FIT5216: Modelling Discrete Optimization Problems
Inclass Task 3: PowerGen

1 Problem Statement

The problem is to plan the long term building of power generation to meet future power needs. There are three types of power plants that can be built:

- **Nuclear** plants cost 10B, last 60 years, and generate 4GW.
- **Coal** plants cost 1B, last 20 years, and generate 1GW.
- **Solar** plants cost 2B, last 30 years, and generate 1GW.

You are given data for the expected power needs in GW and the current generation capacity for each of the next T decades. Data file is of the form

```
int: T; % number of decades
array[1..T] of int: e;    % expected power needs in GW
array[1..T] of int: a;    % existing generation capacity in GW
```

For example, the data (`powergen.dzn`) we will work on is

```
T = 10;
e = [25, 25, 30, 25, 20, 20, 15, 15, 15, 12];
a = [18, 15, 12, 8, 4, 3, 2, 0, 0, 0];
```

indicating 10 decades of planning.

The problem is to decide how many power plants of each type to build in each decade. These decisions are represented by

```
array[1..T] of var 0..infinity: N; % number of nuclear power plants built each decade
array[1..T] of var 0..infinity: C; % number of coal power plants built each decade
array[1..T] of var 0..infinity: S; % number of solar power plants built each decade
```

The constraints of the problem are:

- the energy available in each decade must meet the expected demand
- no more than 40% of all energy can be nuclear in any decade
- no less than 20% of all energy can be solar in each decade

The aim is to find the minimal cost solution.

2 Preliminary Modelling

Before we solve the problem lets do some preliminary work. In order to satisfy the constraints we will need to know how much energy in each decade comes from coal, nuclear and solar.

Lets assume we have made the decisions about how many plants to build, lets build a model to compute how much energy comes from each type in each decade. The model `prelim.mzn` takes these decisions as fixed data

```
array[1..T] of 0..infinity: N; % number of nuclear power plants built each decade
array[1..T] of 0..infinity: C; % number of coal power plants built each decade
array[1..T] of 0..infinity: S; % number of solar power plants built each decade
```

We need to add constraints to compute

```
array[1..T] of var 0..infinity: Nenergy; % nuclear power available in each decade
array[1..T] of var 0..infinity: Cenergy; % coal power available in each decade
array[1..T] of var 0..infinity: Senergy; % solar power available in each decade
```

Since a nuclear plant last 6 decades, then e.g. in decade 4 we get power from plants we build in decades 1,2,3,4; in decade 9 we get power from plants we build in decades 4,5,6,7,8,9, etc. Coal plants only last 2 decades so we get power from plants built in the current decade and the previous one (if it exists, e.g. for coal generation in decade 1, it only comes from plants built in decade 1). Solar plants last 3 decades so we get power from plants in the current decade and the previous two (if they exist). Complete the model `prelim.mzn` to check you have these calculations right (with the checker provided) on the three data files. Once that works go on to the next section.

3 Power generation decisions

Build a MiniZinc model `powergen.mzn` which solves the problem. You should write the model to read a data file, although we will only used the supplied file `powergen.dzn`. You should compute the objective, that is the sum of costs in a variable `cost` so that the grader can show you how good your solution is. Your aim is to minimize cost. You can use the calculation code you devised for `prelim.mzn` to help build the model, now working on the unfixed decisions about building plants.

Once this works add another constraint:

- No more than 10 coal power plant decades over the plan, that is if we sum up the number of coal power plants in action in any decade, for all decades the sum is no more than 10.

What is the cost of trying to avoid putting too much carbon in the atmostphere?

4 Instructions

Edit the provided `mzn` model files to solve the problems described above. Your implementations can be tested locally by using the *Run* icon in the MiniZinc IDE or by using,

```
minizinc ./modelname.mzn ./datafile.dzn
```

at the command line.