BTD and miniBTD

Philippe Jégou, Hélène Kanso, and Cyril Terrioux

Aix Marseille Univ, Université de Toulon, CNRS, ENSAM, LSIS, Marseille, France
{philippe.jegou, hanan.kanso, cyril.terrioux}@lsis.org

1 Solver description

BTD and miniBTD are written in C++ and both implement the algorithm BTD-MAC+RST+Merge [2]. This algorithm exploits the structure of CSP instances thanks to the notion of tree-decomposition [4].

For the competition, we have made the following choices:

- the tree-decompositions are computed thanks to the heuristic H_5-TD-WT [2]
- *dom/wdeg* [1] is exploited for ordering the variables inside a cluster,
- *lexico* is used as value ordering heuristic,
- generalized arc-consistency is enforced by $AC3^{rm}$ in the preprocessing step and $AC8^{rm}$ during the search [3],
- 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 only takes into account the constraints given in extension or in intension and the following global constraints:

- `allDifferent`,
- `allEqual`,
- `element`,
- `sum`.

2 Command line

BTD and miniBTD can be launched thanks to the following command line:

```
SOLVER 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

This work has been partially funded by the french "Agence nationale de la Recherche", reference ANR-16-C40-0028.

References

1. Boussemart, F., Hemery, F., Lecoutre, C., Sais, L.: Boosting systematic search by weighting constraints. In: ECAI. pp. 146–150 (2004)
2. Jégou, P., Kanso, H., Terrioux, C.: Towards a Dynamic Decomposition of CSPs with Separators of Bounded Size. In: CP. pp. 298–315 (2016)
3. Lecoutre, C., Likitvivatanavong, C., Shannon, S., Yap, R., Zhang., Y.: Maintaining Arc Consistency with Multiple Residues. Constraint Programming Letters 2, 3–19 (2008)
4. Robertson, N., Seymour, P.: Graph minors II: Algorithmic aspects of treewidth. Algorithms 7, 309–322 (1986)