

**Dekker, Jip J.; Björdal, Gustav; Carlsson, Mats; Flener, Pierre; Monette, Jean-Noël**  
**Auto-tabling for subproblem presolving in MiniZinc.** (English) [Zbl 1425.68386](#)  
*Constraints* 22, No. 4, 512-529 (2017).

Summary: A well-known and powerful constraint model reformulation is to compute the solutions to a model part, say a custom constraint predicate, and tabulate them within an extensional constraint that replaces that model part. Despite the possibility of achieving higher solving performance, this tabling reformulation is often not tried, because it is tedious to perform; further, if successful, it obfuscates the original model. In order to encourage modellers to try tabling, we extend the MiniZinc toolchain to perform the automatic tabling of suitably annotated predicate definitions, without requiring any changes to solvers, thereby eliminating both the tedium and the obfuscation. Our experiments show that automated tabling yields the same tables as manual tabling, and that tabling is beneficial for solvers of several solving technologies.

**MSC:**

[68T20](#) Problem solving in the context of artificial intelligence (heuristics, search strategies, etc.)  
[90C27](#) Combinatorial optimization

**Keywords:**

[presolving](#); [tabling](#); [modelling methodology](#); [MiniZinc](#)

**Software:**

[CHUFFED](#); [Gecode](#); [MiniSat](#); [MiniSearch](#); [MiniZinc](#); [OR-tools](#); [Oscar.cbls](#)

**Full Text:** [DOI](#)

**References:**

- [1] Abdennadher, S; Rigotti, C, Automatic generation of rule-based constraint solvers over finite domains, *ACM Transactions on Computational Logic*, 5, 177-205, (2004) · [Zbl 1367.68259](#)
- [2] Belov, G., Stuckey, P.J., Tack, G., & Wallace, M. (2016). Improved linearization of constraint programming models. In Rueher, M. (Ed.), *textit{CP 2016, LNCS}*, (Vol. 9892 pp. 49-65): Springer.
- [3] Bergman, D., Cire, A.A., van Hoes, W.J., & Hooker, J. (2016). *Decision diagrams for optimization*. Springer. · [Zbl 06653190](#)
- [4] Bessière, C., & Régin, J.C. (1999). Enforcing arc consistency on global constraints by solving subproblems on the fly. In Jaffar, J. (Ed.), *textit{CP 1999, LNCS}*, (Vol. 1713 pp. 103-117): Springer. · [Zbl 0961.68124](#)
- [5] Björdal, G; Monette, JN; Flener, P; Pearson, J, A constraint-based local search backend for minizinc, *Constraints*, 20, 325-345, (2015) · [Zbl 1325.90076](#)
- [6] Boffill, M., Suy, J., & Villaret, M. (2010). A system for solving constraint satisfaction problems with SMT. In Strichman, O., & Szeider, S. (Eds.), *textit{SAT 2010, LNCS}*, (Vol. 6175 pp. 300-305): Springer. · [Zbl 1306.68153](#)
- [7] Carlsson, M; Johansson, M; Larson, J, Scheduling double round-Robin tournaments with divisional play using constraint programming, *European Journal of Operational Research*, 259, 1180-1190, (2017) · [Zbl 1402.90047](#)
- [8] Carlsson, M., Ottosson, G., & Carlson, B. (1997). An open-ended finite domain constraint solver. In Glaser, H., Hartel, P., & Kuchen, H. (Eds.), *textit{PLILP 1997, LNCS}*, (Vol. 1292 pp. 191-206): Springer. · [Zbl 1405.68326](#)
- [9] Cheng, K.C.K., & Yap, R.H.C. (2008). Maintaining generalized arc consistency on ad hoc  $k$ -ary constraints. In Stuckey, P.J. (Ed.), *textit{CP 2008, LNCS}*, (Vol. 5202 pp. 509-523): Springer.
- [10] Chu, G. (2011). *textit{Improving combinatorial optimization}*. Ph.D. thesis, Department of Computing and Information Systems}. Australia: University of Melbourne.
- [11] De Cat, B., Bogaerts, B., Devriendt, J., & Denecker, M. (2013). Model expansion in the presence of function symbols using constraint programming. In Brodsky, A. (Ed.), *textit{ICTAI 2013}* (pp. 1068-1075): IEEE}. The MinisatID solver is available from <https://dtai.cs.kuleuven.be/software/minisatid>. · [Zbl 1367.68259](#)
- [12] De Landtsheer, R., & Ponsard, C. (2013). *Oscar.cbls: An open source framework for constraint-based local search*, *textit{ORBEL-27, the 27th annual conference of the Belgian Operational Research Society}*. Available at <http://www.orbel.be/orbel27/pdf/abstract293.pdf>; the Oscar.cbls solver is available from <https://bitbucket.org/oscarlib/oscar/branch/CBLS>. · [Zbl 1402.90047](#)

- [13] Dekker, J.J. (2016). Sub-problem pre-solving in MiniZinc. Master's thesis, Department of Information Technology, Uppsala University, Sweden. Available at <http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-307145>.
- [14] Demeulenaere, J., Hartert, R., Lecoutre, C., Perez, G., Perron, L., Régim, J., & Schaus, P. (2016). Compact-table: Efficiently filtering table constraints with reversible sparse bit-sets. In Rueher, M. (Ed.), \textit{CP 2016, LNCS}, (Vol. 9892 pp. 207-223): Springer. · Zbl 1151.90525
- [15] Eén, N., & Biere, A. (2005). Effective preprocessing in SAT through variable and clause elimination. In Bacchus, F., & Walsh, T. (Eds.), \textit{SAT 2005, LNCS}, (Vol. 3569 pp. 61-75): Springer. · Zbl 1128.68463
- [16] Gecode Team (2016). Gecode: A generic constraint development environment. <http://www.gecode.org>.
- [17] Gent, IP; Jefferson, C; Kelsey, T; Lynce, I; Miguel, I; Nightingale, P; Smith, BM; Tarim, SA, Search in the patience game 'black hole', AI Communications, 20, 211-226, (2007) · Zbl 1151.90525
- [18] Gent, IP; Jefferson, C; Linton, S; Miguel, I; Nightingale, P, Generating custom propagators for arbitrary constraints, Artificial Intelligence, 211, 1-33, (2014) · Zbl 1405.68326
- [19] Google Optimization Team (2016). or-tools: Google's software suite for combinatorial optimization. <https://developers.google.com/optimization>.
- [20] IBM Knowledge Center. The strong constraint. [http://www.ibm.com/support/knowledgecenter/SSSA5P\\_12.6.3/ilog.odms.ide.help/OPL\\_Studio](http://www.ibm.com/support/knowledgecenter/SSSA5P_12.6.3/ilog.odms.ide.help/OPL_Studio). · Zbl 1325.90076
- [21] Larson, J; Johansson, M, Constructing schedules for sports leagues with divisional and round-Robin tournaments, Journal of Quantitative Analysis in Sports, 10, 119-129, (2014)
- [22] Larson, J., Johansson, M., & Carlsson, M. (2014). An integrated constraint programming approach to scheduling sports leagues with divisional and round-robin tournaments. In Simonis, H. (Ed.), \textit{CPAIOR 2014, LNCS}, (Vol. 8451 pp. 144-158): Springer. · Zbl 06298789
- [23] Le Provost, T., & Wallace, M. (1992). Domain independent propagation, \textit{FGCS 1992, International conference on fifth generation computer systems} (pp. 1004-1011): IOS Press. · Zbl 0862.68023
- [24] Leo, K., & Tack, G. (2015). Multi-pass high-level presolving. In Yang, Q., & Wooldridge, M. (Eds.), \textit{IJCAI 2015} (pp. 346-352): AAAI Press.
- [25] Monette, J.N., Flener, P., & Pearson, J. (2015). Automated auxiliary variable elimination through on-the-fly propagator generation. In Pesant, G. (Ed.), \textit{CP 2015, LNCS}, (Vol. 9255 pp. 313-329): Springer.
- [26] Nethercote, N., Stuckey, P.J., Becket, R., Brand, S., Duck, G.J., & Tack, G. (2007). MiniZinc: Towards a standard CP modelling language. In Bessière, C. (Ed.), \textit{CP 2007, LNCS}. The MiniZinc toolchain is available at <http://www.minizinc.org>, (Vol. 4741 pp. 529-543): Springer.
- [27] Parlett, D. (Ed.) (1990). \textit{The Penguin Book of Patience.} London: Penguin.
- [28] Pesant, G. (2004). A regular language membership constraint for finite sequences of variables. In Wallace, M. (Ed.), \textit{CP 2004, LNCS}, (Vol. 3258 pp. 482-495): Springer. · Zbl 1152.68573
- [29] Rendl, A., Guns, T., Stuckey, P.J., & Tack, G. (2015). MiniSearch: A solver-independent meta-search language for MiniZinc. In Pesant, G. (Ed.), \textit{CP 2015, LNCS}, (Vol. 9255 pp. 376-392): Springer.
- [30] Simonis, H. (2008). Kakuro as a constraint problem. In Flener, P., & Simonis, H. (Eds.), \textit{ModRef 2018, the 7th International Workshop on Constraint Modelling and Reformulation}. <https://www.it.uu.se/research/group/astra/ModRef08/Simonis.pdf>.

This reference list is based on information provided by the publisher or from digital mathematics libraries. Its items are heuristically matched to zbMATH identifiers and may contain data conversion errors. It attempts to reflect the references listed in the original paper as accurately as possible without claiming the completeness or perfect precision of the matching.