XCSP^3 Competition 2024
– Results –

Gilles Audemard Christophe Lecoutre Emmanuel Lonca

CP'24

September 5, 2024

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

- PyCSP^3: a Python library for modelling constrained problems
  see https://www.pycsp.org/
- XCSP^3: 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 XCSP^3)
- Quite compact and readable models
  - easy handling of data (from JSON): "one line is enough"

    ```
    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 Python
  - repository with more than 340 models (and data); see https://pycsp.org/models
  - a very large number of series of instances

Purpose of Competitions

The goal of a competition is to:

- evaluate solvers in the same conditions
- collect publicly available benchmarks and data (results, traces, ...)
- identify good new ideas/techniques

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, XCSP^3-core)

For the standard tracks:

- `intension`, `extension`
- `regular`, `mdd`
- `allDifferent`, `allDifferentList`, `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 2024 XCSP^3 Competition

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

Problem	Goal	Exploration	Timeout
CSP	one solution	sequential	30 minutes
COP	best solution	sequential	30 minutes
Fast COP	best solution	sequential	3 minutes
Parallel COP	best solution	parallel	30 minutes

Table: Standard Tracks.

Problem	Goal	Exploration	Timeout
Mini CSP	one solution	sequential	30 minutes
Mini COP	best solution	sequential	30 minutes

Table: Mini-solver Tracks.

Computer Infrastructure

- The cluster was provided by CRIL and is composed of nodes with 2 quad-core Intel(R) Xeon(R) CPU E5-2643 0 @ 3.30GHz, each equipped with 32GiB RAM (24GiB for jobs).
- 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 3 min of CPU time and 4.5 min of Wall Clock time,
  - other sequential solvers were allocated 30 min of CPU time and 45 min of Wall Clock time,
  - parallel solvers were allocated 4 CPU and 30 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*
- one series (AircraftAssemblyLine) has been submitted

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)
AverageAvoiding	`allDifferent`, `minimum`
FastMatrixMultiplication	`lex`, `precedence`, `sum`, `table`*
Fillomino	`element`, `sum`, `table`*
FRB	`table`
Hamming	`lex`, `sum`
HyperSudoku	`allDifferent`
MisteryShopper	`allDifferent`, `channel`, `lex`, `table`
Pentominoes	`allDifferent`, `table`
PoolBallTriangle	`allDifferent`
RotatingWorkforce2	`cardinality`, `count`, `regular`, `sum`
Soccer	`allDifferent`, `sum`, `table`
SocialGolfers	`allDifferent`, `cardinality`, `lex`, `sum`
SolitairePattern	`table`
Subisomorphism	`allDifferent`, `table`
Takuzu	`AllDifferentList`, `sum`
WordSquare	`allDifferentList`, `element`, `table`

COP Problems for the Main Track

COP Problems	Global Constraints
AircraftAssemblyLine	`cumulative`, `noOverlap`, `sum`
AztedDiamondSym	`cardinality`, `sum`, `table`*
BinPacking	`binPacking`, `cardinality`, `nValues`, `lex`, `sum`
Cargo	`cumulative`, `element`, `noOverlap`, `sum`
Charlotte	`allDifferent`, `count`, `sum`, `table`*
Drinking	`sum`, `table`
FoolSolitaire	`element`, `table`
LitPuzzle	`sum`
MaxDensOscLife	`lex`, `sum`, `table`*
Pyramid	`allDifferent`
RubiksCube	`allDifferent`, `element`
SameQueensKnights	`sum`
StillLife	`sum`, `table`
TestScheduling	`cumulative`, `noOverlap`, `maximum`
TravelingTournament	`allDifferent`, `cardinality`, `element`, `regular`, `sum`, `table`*
VRP_LC	`circuit`, `cumulative`, `element`, `sum`
WordGolf	`element`, `sum`
Wordpress	`element`, `sum`

Teams/Solvers (in alphabetic order)

- ACE (C. Lecoutre)
- BTD, miniBTD (M. Cherif, D. Habet, P. Jégou, H. Kanso, C. Terrioux)
- Choco (C. Prud'homme)
  Note: Choco-LCG not ready wrt the deadline of the competition
- CoSoCo (G. Audemard)
- CPMpy (T. Guns, W.Vanroose, T. Sergeys, I.Bleukx, J. Devriendt, D. Tsouros, H. Verhaeghe)
  `cpmpy_mzn_chuffed`, `cpmpy_mzn_gecode`, `cpmpy_z3`
  `cpmpy_exact`, `cpmpy_gurobi`, `cpmpy_ortools`
- Exchequer (M. Mariusz Lester)
- Fun-sCOP (T. Soh, D. Le Berre, H. Nabeshima, M. Banbara, N. Tamura)
- 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)
- Toulbar2 (D. Allouche et al.)

Scanning the Results

Available Resources:

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

CSP (200 instances)

Solver	#solved	#SAT	#UNSAT
Picat	123	100	23
CPMpy_ortools	121	98	23
Fun-sCOP	114	94	20

COP (250 instances)

Solver	Score	#Opt	#BB1	#BB2
CPMpy_ortools	145.5	101	38	13
Picat	128.0	111	14	0
CoSoCo	111.5	63	34	29

Fast COP (250 instances)

Solver	Score	#Opt	#BB1	#BB2
Picat	108.5	85	23	1
CoSoCo_fast	96.0	36	49	22
Choco	87.0	39	42	12

Parallel COP (250 instances)

Solver	Score	#Opt	#BB1	#BB2
CPMpy_ortools	195.5	116	78	3
Choco	110.0	64	35	22
Toulbar2	19.5	14	1	9

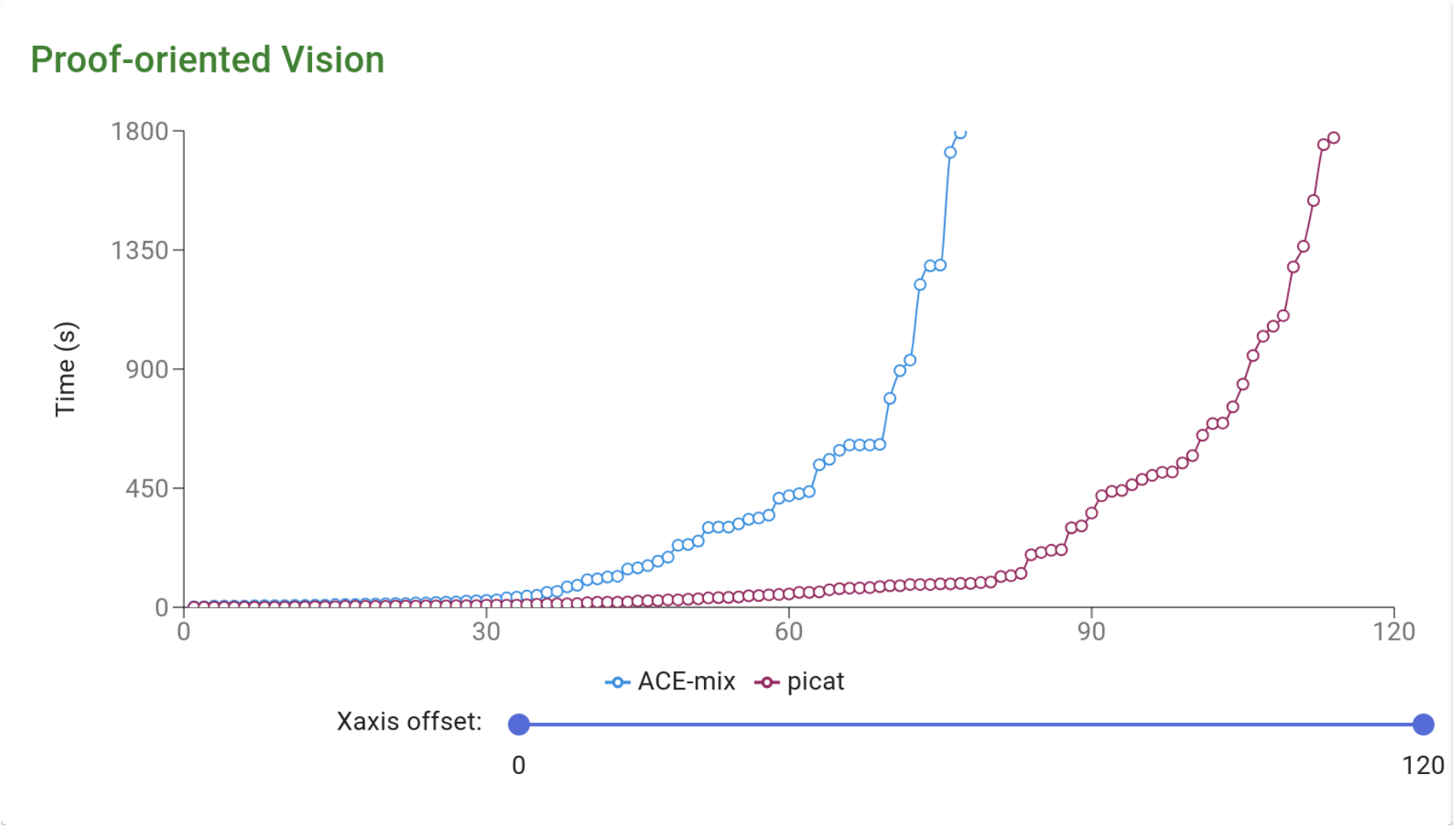
Mini CSP (150 instances)

Solver	#solved	#SAT	#UNSAT
CPMpy_mzn_chuffed	71	57	14
miniBTD	58	46	12
Nacre	54	42	12

Mini COP (150 instances)

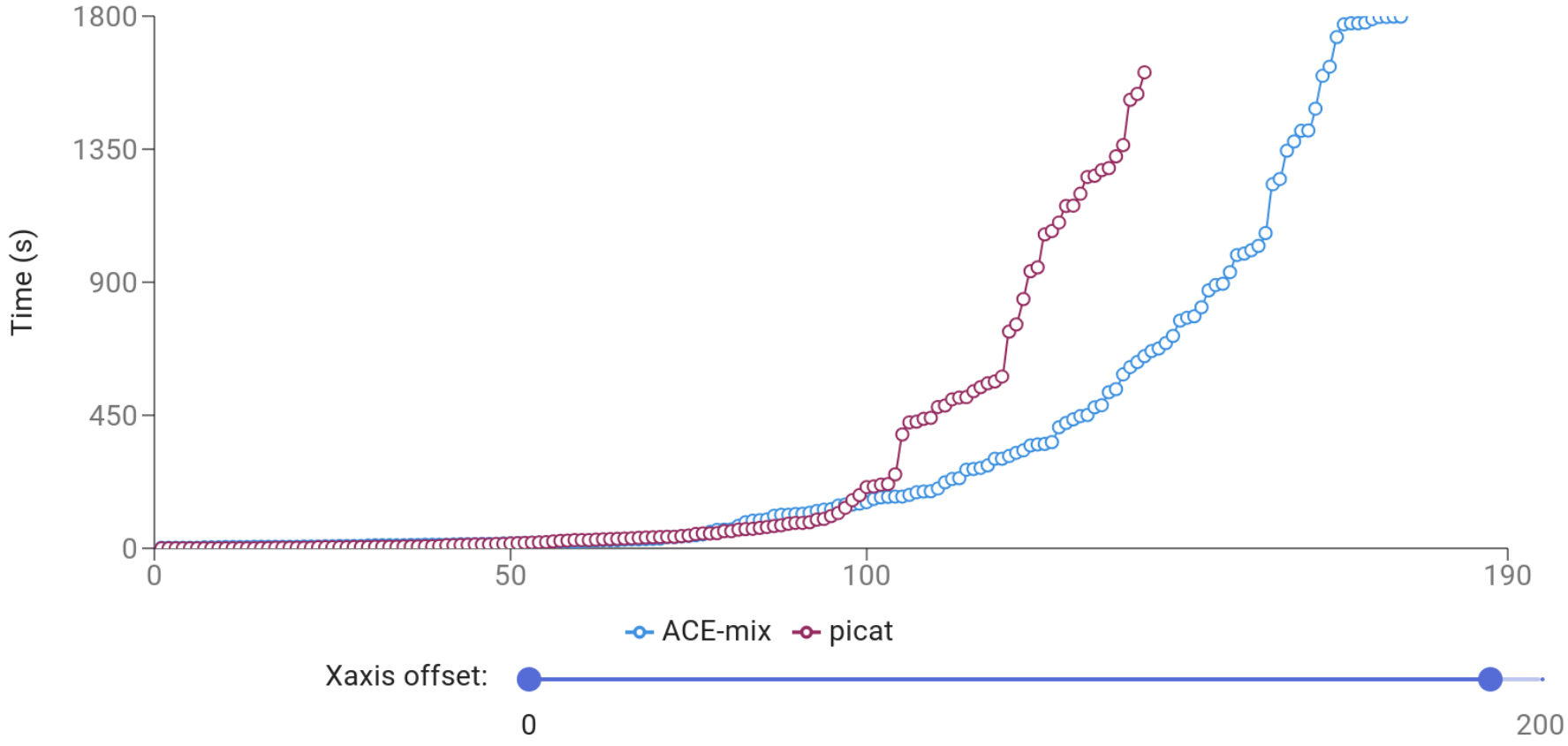
Solver	Score	#Opt	#BB1	#BB2
Exchequer	65.5	46	19	1
MiniRBO	57.0	24	27	12
Toulbar2	37.5	20	15	5
CPMpy_gurobi	37.5	26	10	3

Specific Zoom on COP Track 1/2



Specific Zoom on COP Track 2/2

Search-oriented Vision



Take-away Message

CP solvers now can benefit from:

- solution(-based) phase saving (since 2017)
- three complementary robust variable ordering heuristics (since 2004, 2021 and 2023)

search, reborn!

https://www.cril.univ-artois.fr/XCSP24