

October 2017

Scheduling: CP Optimizer vs Open Source Solvers

Paul Shaw, Philippe Laborie



The CP Open Source Landscape

- There are lots of free CP solvers out there, but not many fully-fledged ones which are taken seriously. The most widely used are:
 - Gecode
 - Choco
 - OR-Tools
- There are a bunch of other solvers, a number of which have leanings towards a particular internal technology (e.g. local search, conflict learning, or lazy clause generation), which make them less general as solvers than the above three.

Gecode

- Comes out of KTH (Sweden) - main driver was Christian Schulte
 - Borrowed a lot from ILOG Solver
 - For a time, was the only credible open source CP Solver
- MIT License
- Team: Christian Schulte (KTH), Guido Tack (Monash), Mikael Lagerkvist (KTH)
- Implements many global constraints from the global constraint catalog
- Native C++. Interfaces are somewhat strange: No Java, Python looks unmaintained, AMPL, Prolog, Lisp, other lesser known systems

Choco

- Developed at the Ecole des Mines de Nantes (EMN)
 - Spin-off company “COSLING” uses Choco as their solver
- BSD License
- Team: Charles Prud'homme (EMN), Jean-Guillaume Fages (COSLING), Xavier Lorca (EMN). Around 10 other non-core contributors.
- Recently received much more attention and looks more active than Gecode
- 100% Java: we are aware of no other interfaces

OR-Tools

- Developed at Google Paris, largely by ex-ILOG employees
 - Highly influenced by ILOG products
- Apache License 2.0
- Team: Laurent Perron, Vincent Furnon, and others
- A part of the solver is specialized in vehicle routing
 - At time of writing, this module seems to produce the most traffic on their forum
- Other tools also available. SAT solver, LP solver and some other OR algorithms
- Native C++. Interfaces in Java, Python, C#

The Question

- A customer may ask the question:
 - “Since CP Optimizer is using CP technology and open source CP solvers exist, why shouldn’t I just save money and use an open source solver for my scheduling problem?”
- There are of course all the standard answers that one can give on IBM’s reputation, how we have great support, our consulting services and so on, that make IBM the right choice. None of which is wrong.
- But these slides are not about that, but about the technical reasons why choosing open source is not a good idea when you have an important scheduling problem, and why CP Optimizer is the right choice.

The Answers

- No open source CP solver has **scheduling as a focus**
- There are no **scheduling experts** in the open source teams
- No open source solver supports **rich scheduling**
- No open source solver has **world-leading solving** of scheduling problems
- No open source solver has **advanced mathematical tools** for model development
- No open source solver demonstrates **year-on-year improvement** in scheduling

No open source CP solver has scheduling as a focus

- Gecode has a single page of constraints on scheduling
 - Only scheduling constraints are for standard resources. No interval variables
- Choco has no scheduling focus
 - One example in the doc is quite involved (lack of scheduling support in Choco)
 - Solved in a fraction of the time with CP Optimizer
- OR-Tools has copied optional interval variables and sequence variables from CP Optimizer
 - But lacks important scheduling concepts such as reservoirs, setups, max usage evaluation, and break calendars

There are no scheduling experts in the open source teams

- Well, there aren't
 - Our team has a scheduling culture dating back to the mid 1990s (ILOG Scheduler)
- Take for example the Gecode page on scheduling
 - http://www.gecode.org/doc-latest/reference/group__TaskModelIntScheduling.html
 - Nearly all algorithms come from the work of Petr Vilim
 - Petr Vilim has been in our team for the past 10 years
- Look at our publications on scheduling:
 - Search for “Petr Vilim”, “Philippe Laborie”, “Jerome Rogerie”

No open source solver supports rich scheduling

- Gecode has very limited support based on only integer variables and constraints to represent resources in quite a limited way
- Choco has essentially no direct scheduling support
- What OR-tools offers is a highly watered-down version of what is in CP Optimizer
- CP Optimizer offers rich modeling with a well-thought out set of scheduling primitives which can be combined in many ways to cover a wide range of scheduling problems
 - Optionality of tasks, general cost functions, general resources, setup times and costs, alternative tasks, recipes, hierarchical structure, state resources and more

No open source solver has world-leading solving of scheduling problems

- Gecode has limited automatic search based on search restarts
- In a Choco scheduling example, it is indicated that supplying a well crafted user-defined search is essential – something CP Optimizer does not require.
- OR-Tools has a single example on solving a Job-shop problem. Here a specific search heuristic is used. The “set times” method, created at ILOG is available.
- CP Optimizer has a sophisticated automatic search using LNS, CPLEX for relaxations, presolve, dedicated heuristics, and more recently failure-directed search
 - *No open source solver comes remotely close to CP Optimizer on solving*
 - CP Optimizer solved many open well-known open scheduling problems:
https://www.researchgate.net/publication/280224455_Failure-directed_Search_for_Constraint-based_Scheduling_Detailed_Experimental_Results

No open source solver has advanced mathematical tools for model development

- CP Optimizer offers mathematical tools, not present in the open source offerings to develop scheduling models
 - Modeling assistance: the solver gives information or warnings on potentially erroneous or inefficient parts of a model
 - Conflict refiner: when the model has no solution, CP Optimizer will deliver a minimal set of constraints which together have no solution, allowing to you nail down the problem
 - Search log: allows you to see what the engine is doing while it solves
 - Objective bound (forthcoming in 12.8): bounds on the objective value are available to indicate the quality of the solutions found

No open source solver demonstrates year on year improvement in scheduling

- CP Optimizer performance on scheduling problems advances year on year
 - We measure performance very carefully on an internal set of over 3000 scheduling problems split into over 140 different families
- This graph shows solving performance from 12.4 to 12.8 normalized to 12.4 performance. Solves are using 4-core machines.

CP Optimizer Scheduling Performance
Versions 12.4 to 12.8



