Université Fédérale de Toulouse, ANITI
INRAE, UR 875, 31326 Toulouse, France

toulbar2

An exact cost function network solver
Github source code from June 13, 2025

- David Allouche
- Abdelkader Beldjilali
- Samuel Buchet
- Marianne Defresne
- Valentin Durante
- Simon de Givry*
- George Katsirelos
- Pierre Montalbano
- Thomas Schiex
- Guidio Sewa
- Fulya Trösser
- Matthias Zytnicki

* contact author
simon.de-givry@inrae.fr

Introduction

toulbar2 is an open-source C++ solver for cost function networks (CFN). It is available at `https://github.com/toulbar2/toulbar2`, with an MIT license and a documentation describing its interfaces with C++ and Python.

The constraints and objective function are factored into local functions on discrete variables. Each function returns a cost for any assignment of its variables. Constraints are represented as functions with costs in $\{0, \top\}$ where $\top$ is an upper bound cost associated with forbidden assignments. toulbar2 looks for a non-forbidden assignment of all variables that minimizes the sum of all functions. In general, this problem is NP-hard [9]. Using on-the-fly translation, toulbar2 can also directly solve optimization problems on other graphical models, such as Maximum Probability Explanation on Bayesian networks, and Maximum A Posteriori on Markov random fields [14, 8]. It can also read partial weighted MaxSAT problems (wcnf format), (quadratic) pseudo-Boolean optimization problems (opb,wbo), integer (quadratic) linear programs (cplex/lp), and constrained satisfaction and optimization problems (CSP/COP in XCSP3 format).

toulbar2 provides and uses by default an *anytime* hybrid best-first branch-and-bound algorithm (HBFS) [1] that tries to quickly provide good solutions together with an upper bound on the gap between the cost of each solution and the (unknown) optimal cost. Thus, even when it is unable to prove optimality, it will bound the quality of the provided solution. It can also apply a variable neighborhood search algorithm that exploits the problem decomposition [21]. Both algorithms are complete (if enough CPU-time is given) and can be run in parallel using OpenMPI [3, 21]. The variable ordering heuristic is *dom/wdeg* [5] combined with *last conflict* [16]. The value ordering heuristic exploits the last solution found if any [12] or else EDAC existential value [11]. EDAC is also used as soft local consistency during search, providing lower bounds and pruning forbidden values [7]. By default, a weaker relaxed version is used for pseudo-Boolean linear constraints [18]. Stronger soft local consistencies (VAC [7], including VAC on pseudo-Boolean linear constraints [19], and VPWC [17]) can be applied in preprocessing or during search. Variable elimination is performed during search and is restricted to variables with at most two neighbors [15]. More preprocessing techniques, such as cost function decomposition [13] and relaxation-aware probing heuristic (RASPS [23]) are or can be done before search, and pruning by dominance is also applied during search [10].

Beyond the service of providing optimal solutions, toulbar2 can also find a sequence of diverse solutions [22] or exhaustively enumerate solutions below a cost threshold and perform a guaranteed approximate weighted counting of solutions. For stochastic graphical models, this means that toulbar2 will compute the partition function (or the normalizing constant). These problems being #P-complete, toulbar2 runtimes can quickly increase on such problems. An extension of toulbar2 for bi-criteria optimization problems provides anytime Pareto fronts [6].

toulbar2 was originally developed by the Toulouse (INRAE, MIAT) and Barcelona (UPC, IIIA-CSIC) teams, hence the name of the solver. Additional contributions by:

- Caen University, France (GREYC) and University of Oran, Algeria for (parallel) variable neighborhood search methods [21] ;
- The Chinese University of Hong Kong and Caen University, France (GREYC) for global cost functions [2] ;
- Marseille University, France (LSIS) for tree decomposition heuristics ;
- Ecole des Ponts ParisTech, France (CERMICS/LIGM) for INCOP local search solver [20] ;
- Université de Toulouse (IRIT) and Université du Littoral Côte d'Opale, France for iterated local search with partition crossover [4] ;
- Artois University, France (CRIL) for the XCSP3 format reader of CSP and COP instances.

XCSP'2025 Competition Configuration Settings

For the XCSP'2025 competition, we used the following settings for toulbar2:

- For *COP sequential* and *mini COP* tracks, HBFS [1] was used with no dichotomic branching, VAC and RASPS in preprocessing, and sorting periodically (at every number of open nodes being a power of 2) its best-first priority queue with *dom/wdeg* heuristic values to break ties ; command line:
  `DIR/toulbar2 -timer=TIMELIMIT -d: -A -vaclin -rasps -raspsini -sopen=1 -v=-1 BENCHNAME`
- For *COP parallel* track, parallel HBFS [3] was used with the available number of cores and MPI compilation settings, and default settings for the rest (except no dichotomic branching) ; command line:
  `mpirun -n NBCORE DIR/toulbar2mpi -timer=TIMELIMIT -d: -v=-1 BENCHNAME`

A more efficient encoding of constraints in extensions with star values (used to compress the list of allowed tuples) has been implemented. It relies on a dual encoding with an additional variable, the domain of which contains the compressed list of allowed tuples (one value per compressed tuple). Binary constraints ensure compatibility between this extra variable and the variables within the scope of the original constraint.

Pseudo-Boolean Competition 2025 Configuration Settings

For the 2025 pseudo-Boolean competition, we used the following settings for toulbar2:

- For *OPT-LIN, PARTIAL-LIN*, and *SOFT-LIN* tracks, HBFS [1] was used with default settings, original integer costs (precision=0), and a hidden variable encoding to recover non-Boolean domains ; command line:
  `DIR/toulbar2 -timer=TIMELIMIT -precision=0 -n: -hve=512 -f=3 -v=-1 BENCHNAME`

We applied a hidden variable encoding [17] for *tight* pseudo-Boolean constraints. A constraint is said *tight* if its number of allowed tuples is less than or equal to its arity plus one. At-Most-One and Exactly-One constraints are tight. An extra preprocessing rule is applied to replace Boolean variables with their corresponding hidden variable in non-tight constraints. The whole process allows to recover original domains (here, up to size 512) for non-Boolean instances that have been translated into pseudo-Boolean format. See the results for *PARTIAL-LIN/wcsp* benchmarks.

Acknowledgements

toulbar2 has been partially funded by the French Agence Nationale de la Recherche (projects STAL-DEC-OPT from 2006 to 2008, ANR-10-BLA-0214 Ficolofo from 2011 to 2014, and ANR-16-CE40-0028 DemoGraph from 2017 to 2021) and a PHC PROCORE project number 28680VH (from 2013 to 2015). It is currently supported by Cluster IA ANITI (ANR-19-P3IA-0004 and ANR-24-P3IA-0004) and ANR project GMLaS (2024-2028).

References

[1] D Allouche, S de Givry, G Katsirelos, T Schiex, and M Zytnicki. Anytime Hybrid Best-First Search with Tree Decomposition for Weighted CSP. In *Proc. of CP-15*, pages 12–28, Cork, Ireland, 2015.

[2] David Allouche, Christian Bessière, Patrice Boizumault, Simon de Givry, Patricia Gutierrez, Jimmy H.M. Lee, Ka Lun Leung, Samir Loudni, Jean-Philippe Métivier, Thomas Schiex, and Yi Wu. Tractability-preserving transformations of global cost functions. *Artificial Intelligence*, 238:166–189, 2016.

[3] Abdelkader Beldjilali, Pierre Montalbano, David Allouche, George Katsirelos, and Simon de Givry. Parallel Hybrid Best-First Search. In *Proc. of CP-22*, volume 235, pages 7:1–7:10, Haifa, Israel, 2022. Slides of the presentation https://miat.inrae.fr/degivry/Beldjilali22Slides.pdf and supplementary materials https://miat.inrae.fr/degivry/Beldjilali22Supp.pdf.

[4] François Beuvin, Simon de Givry, Thomas Schiex, Sébastien Verel, and David Simoncini. Iterated local search with partition crossover for computational protein design. *Proteins: Structure, Function, and Bioinformatics*, 2021.

[5] F Boussemart, F Hemery, C Lecoutre, and L Sais. Boosting systematic search by weighting constraints. In *ECAI*, volume 16, page 146, 2004.

[6] Samuel Buchet, David Allouche, Simon de Givry, and Thomas Schiex. Bi-objective discrete graphical model optimization. In *Proc. of CP-AI-OR'2024*, pages 136–152, Uppsala, Sweden, 2024.

[7] M. Cooper, S. de Givry, M. Sanchez, T. Schiex, M. Zytnicki, and T. Werner. Soft arc consistency revisited. *Artificial Intelligence*, 174(7–8):449–478, 2010.

[8] Martin C. Cooper, Simon de Givry, and Thomas Schiex. Graphical models: Queries, complexity, algorithms (tutorial). In *37th International Symposium on Theoretical Aspects of Computer Science (STACS-20)*, volume 154 of *LIPIcs*, pages 4:1–4:22, Montpellier, France, 2020.

[9] Martin C. Cooper, Simon de Givry, and Thomas Schiex. *Valued Constraint Satisfaction Problems*, pages 185–207. Springer International Publishing, 2020.

[10] S de Givry, S Prestwich, and B O'Sullivan. Dead-End Elimination for Weighted CSP. In *Proc. of CP-13*, pages 263–272, Uppsala, Sweden, 2013.

[11] S. de Givry, M. Zytnicki, F. Heras, and J. Larrosa. Existential arc consistency: Getting closer to full arc consistency in weighted CSPs. In *Proc. of IJCAI-05*, pages 84–89, Edinburgh, Scotland, 2005.

[12] E Demirovic, G Chu, and P J. Stuckey. Solution-based phase saving for CP: A value-selection heuristic to simulate local search behavior in complete solvers. In *Proc. of CP-18*, pages 99–108, Lille, France, 2018.

[13] A Favier, S de Givry, A Legarra, and T Schiex. Pairwise decomposition for combinatorial optimization in graphical models. In *Proc. of IJCAI-11*, Barcelona, Spain, 2011. Video demonstration at https://miat.inrae.fr/degivry/Favier11.mov.

[14] D Koller and N Friedman. *Probabilistic graphical models: principles and techniques.* The MIT Press, 2009.

[15] J. Larrosa. Boosting search with variable elimination. In *Principles and Practice of Constraint Programming - CP 2000*, volume 1894 of *LNCS*, pages 291–305, Singapore, September 2000.

[16] C. Lecoutre, L Saïs, S. Tabary, and V. Vidal. Reasoning from last conflict(s) in constraint programming. *Artificial Intelligence*, 173:1592,1614, 2009.

[17] Pierre Montalbano, David Allouche, Simon de Givry, George Katsirelos, and Tomáš Werner. Virtual pairwise consistency in cost function networks. In *Proc. of CP-AI-OR'2023*, Nice, France, 2023.

[18] Pierre Montalbano, Simon de Givry, and George Katsirelos. Multiple-choice knapsack constraint in graphical models. In *Proc. of CP-AI-OR'2022*, Los Angeles, CA, 2022.

[19] Pierre Montalbano, Simon de Givry, and George Katsirelos. Virtual arc consistency for linear constraints in cost function networks. In *First Workshop on Discrete Optimization with Soft Constraints (Soft 2024)*, Girona, Spain, 2024.

[20] B. Neveu and G. Trombettoni. INCOP: An Open Library for INcomplete Combinatorial OPtimization. In *Proc. of CP-03*, pages 909–913, Cork, Ireland, 2003.

[21] Abdelkader Ouali, David Allouche, Simon de Givry, Samir Loudni, Yahia Lebbah, Lakhdar Loukil, and Patrice Boizumault. Variable neighborhood search for graphical model energy minimization. *Artificial Intelligence*, 278(103194):22p., 2020.

[22] Manon Ruffini, Jelena Vucinic, Simon de Givry, George Katsirelos, Sophie Barbe, and Thomas Schiex. Guaranteed diversity and optimality in cost function network based computational protein design methods. *Algorithms*, 14(6):168, 2021.

[23] Fulya Trösser, Simon de Givry, and George Katsirelos. Relaxation-aware heuristics for exact optimization in graphical models. In *Proc. of CP-AI-OR'2020*, pages 475–491, Vienna, Austria, 2020.