Solver description

Florian Fontan, Guillaume Verger

June 21, 2025

1 Outline

Both dominating set and hitting set instances are solved as unicost set covering instances.

The resolution follows 4 successive phases:

- Instance reduction
- Initial solution
- Large neighborhood search
- Local search

2 Instance reduction

The goal of this phase is to reduce the size of the instance to solve.

The reductions applied are:

- Mandatory sets removal
- Dominated sets removal
- Dominated elements removal
- Set folding (generalized from vertex cover)
- Unconfined sets reduction (generalized from vertex cover)
- Small components reduction

3 Initial solution

The goal of this phase is to get an initial solution for the next phases.

4 greedy algorithms are used.

Classical greedy This is the classical greedy algorithm for the set covering problem. While not all elements are covered, the set covering the highest number of uncovered elements is added to the solution.

Greedy algorithm from [2] This greedy algorithm is similar to the classical greedy, but use a different set scoring function.

Dual greedy In this greedy algorithm, while not all elements are covered, an uncovered element is selected and a set covering this element is added to the solution.

Reverse greedy This greedy algorithm starts with a solution containing all sets. While the solution contains redundant sets, a redundant set is removed.

The first two greedy algorithms work well when the number of sets in the solution is small compared to the total number of sets. On the other hand, the reverse greedy works well when most sets are in the solution.

To determine which greedy algorithm to run, we asses the number of sets in the solution by running the dual greedy algorithm. The dual greedy is always very fast, even if, in general, it yields worse solutions than the other greedy algorithms.

4 Large neighborhood search

The goal of this step is to very quickly improve the solution from the previous phase.

To this end, a large neighborhood search is used. This large neighborhood search works as follows: at each iteration:

- The set with the worst score is removed from the solution
- The score of the uncovered elements is incremented
- While not all elements are covered, the set with the best score is added to the solution and while the solution contains redundant sets, a redundant set is removed

5 Local search

The goal of this last phase is to improve the current solution to get the best possible solution.

It implements a local search with a row weighting scheme similar to the one proposed by [2].

When the number of set neighbors is to high so that computing them make the solver hit the memory limit, it runs the local search with row weighting from [1] instead.

References

[1] Chao Gao, Xin Yao, Thomas Weise, and Jinlong Li. An efficient local search heuristic with row weighting for the unicost set covering problem. *European Journal of Operational Research*, 246(3):750–761, November 2015.

[2] Weibo Lin, Fuda Ma, Zhouxing Su, Qingyun Zhang, Chumin Li, and Zhipeng Lü. Weighting-based parallel local search for optimal camera placement and unicost set covering. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, GECCO '20, pages 3–4, Cancún, Mexico, July 2020. Association for Computing Machinery.