Naxos Solver

Nikolaos Pothitos

Department of Informatics and Telecommunications
National and Kapodistrian University of Athens
Panepistimiopolis, 157 84 Athens, Greece
pothitos@di.uoa.gr

- Naxos Solver is a C++ Constraint Programming library.
- It is freely available in the open source community together with its full documentation [3].
- It enforces a bounds-consistency variant in order to propagate changes that happen in Constraint Satisfaction Problem variables [1]. This happens in every single node of the search tree.
- It supports the definition of custom search methods via a declarative API [4].
- Declarativity is a Naxos key feature and it has do with describing problems and search methods independently. Naxos serves the Constraint Programming principle: *The user states the problem, the computer solves it* [2].
- Naxos is being used in various applications, such as the construction of the timetable for the Department of Informatics and Telecommunications of the University of Athens every semester [5].

Acknowledgments

I would like to thank Prof. Panagiotis Stamatopoulos for his inspiration to build the solver and his continuous supervision and scientific support. I also thank Foivos Theocharis for the search methods modules implementation.

References

[1] C. Bessière. Constraint propagation. In F. Rossi, P. van Beek, and T. Walsh, editors, *Handbook of Constraint Programming*, volume 2 of *Foundations of Artificial Intelligence*, chapter 3, pages 29–83. Elsevier, Amsterdam, 2006.

[2] E. C. Freuder and B. O'Sullivan. Grand challenges for constraint programming. *Constraints*, 19(2):150–162, 2014.

[3] N. Pothitos. Naxos Solver. `http://github.com/pothitos/naxos`, 2017.

[4] N. Pothitos and P. Stamatopoulos. Building search methods with self-confidence. `http://di.uoa.gr/~pothitos/Search_Methods.pdf`, Submitted, 2017.

[5] N. Pothitos, P. Stamatopoulos, and K. Zervoudakis. Course scheduling in an adjustable constraint propagation schema. In *ICTAI 2012: 24th IEEE International Conference on Tools with Artificial Intelligence*, volume 1, pages 335–343. IEEE, 2012.