toulbar2 Reference Manual

Release 1.0.0

INRAE

Dec 13, 2024

CONTENTS

1 Introduction 1

CHAPTER

ONE

INTRODUCTION

Cost Function Network Solver	toulbar2
Copyright	toulbar2 team
Source	https://github.com/toulbar2/toulbar2

toulbar2 can be used as a stand-alone solver reading various problem file formats (wcsp, uai, wcnf, qpbo) or as a C++ library.

This document describes the WCSP native file format and the toulbar2 C++ library API.

Note

Use cmake flags `LIBTB2=ON` and `TOULBAR2_ONLY=OFF` to get the toulbar2 C++ library libtb2.so and toulbar2test executable example.

See also : `src/toulbar2test.cpp`.

Exact optimization for cost function networks and additive graphical models

(_README_1)= ## What is toulbar2?

toulbar2 is an open-source black-box C++ optimizer for cost function networks and discrete additive graphical models. This also covers Max-SAT, Max-Cut, QUBO (and constrained variants), among others. It can read a variety of formats. The optimized criteria and feasibility should be provided factorized in local cost functions on discrete variables. Constraints are represented as functions that produce costs that exceed a user-provided primal bound. toulbar2 looks for a non-forbidden assignment of all variables that optimizes the sum of all functions (a decision NP-complete problem).

toulbar2 won several competitions on deterministic and probabilistic graphical models:

- Max-CSP 2008 Competition [CPAI08][cpai08] (winner on 2-ARY-EXT and N-ARY-EXT)
- Probabilistic Inference Evaluation [UAI 2008][uai2008] (winner on several MPE tasks, inra entries)
- 2010 UAI APPROXIMATE INFERENCE CHALLENGE [UAI 2010][uai2010] (winner on 1200-second MPE task)
- The Probabilistic Inference Challenge [PIC 2011][pic2011] (second place by ficolofo on 1-hour MAP task)
- UAI 2014 Inference Competition [UAI 2014][uai2014] (winner on all MAP task categories, see Proteus, Robin, and IncTb entries)
- [XCSP3][xcsp] Competitions (second place on Mini COP and Parallel COP tracks in 2022, first place on Mini COP in 2023, third place in 2024)
- UAI 2022 Inference Competition [UAI 2022][uai2022] (winner on all MPE and MMAP task categories)

[cpai08]: http://www.cril.univ-artois.fr/CPAI08/ [uai2008]: http://graphmod.ics.uci.edu/uai08/Evaluation/Report [uai2010]: http://www.cs.huji.ac.il/project/UAI10/summary.php [pic2011]: http://www.cs.huji.ac.il/project/

PASCAL/board.php [uai2014]: https://personal.utdallas.edu/~vibhav.gogate/uai14-competition/leaders.html [xcsp]: https://xcsp.org/competitions [uai2022]: https://uaicompetition.github.io/uci-2022/results/final-leader-board

toulbar2 is now also able to collaborate with ML code that can learn an additive graphical model (with constraints) from data (see the associated [paper](https://miat.inrae.fr/schiex/Export/Pushing_Data_in_your_CP_model.pdf), [slides](https://miat.inrae.fr/schiex/Export/Pushing_Data_in_your_CP_model-Slides.pdf) and [video](https://www.youtube.com/watch?v=IpUr6KIEjMs) where it is shown how it can learn user preferences or how to play the Sudoku without knowing the rules). The current CFN learning code is available on [GitHub](https://github.com/toulbar2/CFN-learn).

(_README_2)= ## Installation from binaries

You can install toulbar2 directly using the package manager in Debian and Debian derived Linux distributions (Ubuntu, Mint,...):

sudo apt-get update sudo apt-get install toulbar2 toulbar2-doc

For the most recent binary or the Python API, compile from source.

(_README_3)= ## Python interface

An alpha-release Python interface can be tested through pip on Linux and MacOS:

python3 -m pip install –upgrade pip python3 -m pip install pytoulbar2

The first line is only useful for Linux distributions that ship "old" versions of pip.

Commands for compiling the Python API on Linux/MacOS with cmake (Python module in lib/*/pytb2.cpython*.so):

pip3 install pybind11 mkdir build cd build cmake -DPYTB2=ON .. make

Move the cpython library and the experimental [pytoulbar2.py](https://github.com/toulbar2/toulbar2/raw/master/pytoulbar2/pytoulbar2.py) python class wrapper in the folder of the python script that does "import pytoulbar2".

(_README_4)= ## Download

Download the latest release from GitHub (https://github.com/toulbar2/toulbar2) or similarly use tag versions, e.g.:

git clone –branch 1.2.0 https://github.com/toulbar2/toulbar2.git

(_README_5)= ## Installation from sources

Compilation requires git, cmake and a C++-20 capable compiler (in C++20 mode).

Required library: * libgmp-dev * bc (used during cmake)

Recommended libraries (default use): * libboost-graph-dev * libboost-iostreams-dev * libboost-serialization-dev * zlib1g-dev * liblzma-dev * libbz2-dev * libeigen3-dev

Optional libraries: * libjemalloc-dev * pybind11-dev * libopenmpi-dev * libboost-mpi-dev * libicuuc * libicui18n * libicudata * libxml2-dev * libxcsp3parser

On MacOS, run ./misc/script/MacOS-requirements-install.sh to install the recommended libraries. For Mac with ARM64, add option -DBoost=OFF to cmake.

Commands for compiling toulbar2 on Linux/MacOS with cmake (binary in build/bin/*/toulbar2):

mkdir build cd build cmake .. make

Commands for statically compiling toulbar2 on Linux in directory toulbar2/src without cmake:

bash cd src echo '#define Toulbar_VERSION "1.2.0"' > ToulbarVersion.hpp g++ -o toulbar2 -std=c++20 -O3 -DNDEBUG -march=native -flto -static -static-libgcc -static-libstdc++ -DBOOST -DLONGDOUBLE_PROB -DLONGLONG_COST -DWCSPFORMATONLY

> -I. -I./pils/src tb2*.cpp applis/*.cpp convex/*.cpp core/*.cpp globals/*.cpp incop/*.cpp mcriteria/*.cpp pils/src/exe/*.cpp search/*.cpp utils/*.cpp vns/*.cpp ToulbarVersion.cpp -lboost_graph -lboost_iostreams -lboost_serialization -lgmp -lz -lbz2 -llzma

Use OPENMPI flag and MPI compiler for a parallel version of toulbar2:

> bash cd src echo '#define Toulbar_VERSION "1.2.0"' > ToulbarVersion.hpp mpicxx -o toulbar2 -std=c++20 -O3 -DNDEBUG -march=native -flto -DBOOST -DLONGDOUBLE_PROB -DLONGLONG_COST -DWCSPFORMATONLY -DOPENMPI
>
> > -I. -I./pils/src tb2*.cpp applis/*.cpp convex/*.cpp core/*.cpp globals/*.cpp incop/*.cpp mcriteria/*.cpp pils/src/exe/*.cpp search/*.cpp utils/*.cpp vns/*.cpp ToulbarVersion.cpp -lboost_graph -lboost_iostreams -lboost_serialization -lboost_mpi -lgmp -lz -lbz2 -llzma

Replace LONGLONG_COST by INT_COST to reduce memory usage by two and reduced cost range (costs must be smaller than 10^8).

Replace WCSPFORMATONLY by XMLFLAG3 and add libxcsp3parser.a from xcsp.org in your current directory for reading XCSP3 files:

> bash cd src echo '#define Toulbar_VERSION "1.2.0"' > ToulbarVersion.hpp mpicxx -o toulbar2 -std=c++20 -O3 -DNDEBUG -march=native -flto -DBOOST -DLONGDOUBLE_PROB -DLONGLONG_COST -DXMLFLAG3 -DOPENMPI
>
> > -I/usr/include/libxml2 -I. -I./pils/src -I./xmlcsp3 tb2*.cpp applis/*.cpp convex/*.cpp core/*.cpp globals/*.cpp incop/*.cpp mcriteria/*.cpp pils/src/exe/*.cpp search/*.cpp utils/*.cpp vns/*.cpp ToulbarVersion.cpp -lboost_graph -lboost_iostreams -lboost_serialization -lboost_mpi -lxml2 -licuuc -licui18n -licudata libxcsp3parser.a -lgmp -lz -lbz2 -llzma -lm -lpthread -ldl

 toulbar2 is currently maintained by Simon de Givry, INRAE - MIAT, Toulouse, France (simon.de-givry@inrae.fr)