

Looking into the crystal-ball: a bright future for CP

Michela Milano · Pascal Van Hentenryck

Published online: 9 March 2014
© Springer Science+Business Media New York 2014

1 Introduction

An interesting exercise for a research community is to foresee its future, identify key research challenges and open problems and be forward-looking in both technology and application fields.

The Constraint Programming community, in more than 20 years, has performed this exercise several times. One example is the recent book on *Trends in Constraint Programming* edited in 2007 by Benhamou, Jussien and O’Sullivan [1] and the recent challenge paper at AAAI 2012 by O’Sullivan [17] entitled *Opportunities and challenges for Constraint Programming*. Other examples are the earlier papers by Van Hentenryck and Saraswat [8] on *Strategic Directions in Constraint Programming* in 1996 and the ACM 50th Anniversary Issue of ACM Computing Surveys of the same year on Strategic Directions in Computing Research containing a number of forward looking papers [2–7, 9–16, 18–21].

Some of these papers have correctly foreseen the future, some challenges have received less than expected attention, but overall these papers have always provided a useful guide for researchers and newcomers.

Here we are offering another snapshot of the constraint programming community and what its members consider as promising directions. We have intentionally invited contribution without structuring them into a schema. Rather we have preferred a bottom-up call for proposals providing supported-by-evidence viewpoints on future directions of CP.

M. Milano (✉)
DISI-University of Bologna, Bologna, Italy
e-mail: michela.milano@unibo.it

P. Van Hentenryck
NICTA and University of Melbourne, Parkville, Australia
e-mail: Pascal.VanHentenryck@nicta.com.au

The result is of course not structured, but, in our opinion, extremely interesting. The reader should not expect technical papers, with algorithms, theorems, proofs and computational results. The exercise that all the authors have done is to foresee what will be next, what are open challenges that constraint technology should face to evolve.

2 Content of this issue

The current issue contains six papers: two of them concern constraint technology, while four of them are on broad application fields that can benefit from the use of constraint programming and how constraint programming should be enhanced to work on these fields.

2.1 The future of optimization technology

The opening paper of the special issue has been written by an Australian team composed of De La Banda, Wallace, Stuckey and Van Hentenryck and describes how they foresee the future trends in constraint and optimization technology. The paper claims that there are basically two needs. On one hand developers ask for a simplification in the use of optimization technology and for lowering the barriers of entry to using this technology. On the other hand, the raising complexity and scale of new application fields calls for technology advancements and hybridizations.

One research direction that the authors think is relevant concerns conceptual modeling which captures the problem specification in a solver agnostic way through modeling concepts that cover most of the uses of optimization today namely search, patterns for hybridization, symmetry and dominance relations, stochastic and dynamic optimization, debugging and explanation. As a consequence of conceptual modeling, model analysis and transformation tools should be developed.

Finally, as far as solving is concerned, the authors advocate the development of a comprehensive solver containing all the current techniques for optimization with smooth hybridization capabilities and achieving the same performance as a dedicated hybrid solver. Future solvers should also be able to cope with massive amounts of data and to be able to be massively parallelised on modern hardware architectures.

2.2 Toward sustainable development of constraint programming

The second paper, written by Beldiceanu, Flener, Monette, Pearson and Simonis, goes in line with the first, while mostly concentrating on solvers. The main challenge that the paper tries to answer is how to build solvers whose concepts can be easily migrated into the next-generation solvers.

The authors then refine some of the ideas they have on solvers into more technical specifications that would make the solver concepts easily exportable. They focus on global constraints and propagators. In particular, the authors propose the use of generators of automata with regular structure which can be concisely and declaratively described. Also, they advocate the use of machine learning techniques for learning such generators.

Indexicals are another aspect considered in the paper as they should be extended to cope with combinatorial objects of any kind and to automatically discover invariants and synthesise propagators.

Finally verification and explanation is considered arguing that a challenge for the field would be to automatically extract visualizers for global constraints and explanations starting from the declarative description of the constraint.

Finally, the interface of CP with other computer science areas is considered along with actions that could be pursued to advertise CP in other related areas.

2.3 Grand challenges of constraint programming

This paper, written by the current and previous directors of the Cork Constraint Computation Center, O’Sullivan and Freuder, is about the definition of grand challenges for a field and for a research community.

Then the paper analyses one of the grand challenges arising in the modern, connected world: managing and exploiting the so called Big Data, and considers some of its implications for Constraint Programming. The authors identify several areas where big data are produced and should be managed: notable examples are data-centers optimization, human mobility and smart cities, natural resource management and innovative enterprise and public service delivery. Also the dynamic nature of these problems poses a challenge on optimisation of evolving problems, on-line optimization and optimal stopping theory and online distributed stochastic optimization. In these settings, exploiting problem structure and automating large neighborhood search are essential.

Finally, the paper concludes with some open problems to the Holy Grail approach, recognizing that some steps have been done, but there is room for improvement in the future.

2.4 Qualitative modeling in constraint programming

The fourth paper of the issue, written by Kelsey, Kotthoff, Jefferson, Linton, Miguel, Nightingale, and Gent, is about exploiting constraint programming in qualitative modeling. Qualitative modeling has the aim to model the behaviour of systems without estimating parameter values and fixing the exact quantitative dynamics. Qualitative modeling is applied for example to model the dynamics of physical and biological systems at a higher level of abstraction than that obtained by estimation of numerical parameter values for a fixed quantitative model.

The paper presents a cell dynamic case study modelled with constraints.

Finally, a number of avenues of research are listed that would allow not only the production of high quality qualitative models, but also a robust schema for deriving a suitable quantitative model from the space of solutions of a CSP that represents a qualitative model.

2.5 Strategic decision making on complex systems

The fifth paper, authored by Lombardi and Milano from University of Bologna concerns a new challenge for decision making and optimization techniques: deciding in domains where decisions affect and are affected by complex systems. Complex systems exhibit phenomena emerging from a collection of interacting objects, capable to self organize and to adapt their behaviour according to their history and feedback. Such systems are unfortunately impervious to modeling efforts via state-of-the-art combinatorial optimization techniques.

In this setting, we need to have a predictive model, such as a simulator and use it to derive (learn) relations between decisions and their impact on the complex system. Beside

a learning step, several forms of interaction between a decision making component and the predictive models are foreseen in the paper. Finally, the authors advocate the use of ultra large scale decision and optimization solvers able to cope with ultra large scale, uncertain and often ill-defined problems, needing a paradigm shift with respect to traditional techniques.

2.6 Collective decision making: a great opportunity for constraint reasoning

The last paper, authored by Rossi, is about collective decision making, an growing area that has received much attention in the last few years. It is enabled by social networks where small groups of people (friends or families) share an always increasing amount of data concerning their opinions, preferences, life styles, and then can use these data to reach a collective decision.

Rossi argues that constraint programming with its hard and soft constraints could be a perfect environment for expressing people preferences and opinions on specific decisions, where to reason on these preferences and come out with a decision that is collectively taken. CP-nets and soft constraints are two notable examples of formalisms that can be used to model preferences. They have different expressive power and reasoning upon them has different computational complexity.

An interesting section follows on strategic behaviour and its convergence. It opens the possibility of integrating constraint programming with game and voting theory.

Finally, a section on preferences learning and elicitation concludes the paper by arguing that in presence of uncertain preferences one can infer these by interacting with the user or deriving them from historical or related data.

Acknowledgments We would like to thank the Editor in Chief, Gilles Pesant, for his help and suggestions on this special issue. A number of colleagues have acted as reviewers of these position papers. They have provided deep insights and very interesting advices to authors for improving their papers. Finally, we would like to thank all authors for sharing their interesting views and possibly help the community to grow.

References

1. N. Jussien B. O'Sullivan, F.B. (Ed.) (2007). *Trends in constraint programming*. Wiley.
2. Brodsky, A. (1996). Constraint database technology for electronic trade with complex objectives. *ACM Computing Surveys*, 28(4es), 107.
3. Codognot, P. (1996). The virtuality of constraints and the constraints of virtuality. *ACM Computing Surveys*, 28(4es), 60.
4. Dechter, R. (1996). Bucket elimination: a unifying framework for processing hard and soft constraints. *ACM Computing Surveys*, 28(4es), 61.
5. Dincbas, M. (1996). Constraint programming. *ACM Computing Surveys*, 28(4es), 62.
6. Freuder, E.C. (1996). In pursuit of the holy grail. *ACM Computing Surveys*, 28(4es), 63.
7. Hentenryck, P.V. (1996). Constraint programming for combinatorial search problems. *ACM Computing Surveys*, 28(4es), 76.
8. Hentenryck, P.V., & Saraswat, V.A. (1996). Strategic directions in constraint programming. *ACM Computing Surveys*, 28(4), 701–726.
9. Hermenegildo, M.V. (1996). Some callenges for constraint programming. *ACM Computing Surveys*, 28(4es), 64.
10. Jaffar, J., & Yap, R.H.C. (1996). Constraint programming 2000. *ACM Computing Surveys*, 28(4es), 65.
11. Kasif, S. (1996). Towards a constraint-based engineering framework for algorithm design and application. *ACM Computing Surveys*, 28(4es), 66.

12. Mackworth, A.K. (1996). Constraint-based design of embedded intelligent systems. *ACM Computing Surveys*, 28(4es), 67.
13. McAllester, D.A. (1996). The rise of nonlinear mathematical programming. *ACM Computing Surveys*, 28(4es), 68.
14. McAloon, K. (1996). Constraint-based programming. *ACM Computing Surveys*, 28(4es), 69.
15. Montanari, U., & Rossi, F. (1996). Constraint solving and programming: what's next? *ACM Computing Surveys*, 28(4es), 70.
16. Older, W.J. (1996). Clp (intervals). *ACM Computing Surveys*, 28(4es), 71.
17. O'Sullivan, B. (2012). Opportunities and challenges for constraint programming. In *Proceedings of the twenty-sixth AAAI conference on artificial intelligence, 22–26 July 2012*. Toronto, Ontario, Canada.
18. Puget, J.F. (1996). Future of constraint programming. *ACM Computing Surveys*, 28(4es), 72.
19. Ramakrishnan, R. (1996). Constraints in databases. *ACM Computing Surveys*, 28(4es), 73.
20. Saraswat, V.A. (1996). Compositional computing. *ACM Computing Surveys*, 28(4es), 74.
21. Smolka, G. (1996). Constraints in oz. *ACM Computing Surveys*, 28(4es), 75.