

## CS4626 Special Topics in Computing: Constraint Programming and Optimisation

### 1<sup>st</sup> Continuous Assessment 2019/20

The Pioneer lander will be dropped onto a passing comet to conduct some science experiments, and then beam the results back to the parent spacecraft. Because it sources its energy from sunlight, the Pathfinder has a limited time to run the experiments. Experiments are divided into different types, and each type requires a fixed amount of time. Each type has an associated value -- i.e. the value of the information the scientists hope to gain. The number of possible experiments of each type will depend on exactly where it lands, and will be determined by the Pioneer from initial sensing. There are some other restrictions on the set of experiments - for example, some experiments provide necessary input for others, so there must be at least as many of the first type compared to the second. The problem is for the Pathfinder to decide how many experiments of each type it should run, to maximise the total sum of value from the individual experiments.

Problem instances (as generated by the Pioneer) are described in a standard format in text files:

```
n t b
t0      v0      m0
t1      v1      m1
...
t(n-1)  v(n-1)  m(n-1)
e00     e01
e10     e11
...
e(b-1)0 e(b-1)1
```

where  $n$  is the number of possible experiment types,  $t$  is the total number of hours available, and  $b$  is the number of 'before' relationships between pairs of experiment types. There is then a series of  $n$  lines ( $t_i \ v_i \ m_i$ ), where  $t_i$  is the time required for an experiment of type  $i$ ,  $v_i$  is the scientific value of experiments of type  $i$ , and  $m_i$  is the maximum number of experiments of type  $i$  possible in this location. There is then a series of  $b$  lines each specifying a pair of experiment types, where there must be at least as many of type  $e_{i0}$  as  $e_{i1}$

.So for the instance described in pioneer0.txt,

```
4 21 1
3 2 6
5 5 3
4 7 1
2 1 8
1 2
```

there are 4 experiment types, with up to 21 hours available. Experiments of type 0 take 3 hours, with value 2, and there are 6 of them; expt 1 takes 5 hours, with value 5, and there are 3 of them, etc.. Finally, there must be at least as many experiments of type 1 as of type 2. The optimal solution gives a total value of 23, occupying 21 hours

A class `PioneerData.java` is provided, which will read the problem instance from a data file, and offers the following public getter methods:

```
public int getNumTypes();
public int getTotalHours();
public int getNumBeforees();
public int[] getHours();
public int[] getValues();
public int[] getTotals();
public int[][] getBeforees();
public int getMaxHours();
public int getMaxValues();
public int getMaxTotals();
```

### The assignment

Create a Choco class for modeling and solving problems as described above. Your class must be called "PioneerXYZ.java", where XYZ is replaced with your student ID number. It must have an executable main method (which should throw an IOException), and it must read in data from text files in the above format. Your solution should find the optimal solution to the problem instance. Do not include the PioneerData.java class in your submission, but do include any other Java classes that you need to get the solution. Make sure you use meaningful variable names, and include comments to explain your model.

### Submission

Your java class (as a .java file) must be emailed to Ken Brown (k.brown@cs.ucc.ie) by the submission deadline: **1pm, Wednesday 5<sup>th</sup> February, 2020.**

### Marking scheme

There are 5 marks available for this assignment. Marks will be awarded for submitting a working Java program which generates solutions using the Choco solver, for generating correct solutions, for the quality of model, and for good use of Choco facilities including global constraints if needed. Marks will not be awarded for the quality of the Java code that supports the model, so don't waste time implementing a GUI to show the results (although if your code is inadequately commented and is difficult to understand, you may lose marks).

### Plagiarism

This submission is part of the formal assessment for the module, and so must be your own work. You should not submit anyone else's work as if it were your own, and you should not be collaborating with each other on how to solve the problem. However, you are free to re-use, without acknowledgement, any piece of Java/Choco code that has been posted on the CS4626 web pages or included in CS4626 Lecture slides, or included in the Choco manual.