ASP Challenge Problem: Fastfood Problem

Jori Bomanson[1] and Tomi Janhunen[2]
1 Department of Computer Science
Aalto University
P.O. Box 15400, FI-00076 AALTO, Finland
2 Information Technology and Communication Sciences
Tampere University
FI-33014 TAMPERE UNIVERSITY, Finland

1 Problem Description

This is a problem where there are n restaurants along a single road and the task is to pick a k-subset of the restaurants as depots. This is to be done in a way that minimizes the sum of distances along the road from each restaurant to the depot closest to it.

The idea behind the problem is that a single company owns a fixed set of restaurants along a road and the company has the funds to establish a fixed number k of depots at chosen restaurant locations. These depots will be used to store goods intended for use in the restaurants. The goal is to make this decision optimally in the way that those goods can be transferred to restaurants as easily as possible.

2 Input Description

The input consists of facts of the following form:

- `restaurant(id, position)`, which represents a restaurant, where `id` is a unique identifier of the restaurant and `position` represents a dimensional positional along the road.
- `number_depots(num)`, which indicates the total number of different restaurants.

3 Output description

A solution contains atoms of the form:

- `depot(id, position)`, representing a subset of the arguments of the restaurant predicate. These mark the restaurant locations where depots are to be established.

4 Example

Below is an example input-output pair. The optimal output shown is unique in this specific example.

- **Input**

```
restaurant(a,1).
restaurant(b,2).
restaurant(c,7).
restaurant(d,100).
number_depots(2).
```

- **Optimal output**

```
depot(b,2)
depot(d,100)
```

5 Sorting schema

Given a solver s and an instance i, the score of s on the instance i, denotes as $score(s, i)$, is computed as follows:

- $score(s, i) = 1.5$ if s proves the optimality of the solution
- $score(s, i) = 0$ if s found no solution
- $score(s, i) = \frac{cost(sbest,i)+1}{cost(s,i)+1}$

where $cost(s, i)$ is the cost of the solution printed by the solver s on the instance i and $cost(sbest, i)$ is the best cost printed on the instance i. The score is rounded off to three decimal digits. The overall score of a solver s is computed as the sum of the scores of s for each instance i. The solver with the maximum score is the winner.

Example. Let s_1, s_2, s_3 and s_4 be solvers and let i be an instance. Assume the following costs:

$$cost(s_1, i) = 100\ (opt) \quad cost(s_2, i) = 100 \quad cost(s_3, i) = 200 \quad cost(s_4, i) = 400$$

Then, their scores are the following:

$$score(s_1, i) = 1.5 \quad score(s_2, i) = 1 \quad score(s_3, i) = 0.502 \quad score(s_4, i) = 0.252$$

6 References

This is an optimization version of a decision benchmark from the ASP Competition 2011, and namely the Fastfood Optimality Check benchmark authored by Wolfgang Faber which is available at `https://www.mat.unical.it/aspcomp2011/FinalProblemDescriptions/FastfoodOptimalityCheck`. That benchmark is further based on the "Fast Food" problem, number 662 of volume VI of the ACM programming contests problem set archive (http://acm.uva.es/p/v6/662.html). The instances here have been generated anew to be harder than previously.