Research Challenges and Remarks on CP

Nicolas Beldiceanu¹ Pierre Flener²
Jean-Noël Monette² Justin Pearson²
Helmut Simonis³

Mines de Nantes, France

Uppsala University, Sweden

4C, University College Cork, Ireland

CP 2012 Panel: The Future of CP
12 October 2012
Questions

- **Automata:** Propagate by declarative automaton *generators*
- **Verification:** Verify propagator properties (correctness, etc)
- **Synthesis:** From a declarative constraint specification:
  - Generate a (solver-independent) *propagator*
  - Generate a *visualiser*
  - Generate *explanations* (also from a high-level propagator)
- **Scalability:** Design propagators that perform limited propagation (while satisfying some side constraints)
The Global Constraint Catalogue was enriched with a lot of meta data: the latest working version is always at http://www.emn.fr/z-info/sdemasse/aux/doc/catalog.pdf and currently has 3,289 pages.

A community effort should now be started to:

- Identify more core concepts (such as ALLDIFFERENT)
- Define derived concepts applying across all core concepts (generalisation, specialisation, open and soft variants, ...)
- Maintain links to modelling languages and libraries
Towards the Development of *Sustainable* CP Solvers

- Promote **source code** to be associated with submitted and published papers on algorithms; see for instance Prosser’s *Technical Report 2012-333*

- Promote **open-source** solvers, such as Choco, ECLiPSe, Gecode, JaCoP, and Minion

- Promote **solver-independent** algorithms

- Promote **declarative** propagator descriptions
Organise **out-reach meetings** with experts of CS areas, such as *CP meets ML* and *CP meets CAV*

Develop on-line **material** explaining CP to CS experts

Maintain a **showcase** of *significant* benchmarks where:
- CP solvers outperform other solvers
- CP practitioners challenge practitioners of other solvers

Cooperation with at least the SAT, SMT, MIP communities

Limit impact of absence of standard interface to CP solvers
Where can CP make significant contributions?
Integration of combinatorial problem solving technologies (CP, LS, MP, SMT, . . . )

What technical steps are needed for such contributions?
Compilers and interpreters of declarative formulae for the different technologies

What would CP offer compared to other technologies?
Versatility: different models, inference, search, etc