Constraint Programming Letters 2 (2008) 1–2 Submitted 12/2007; Published 1/2008

Introduction to the Special Issue on Arc Consistency

M.R.C. van Dongen DONGEN@CS.UCC.IE
University College Cork
Western Road
Cork
Ireland

Editor: M.R.C. van Dongen

Keywords: Filtering Algorithms, Arc Consistency, Bitwise Operations, Residual Support, Maintain Arc Consistency, Time Complexity, Average Time Complexity

Welcome to the second volume of Constraint Programming Letters, which is dedicated to what has become one of the default standard consistencies: arc consistency. This issue brings together a nice blend of 4 papers which should be interesting for researchers who are interested in the complexity of algorithms and for those who are interested in ways to improve existing general purpose arc consistency algorithms. All papers have been formally reviewed and we should like to thank the reviewers for their work.

The remainder of this introduction consists of a brief introduction to each paper in this volume. May you enjoy them!

Lecoutre et al. (2008) present a detailed study on the use of multiple *residues* in arc consistency algorithms during search. Here residues are used to cache previous supports without the need to maintain them upon backtracking. This makes them interesting in combination with a Maintain Arc Consistency (MAC) algorithm. They present a generalised arc consistency algorithm which uses multiple residues and exploits multi-directionality. A theoretical evaluation is presented of the binary version of the algorithm when used as a standalone algorithm and when used as the arc consistency component of a MAC solver. Finally, they present experimental evidence, which shows that if the number of residues is small (1–5) then the number of consistency checks decreases sharply with only a moderate increase in the number of validity checks.

Lecoutre and Vion (2008) propose to represent domains and constraints as bit sequences and to exploit the computer hardware to speed up the observed running time of arc consistency algorithms. Detailed implementation information is presented on how this idea may be used to improve existing arc consistency and MAC algorithms. They present the results from an experimental study which shows the relevance of this approach. Finally, they point out that this approach also has applications to symmetry breaking by dominance detection and to neighbourhood substitutability.

Mehta (2008) presents a new approach to reducing checks and revisions in arc consistency algorithms. At a finer level *support conditions* are exploited which guarantee the existence of a support for some value. If a support condition holds then there is no need to identify the support, thereby saving a sequence of consistency checks. At a coarser level *revision conditions* are exploited which guarantee the existence of a support for all values in the domain of a variable. Should a revision condition hold then an entire revision and queue maintenance can be avoided. Experimental results demonstrate that combining support and revision conditions may save up to 50% of the revisions and 50% of the solution time.

Van Dongen et al. (2008) study the expected time complexity of arc consistency algorithms for random CSPs. Their results are parameterised over the probability that a random check succeeds — this corresponds to random Model A or D CSPs. They present exact formulae for the expected time complexity and the variance of three optimal revision algorithms for random 2-variable CSPs. A comparison of the expected time complexities shows that on average there is very little difference between the algorithms for 2-variable CSPs. A variance analysis shows that the checks which are required by the algorithms is sharply concentrated about their expected values. Finally, they present a (conservative) upper bound for the expected time complexity of the AC-3 algorithm for the general n-variable CSP. This conservative upper bound improves a factor d on the algorithm's worst-case time complexity.

References

Christophe Lecoutre and Julien Vion. Enforcing arc consistency using bitwise operations. *Constraint Programming Letters*, 2:21–35, 2008.

Christophe Lecoutre, Chavalit Likitvivatanavong, Scott G. Shannon, Roland H. C. Yap, and Yuanlin Zhang. Maintaining arc consistency with multiple residues. *Constraint Programming Letters*, 2: 3–19, 2008.

Deepak Mehta. Reducing checks and revisions in the coarse-grained arc consistency algorithms. *Constraint Programming Letters*, 2:37–53, 2008.

M.R.C. van Dongen, A.B. Dieker, and A. Sapozhnikov. The expected value and the variance of the checks required by revision algorithms. *Constraint Programming Letters*, 2:55–77, 2008.