining the best competitor in term of proved optima), results would have been still more tight.

	Solver	Score	#Opt	#BB1	#BB2
1	Choco	99.5	30	64	11
2	Mistral	84.0	30	50	8
3	CoSoCo	77.5	23	47	15

See https://www.cril.univ-artois.fr/XCSP23/competitions/cop/fast-cop

	Solver	Score	#Opt	#BB1	#BB2
1	Choco	219	41	178	0
2	toulbar2	27.5	11	15	3

See https://www.cril.univ-artois.fr/XCSP23/competitions/cop/parallel-cop.

Some technical problem with // toulbar2 needs to be fixed. Data might be updated with a fixed version (but ranking will remain as it is)

Mini CSP (150 instances)

	Solver	#solved	#SAT	#UNSAT
1	Exchequer	74	65	9
2	miniBTD	64	56	8
3	Nacre	50	44	6

See https://www.cril.univ-artois.fr/XCSP23/competitions/csp/mini-csp

Some technical problem with MiniCSP Seapearl needs to be fixed. Data might be updated with a fixed version (but ranking will remain as it is)

Mini COP (155 instances)

	Solver	Score	#Opt	#BB1	#BB2
1	toulbar2	77.0	12	64	2
2	Exchequer	58.5	20	38	1
3	Sat4j-both	38.5	8	30	1

See https://www.cril.univ-artois.fr/XCSP23/competitions/cop/mini-cop

Precise analysis of results is possible because **all details** are on https://www.cril.univ-artois.fr/XCSP23/

Forthcoming:

- PyCSP³, version 2.2 in October 2023
- 2024 $\mathrm{XCSP^3}$ Competition in Summer 2024