

Shishmarev, Maxim; Mears, Christopher; Tack, Guido; Garcia de la Banda, Maria
Visual search tree profiling. (English) Zbl 1334.90170
Constraints 21, No. 1, 77-94 (2016).

Summary: Understanding how the search space is explored for a given constraint problem - and how it changes for different models, solvers or search strategies - is crucial for efficient solving. Yet programmers often have to rely on the crude aggregate measures of the search that are provided by solvers, or on visualisation tools that can show the search tree, but do not offer sophisticated ways to navigate and analyse it, particularly for large trees. We present an architecture for *profiling* a constraint programming search that is based on a lightweight instrumentation of the solver. The architecture combines a visualisation of the search tree with various tools for convenient navigation and analysis of the search. These include identifying repeated subtrees, high-level abstraction and navigation of the tree, and the comparison of two search trees. The resulting system is akin to a traditional program profiler, which helps the user to focus on the parts of the execution where an improvement to their program would have the greatest effect.

MSC:

90C30 Nonlinear programming

Cited in 1 Document

Keywords:

constraint programming; search tree; profiling; comparison; visualisation

Software:

Choco; CHUFFED; COMET; DiSCiPl; DPvis; Gecode; MinisatID; Oz Explorer; SATGraf; VIFID/TRIFID; ZeroMQ

Full Text: [DOI](#)

References:

- [1] Bauer, A., Botea, V., Brown, M., Gray, M., Harabor, D., & Slaney, J. (2010). An integrated modelling, debugging, and visualisation environment for G12. In *CP 2010. LNCS*, (Vol. 6308 pp. 522-536): Springer. · [Zbl 1128.68484](#)
- [2] Burch, M., Raschke, M., & Weiskopf, D. (2010). Indented pixel tree plots. In *Advances in Visual Computing* (pp. 338-349): Springer.
- [3] Carro, M., & Hermenegildo, M. Tools for constraint visualisation: The VIFID/TRIFID tool. In: Deransart et al. [6], pp. 253-272.
- [4] Choi, C.W., Lee, J.H.M., & Stuckey, P.J. (2007). Removing propagation redundant constraints in redundant modeling. *ACM Transactions on Computational Logic*, 28(4). doi:10.1145/1276920.1276925. · [Zbl 1367.68262](#)
- [5] Chu, G.G. (2011). *Improving combinatorial optimization*. Ph.D. thesis: University of Melbourne.
- [6] Deransart, P., Hermenegildo, M.V., & Maluszynski, J. (Eds.) (2000). *Analysis and visualization tools for constraint programming, constraint debugging (DiSCiPl project)*, LNCS, Vol. 1870: Springer.
- [7] Deransart, P. (2004). Main results of the OADymPPaC project. In B. Demoen, & V. Lifschitz (Eds.), *ICLP, LNCS*, (Vol. 3132 pp. 456-457): Springer.
- [8] Dooms, G., Van Hentenryck, P., & Michel, L. (2007). Model-driven visualizations of constraint-based local search. In *CP 2007, LNCS*, (Vol. 4741 pp. 271-285): Springer. · [Zbl 1179.68141](#)
- [9] Feydy, T., & Stuckey, P.J. (2009). Lazy clause generation reengineered. In I.P. Gent (Ed.), *CP, Lecture Notes in Computer Science*, (Vol. 5732 pp. 352-366): Springer.
- [10] Google (2015). Protocol buffers. <https://developers.google.com/protocol-buffers/>.
- [11] Goulard, F., & Benhamou, F. Debugging constraint programs by store inspection. In: Deransart et al. [6], pp. 273-297.
- [12] iMatix (2015). ZeroMQ. <http://zeromq.org>.
- [13] Jussien, N., Rochart, G., & Lorca, X. (2008). Choco: an open source Java constraint programming library. In *CPAIOR'08 Workshop on Open-Source Software for Integer and Constraint Programming (OSSICP'08)* (pp. 1-10).

- [14] Meier, M. (1995). Debugging constraint programs. In *CP'95, LNCS*, (Vol. 976 pp. 204-221): Springer.
- [15] MinisatID team (2015). MinisatID solver. <https://dtai.cs.kuleuven.be/software/minisatid>.
- [16] Müller, T. (1999). Practical investigation of constraints with graph views. In K. Sagonas, & P. Tarau (Eds.) *Proceedings of the International Workshop on Implementation of Declarative Languages (IDL'99)*.
- [17] Newsham, Z., Lindsay, W., Liang, J.H., Czarnecki, K., Fischmeister, S., & Ganesh, V. (2014). SATGraf: Visualizing community structure in boolean SAT instances. <https://ece.uwaterloo.ca/vganesh/EvoGraph/Home.html>. · [Zbl 06512565](#)
- [18] Schulte, C. (1997). Oz explorer: a visual constraint programming tool. In L. Naish (Ed.), *ICLP* (pp. 286-300): MIT Press.
- [19] Schulte, C., Lagerkvist, M.Z., & Tack, G. (2006). Gecode: Generic constraint development environment. <http://www.gecode.org>.
- [20] Schulte, C., Tack, G., & Lagerkvist, M.Z. (2015). Modeling and programming with Gecode. <http://www.gecode.org/doc-latest/MPG.pdf>.
- [21] Simonis, H., & Aggoun, A. Search-tree visualisation. In: Deransart et al. [6], pp. 191-208. · [Zbl 0942.68551](#)
- [22] Simonis, H., Davern, P., Feldman, J., Mehta, D., Quesada, L., & Carlsson, M. (2010). A generic visualization platform for CP. In *CP 2010, LNCS*, (Vol. 6308 pp. 460-474): Springer.
- [23] Sinz, C. (2004). Visualizing the internal structure of SAT instances (preliminary report). In *SAT*.
- [24] Sinz, C; Dieringer, EM; Bacchus, F (ed.); Walsh, T (ed.), Dpvis a tool to visualize the structure of SAT instances, 257-268, (2005), Berlin Heidelberg · [Zbl 1128.68484](#)
- [25] Smith, B.M., Stergiou, K., & Walsh, T. (1999). Modelling the Golomb ruler problem. Tech. rep., University of Leeds School of Computer Studies.
- [26] Van Cauwelaert, S., Lombardi, M., & Schaus, P. (2015). Understanding the potential of propagators. In *Integration of AI and OR Techniques in Constraint Programming* (pp. 427-436): Springer. · [Zbl 06605772](#)
- [27] Wallace, M.G., Novello, S., & Schimpf, J. (1997). ECLiPSe : a platform for constraint logic programming. *ICL Systems Journal*, *12*(1).

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.