Discrete Optimization Specialization: Workshop 9
Preparing Medicinal Herbs

1 Introduction

Shennong has resolved the curse on the village so that no new boys and girls will become sick. He now needs to prepare herbs to heal the children who are already sick. He will need to arrange the preparation of many herbs such as: ginseng, poria, angelica, etc. Each preparation can involve a number of steps: cutting, washing, soaking, brushing, bleaching, frying, simmering, baking, cooking and steaming. Sometimes the order of the steps is important: for example cutting ginseng should only occur after it is bleached. Each of the steps can be done with different tools. For example one can cut with a pottery knife, an iron knife or a gold knife. Only one process can use a tool at the same time. For example cutting ginseng and cutting angelica both require the gold knife so they cannot happen at the same time. The aim is to arrange a schedule for preparing all the herbs in the minimal time which satisfies the constraints

Preparing Herbs — `prepare.mzn`

The data for the problem is encoded as, a number of tasks that need to be performed, such as cutting ginseng, bleaching ginseng, and cutting angelica, with their durations; a number of precedences among tasks, and list of sets of tasks requiring the same tool that cannot be performed simultaneously.

An example data file is

```
n = 8;    % 8 tasks
d = [1,5,6,4,9,4,8,7];

m = 6;    % 6 precedences
pre = [1, 5, 8, 5, 7, 8];
post= [4, 6, 7, 4, 3, 3];

o = 2;    % required tools
disj= [{1,3,5,7}, {4,5,6,8}];
```

We are given a model for the problem `prepare.mzn`. The aim of this workshop is to find the best programmed search strategy you can for the given data files `prepare1.dzn` – `prepare14.dzn`. You should not modify the model in any way except to add a search annotation to the `solve` item.

Experiment with which variables to search over s, e or b and the variable selections: `input_order`, `first_fail`, `smallest`, `largest`, and value selections: `indomain_min`, `indomain_max`, `indomain_median`, `indomain_random`, `indomain_split`, `indomain_reverse_split`.

Use the `-s` statistics flag (or check box in the IDE) to compare how much search each search strategy uses. Use the command line `--fzn-flags '--time X'` (or the solver flags check box in the IDE) to limit the execution time to X milliseconds.

Compare strategies in terms of

- *solving*: how many failures to find a first solution (note you can test this by changing the solve item to `solve satisfy`)
- *completeness*: how often it proves the optimal solution (in the time you are prepared to wait)
- *search-space*: how many failures to prove the optimal solution
- *first-solution*: how good the first solution is
- *robustness*: how good a solution it finds in some fixed time (1 minute say)

Note that data files from 7–14 are considerably harder than 1–6. What do you think is the best search strategy overall? Justify your answer.

Preparing Herbs Again — `prepare_disj.mzn`

A model for the same problem making use of the `disjunctive` global constraint is given in file `prepare_disj.mzn`. The b variables are no longer part of this model.

How does this model change the tradeoff for the search strategy compared to the previous model? Is the best strategy for the previous model still the best for this model?

2 Technical Requirements

For completing the workshop you will need MiniZinc 2.1.x (`http://www.minizinc.org/software.html`).