

SUNNY-CP 2.2

SUNNY-CP: a Parallel CP Portfolio Solver

sunny-cp [5] is a parallel portfolio solver that allows one to solve a Constraint (Satisfaction/Optimization) Problem defined in the MiniZinc language. It essentially implements the SUNNY algorithm described in [1][2][3] and extends its sequential version [4].

sunny-cp is built on top of state-of-the-art constraint solvers, including: Choco, Chuffed, HaifaCSP, JaCoP, MinisatID, OR-Tools, Picat, Chuffed, and Gecode.

In a nutshell, sunny-cp relies on two sequential steps:

1. PRE-SOLVING: consists in the parallel execution of a (maybe empty) static schedule and the neighborhood computation of underlying k-NN algorithm;
2. SOLVING: consists in the parallel and cooperative execution of a number of the predicted solvers, selected by means of SUNNY algorithm.

sunny-cp won the gold medal in the open track of MiniZinc Challenges 2015, 2016, and 2017, and the silver medal in 2018 and 2019 [6].

Contents of this git repository

- bin contains the executables of sunny-cp
- kb contains the utilities for the knowledge base of sunny-cp
- src contains the sources of sunny-cp
- solvers contains the utilities for the constituent solvers of sunny-cp
- test contains some MiniZinc examples for testing sunny-cp
- tmp is aimed at containing the temporary files produced by sunny-cp
- docker contains the dockerfile used to generate the image in the dockerhub

Installation & Usage

To install sunny-cp it is possible to use [Docker](#) available for the majority of the operating systems.

It can then be used by command line or by simply sending a post request to the server deployed by using docker.

The Docker image is available in Docker Hub. To download its image please run the following command.

```
sudo docker pull jacopomauro/sunny-cp
```

To execute sunny-cp from command line you can use the command

```
docker run --rm -i -t --entrypoint /bin/bash jacopomauro/sunny-cp
```

You will get shell control inside the docker container and you can trigger sunny-cp by invoking the sunny-cp command (sunny-cp --help for getting information on its command line usage).

Note that sunny-cp will run inside the container. MiniZinc files can be shared from the host computer to the Docker container by using Docker volumes (see <https://docs.docker.com/storage/volumes/> for more information). For example, assuming that the MiniZinc file test.mzn to solve is in the folder /host_dir , to run sunny-cp on that MiniZinc model it is possible to first invoke the command

```
docker run --rm -it --entrypoint /bin/bash -v /host_dir:/cont_dir jacopomauro/sunny-cp
```



and then, after getting the shell, run the command

```
sunny-cp /cont_dir/test.mzn
```

Usage of sunny as a service

sunny-cp can be also used a service and access by using HTTP post requests. To deploy sunny's service run the following command.

```
sudo docker run -d -p <PORT>:9001 --name sunny_cp_container jacopomauro/sunny-cp
```

where <PORT> is the port used to use the functionalities of the service.

Assuming that <MZN> is the path of the mzn file to solve, to run the solver on it is possible to invoke it by a multipart post request as follows.

```
curl -F "mzn=@<MZN>" http://localhost:<PORT>/process
```

This will run sunny-cp with the default parameters on the minizinc instance. If a <DZN> file is also needed, sunny-cp can be invoked as follows.

```
curl -F "mzn=@<MZN>" -F "dzn=@<DZN>" http://localhost:<PORT>/process
```

sunny-cp options can be passed by adding the string "option=value" as additional part of the request.

For instance to solve the <MZN> using only the gecode solver (option -P) the post request to perform is the following one.

```
curl -F "-P=gecode" -F "mzn=@<MZN>" http://localhost:<PORT>/process
```

To see the options supported by sunny-cp please run the following command.

```
curl -F "--help=" http://localhost:<PORT>/process
```

To select sunny-cp flags (like --help above) it is possible to add the string "flag=". For example, the option --mzn is set with -F "--mzn=".

Note that the post requests will return the output generated by sunny-cp at the end of its execution. In case partial solutions are need, sunny should be accessed from the command line as explained above. Moreover, canceling the HTTP request will not automatically kill the execution of sunny-cp that will instead continue to execute the solver until a solution is found or the timeout has been reached.

To understand what are the solvers installed you can use the following get request.

```
curl http://localhost:<PORT>/solvers
```

It is also possible to get the feature vector of an instance by using the following post request.

```
curl -F "mzn=@<MZN>" -F "dzn=@<DZN>" http://localhost:<PORT>/get_features
```

To clean up please lunch the following commands:

```
sudo docker stop sunny_cp_container
sudo docker rm sunny_cp_container
sudo docker rmi jacobomauro/sunny-cp
```

Solvers

By default, sunny-cp uses the solvers contained in the MiniZinc bundle, that is:

- [Gecode](#)
- [Chuffed](#)
- [OSICBS](#)

It is however possible to use, via Docker, the following solvers:

- [OR-Tools](#) (version v6.9.1)
- [Choco](#) (version 4.0.4)
- [Picat SAT](#) (version 2.3)
- [JaCoP](#) (version 4.4)
- [MinisatID](#) (version 3.11.0)
- [HaifaCSP](#) (version 1.3.0)
- [Yuck](#) (version 20180303)

Once a solver is installed on your machine, it is easy to add it to the portfolio and to customize its settings. For more details, see the README file in the /solvers folder and the sunny-cp usage.

Note that sunny-cp does not guarantee that its constituent solvers are bug free. However, the user can check the soundness of a solution with the command line option `--check-solvers`. In particular we recommend the usage of this option for haifacsp and minisatid (they current versions are indeed bugged).

Features

During the presolving phase (in parallel with the static schedule execution) sunny-cp extracts a feature vector of the problem in order to compute the solvers schedule possibly run in the solving phase. By default, the feature vector is extracted by mzn2feat tool available at <https://github.com/CP-Unibo/mzn2feat>.

When using docker this tool is already installed in the image. In case of local installation it needs otherwise to be installed manually following the instruction in the README file of mzn2feat.

Note that the user can define its own extractor by implementing a corresponding class in `src/features.py`

Knowledge Base

The SUNNY algorithm on which sunny-cp relies needs a knowledge base, that consists of a folder containing the information relevant for the schedule computation. For the time being the default knowledge base of SUNNY-CP is empty. However, the user has the possibility of defining a knowledge base to use.

The sunny-cp/kb/mznc1215 folder contains a knowledge base consisting of 76 CSP instances and 318 COP instances coming from the MiniZinc challenges 2012--2015. Moreover, the knowledge base mznc15 used in the MiniZinc Challenges 2016--2017 is also available. For more details, see the README file in /kb folder.

Additional info

Previous versions of SUNNY-CP supported solvers that are currently not included in the docker image due to compilation problems or the fact that are not publicly available/free. The old solvers that are not provided with the current default configuration are:

- [Mistral](#) (version does not print any output)
- [G12/Gurobi](#) (not free)
- [iZplus](#) (not publicly available)
- [Opturion](#) (not free)

We invite the developers interested in adding their solver to the default image of sunny-cp to contact us.

Authors

sunny-cp is developed by Roberto Amadini (University of Bologna) and Jacopo Mauro (University of Southern Denmark). For any question or information, please contact us: roberto.amadini at unibo.it mauro.jacopo at gmail.com

Acknowledgement

We would like to thank the staff of the former Optimization Research Group of NICTA (National ICT of Australia) for granting us the computational resources needed for building and testing sunny-cp. We also thank all the developers of the constituent solvers without which sunny-cp could not exist.

References

- [1] R. Amadini, M. Gabbrielli, and J. Mauro. SUNNY: a Lazy Portfolio Approach for Constraint Solving 2013. In ICLP, 2014.
- [2] R. Amadini, M. Gabbrielli, and J. Mauro. Portfolio Approaches for Constraint Optimization Problems. In LION, 2014.
- [3] R. Amadini, and P.J. Stuckey. Sequential Time Splitting and Bounds Communication for a Portfolio of Optimization Solvers. In CP, 2014.
- [4] R. Amadini, M. Gabbrielli, and J. Mauro. SUNNY-CP: a Sequential CP Portfolio Solver. In SAC, 2015.
- [5] R. Amadini, M. Gabbrielli, and J. Mauro. A Multicore Tool for Constraint Solving. In IJCAI, 2015.
- [6] MiniZinc Challenge webpage. <http://www.minizinc.org/challenge.html>