BTD_12 and miniBTD_12

Djamal Habet[1], Philippe Jégou[1], Hélène Kanso[1,2], and Cyril Terrioux[1]

[1] Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
{djamal.habet, philippe.jegou, cyril.terrioux}@lis-lab.fr
[2] Effat University, Jeddah, Saudi Arabia
hkanso@effatuniversity.edu.sa

1 Solver description

BTD_12 and miniBTD_12 are written in C++ and both implement the algorithm BTD-MAC+RST+Merge [1]. This algorithm exploits the structure of CSP instances thanks to the notion of tree-decomposition [3].

For the competition, we have made the following choices:

- the tree-decompositions are computed thanks to the heuristic H_5-TD-WT [1]
- an heuristic based on the exponential recency weighted average [2,4] is exploited for ordering the variables inside a cluster,
- *lexico* is used as value ordering heuristic,
- generalized arc-consistency is enforced by a propagation-based system exploiting events,
- restarts are performed according to a geometric restart policy based on the number of backtrack with an initial cutoff set to 100 and an increasing factor set to 1.1,
- the first root cluster is the cluster having the maximum ratio number of constraints to its size minus one and then, at each restart, the selected root cluster is one which maximizes the sum of the weights of the constraints whose scope intersects the cluster.

Note that, at now, BTD_12 only takes into account the following constraints:

- `intension`,
- `extension`,
- `allDifferent` (the element `<except>` is not supported),
- `allEqual`,
- `ordered`,
- `sum`,
- `maximum` (the variant `<arg_max>` is not supported),
- `minimum` (the variant `<arg_min>` is not supported),
- `element`,
- `channel`,
- `noOverlap` (only the one dimensional form),
- `instantiation`.

2 Command line

BTD_12 and miniBTD_12 can be launched thanks to the following command line:

```
SOLVER 12 TIMELIMIT BENCHNAME
```

where:

– **SOLVER** is the path to the executable BTD or miniBTD,
– **TIMELIMIT** is the number of seconds allowed for solving the instance,
– **BENCHNAME** is the name of the XML file representing the instance we want to solve.

Acknowledgments

References

1. Jégou, P., Kanso, H., Terrioux, C.: Towards a Dynamic Decomposition of CSPs with Separators of Bounded Size. In: CP. pp. 298–315 (2016)
2. Liang, J.H., Ganesh, V., Poupart, P., Czarnecki, K.: Exponential recency weighted average branching heuristic for sat solvers. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. pp. 3434–3440. AAAI Press (2016)
3. Robertson, N., Seymour, P.: Graph minors II: Algorithmic aspects of treewidth. Algorithms 7, 309–322 (1986)
4. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA, USA, 1st edn. (1998)