Introducing $buggy_{2-5}$ and $buggy^{s}_{2-5}$

M.R.C. van Dongen

University College Cork

Abstract. This paper presents a brief introduction to two solvers, $buggy_{2-5}$ and $buggy^{s}_{2-5}$, which I have submitted to the binary extensional and binary intensional categories of the Second International CSP Solver Competition. Both solvers are a combination of preprocessing followed by MAC search. The preprocessing consists of domain squashing and enforcing weak k-singleton arc consistency.

1 Main Description

This position paper briefly describes the two solvers, $buggy_{2-5}$ and $buggy^{s}_{2-5}$, which have been submitted to the Second International CSP Solver Competition for the binary extensional and binary intensional categories.

The paper is deliberately kept to a minimum. The main reason is that there's not much that can be said about the solvers. As a matter of fact, initially, I didn't want to participate at all. However, after the low number of contestants for the first round, I decided that it should be better if I did participate. Another reason, which is related to the first, is that I am currently in the process of re-implementing my basic solver and its data types. This basic solver, which underlies $buggy_{2-5}$ and $buggy^{s}_{2-5}$, is only half finished.

The solvers can be best described as follows. Both do some preprocessing, enforce consistency and, if needed, they start a MAC search [Sabin and Freuder, 1994]. The only difference between the solvers is the level of preprocessing. In the following, I shall first describe the local consistency that is enforced by the solvers, then the preprocessing,[1] and then the solution strategy.

Enforcing local consistency means enforcing weak k-singleton arc consistency (weak k-SAC) [van Dongen, 2006] for increasing levels of k, $2 \leq k \leq 5$ using greedy search. Weak k-sac is equivalent to SAC [Debruyne and Bessière, 1997; 2001; Bessière and Debruyne, 2005; Lecoutre and Cardon, 2005; Prosser *et al.*, 2000] if $k = 1$ but stronger if $k > 1$. They start by enforcing arc consistency, followed by weak 2-SAC, weak 3-SAC, weak 4-SAC and weak 5-SAC. The difference between the solvers is that $buggy^{s}_{2-5}$ may enforce higher levels of consistency. Both solvers terminate in case of an inconsistency. In the process of establishing weak k-SAC it is possible that a solution is found, in which case the algorithms terminate. The algorithms for enforcing weak k-SAC [van Dongen, 2006] are closely related to Lecoutre and Cardon's algorithms for establishing SAC [Lecoutre and Cardon, 2005].

[1] Here preprocessing should not be confused with converting the competition's `XML` format to the solvers' native format: this was done at solution-time.

2 Preprocessing

The preprocessing that is done by the solvers involves domain squashing [Gault and Jeavons, 2004]. Here it is assumed, without loss of generality, that the domains have unique values, allowing us to squash more values [Bulatov and Jeavons, 2003]. This preprocessing more or less amounts to eliminating some substitutable values. The domain squashing is performed after enforcing arc consistency. Since the domains squashing may be quite a large overhead if the domains are large, $buggy_{2-5}$ only smashes the extremal values, that is to say the smallest and largest values in the domains. The following paragraph spends a few more words on extremal value squashing. The same strategy is adopted by $buggy^s_{2-5}$, which then attempts to also smash the remaining values. It is allowed a fantastic magic number of 490 seconds maximum domain squashing time. For many instances much more is needed to ensure that no more squashing is possible. It turned out that its extra squashing enabled $buggy^s_{2-5}$ to solve two more instances than $buggy_{2-5}$ but at the price of much more solution time.

Squashing the extremal values is *very* effective, especially for the jobshop and openshop instances, since many (almost) extremal values, v, satisfy the property that if v is part of any solution then so is $v + 1$ or $v - 1$, in which case we can eliminate v. For some of instances, including modified Radio Link Frequency Assignment Problem instances, squashing made the difference between finding a solution or not.

3 Solution Strategy

Before section briefly describes the solution strategy of the solvers, each of which have a different objective. The main goal of $buggy_{2-5}$ is to find a solution. For $buggy_{2-5}$ this is different. Its main objective is to make the problem *inverse consistent* [Freuder and Elfe, 1996], i.e. remove the values that do not participate in any solution. In the process of making the problem inverse consistent, $buggy^s_{2-5}$ outputs the first solution if there is one.

Both solvers terminate as soon as they can decide the problem is unsatiafiable. If the problem is satisfiable the solvers work as follows.

$buggy_{2-5}$ In the case $buggy_{2-5}$ starts by enforcing arc consistency. Next it starts by enforcing weak 2-SAC–5-SAC, stopping if it can find a solution and starting a MAC search otherwise.

$buggy^s_{2-5}$ The strategy of $buggy^s_{2-5}$ is different. It enforces arc consistency followed by weak 2-SAC, weak 3-SAC, and so on. In the process of doing this, it marks all values which participate to a solution and terminates if there are no more values left that do not participate to a solution.

4 Future Work

The solver's data types are still in a development state. They still are not ideal, let alone optimal. I expect that much can be improved. As soon as the implementation is finalised, I intend to write a paper describing the data types and how they affect the implementation of the algorithms that are built on top of them. Work is underway to generalise the domain squashing.

References

[Bessière and Debruyne, 2005] C. Bessière and R. Debruyne. Optimal and suboptimal singleton arc consistency algorithms. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 54–59, 2005.

[Bessière *et al.*, 2005] C. Bessière, R. Coletta, and T. Petit. A generic framework for learning implied constraints. In *Proceedings of the Eleventh International Conference on Principles and Practice of Constraint Programming (CP'2005)*, pages 23–34, 2005.

[Bulatov and Jeavons, 2003] A. Bulatov and P.G. Jeavons. An algebraic approach to multi-sorted constraints. In F. Rossi, editor, *Proceedings of the Ninth International Conference on Principles and Practice of Constraint Programming, CP'2003*, pages 183–198, 2003.

[Debruyne and Bessière, 1997] R. Debruyne and C. Bessière. Some practicable filtering techniques for the constraint satisfaction problem. In M.E. Pollack, editor, *Proccedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI'97)*, pages 412–417, 1997.

[Debruyne and Bessière, 2001] R. Debruyne and C. Bessière. Domain filtering consistencies. *Journal of Artificial Intelligence Research*, 14:205–230, 2001.

[Freuder and Elfe, 1996] E.C. Freuder and C.D. Elfe. Neighborhood inverse consistency preprocessing. In W.J. Clancey and D. Weld, editors, *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 202–208, 1996.

[Gault and Jeavons, 2004] R. Gault and P.G. Jeavons. Implementing a test for tractability. *Journal of Constraints*, 9:139–160, 2004.

[Lecoutre and Cardon, 2005] C. Lecoutre and S. Cardon. A greedy approach to establish singleton arc consistency. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, 2005.

[Prosser *et al.*, 2000] P. Prosser, K. Stergiou, and T. Walsh. Singleton consistencies. In R. Dechter, editor, *Proceedings of the Sixth International Conference on Principles and Practice of Constraint Programming, CP'2000*, pages 353–368, 2000.

[Sabin and Freuder, 1994] D. Sabin and E.C. Freuder. Contradicting conventional wisdom in constraint satisfaction. In A.G. Cohn, editor, *Proceedings of the Eleventh European Conference on Artificial Intelligence*, pages 125–129. John Wiley and Sons, 1994.

[van Dongen, 2006] M.R.C. van Dongen. Beyond singleton arc consistency. In *Proceedings of the Seventeenth European Conference on Artificial (ECAI'2006)*, 2006. SEE ALSO [Bessière *et al.*, 2005].