Optimize a Data Center

Artificial Intelligence 21/22

Beatriz Aguiar, João Marinho & Margarida Vieira

Problem Specification

Given a schema of a data center and a list of available servers, the goal is to assign servers to slots within the **rows** and to logical **pools** so that the **lowest guaranteed capacity** of all pools is **maximized**.

Servers are **physically** divided into **rows**. Rows share resources, therefore if a resource fails in a row, all servers in that row become unavailable.

Servers are also **logically** divided into **pools**. Each server belongs to exactly one pool, and provides it with some amount of computing resources, called capacity, c. The capacity of a pool is the sum of the capacities, c_i, of the available servers in that pool.

To ensure reliability of a pool, it is therefore desirable to distribute its servers between different rows. The **guaranteed capacity** of a pool is the minimum capacity it will have when at most one data center row goes down.

Input data

R - number of rows in the data center

S - number of slots in each row of the data center

U - number of unavailable slots

P - number of pools to be created

M - number of servers to be allocated

Output data

OR - number of the row in the output file

OS - number of the slot in the output file

OP - number of the pool in the output file

X - server not allocated

Objective Function

$$for\ i\ (0\ \leq i < P), the\ guaranteed\ capacity, gc_i, can\ be\ defined\ as$$

$$gc_i = min_{0 \leq i \leq P}\left(\sum_{k=0, server\ k\ in\ pool\ i, server\ k\ in\ row\ r}^{M-1} c_i\right) - \sum_{k=0, server\ k\ in\ pool\ i, server\ k\ in\ row\ r}^{M-1} gc_i$$

$$f(x) = min_{0 \leq i \leq P} gc_i$$

Optimization Problem

1 Neighbourhood and crossover functions

Neighbourhood function

- Change pool
- Change slot
- Change row
- Deallocate server

Crossover function

- Mix two parent solutions

2 Evaluation function

Given our goal is to maximize the guaranteed capacity, our evaluation function will calculate the maximum guaranteed capacity of the solution, i.e., the maximum value for all the possible lowest guaranteed capacities of the data center.

3 Solution representation

2 5 1 2 5	2 rows of 5 slots each, 1 slot unavailable, 2 pools and 5 servers.
0 0	Coordinates of the first and only unavailable slot.
3 10	First server takes three slots and has a capacity of 10.
3 10	So does the second one.
2 5	The third one takes two slots and has a capacity of 5.
1 5	The fourth one takes just one slot and has a capacity of 5.
1 1	The fifth one takes just one slot too and has a capacity of 1.

4 Hard constraints

- Each slot of the data center must be occupied by at most one server
- No server occupies any unavailable slot of the data center
- No server extends beyond the slots of the row
- Each server belongs to exactly one pool and one row

Implementation work

Data structures

- Python classes for the various problem entities
  - **DataCenter**, main class that contains all global data – servers, rows, pools
  - **Server**, with size, capacity and associated pool
  - **Row**, with list where –1 is empty, -2 is unavailable and x, x >= 0, represents the number of the server
- Lists for storing the problem domain and the solution

Programming Language

IDE