

1 DataZinc Format Description

This document describes the formatting of one of the DataZinc (dzn) files that store all the data required to fully specify an instance of the MSPSP.

Each dzn file contains the following:

- **seed**: Seed used for the randomisation. Integer between 1,000,000 and 9,999,999.
- **mint**: Lower bound on the total project duration. Naïvely calculated using the length of the critical path.
- **maxt**: Upper bound on the total project duration. Naïvely calculated using as the sum of all the processing times.
- **nActs**: Total number of activities. Includes the dummy start and end activities. Note, the n value in an instance's file name does not include the dummy start and end activities.
- **dur**: Duration/processing time of each activity. Stored as an $1 \times n$ array.
- **nSkills**: Total number of skills.
- **sreq**: Skill requirement for each activity. Stored as an $n \times l$ array. If entry $(i, j) = 0$ then activity i does not require any resources mastering skill j .
- **nResources**: Total number of resources.
- **mastery**: Skills each resource has mastered. Stored as an $m \times l$ array containing either **true** or **false**.
- **nPrecs**: Total number of precedence relations. Define this as $(|E|)$.
- **pred**: A $1 \times |E|$ array storing the predecessor for each precedence relation.
- **succ**: A $1 \times |E|$ array storing the successor for each precedence relation.

The following items are pieces of data that can be derived from the basic data specified above. We decided to compute these in a preprocessing step and store it in the data file to improve the solve times.

- **nUnrels**: Total number of unrelated activity pairs. Define this as $(|U|)$.
- **unpred**: A $1 \times |U|$ array storing the first activity for each unrelated pair.

- **unsucc**: A $1 \times |U|$ array storing the second activity for each unrelated pair.
- **USEFUL_RES**: A $n \times 1$ array storing the set of useful resources for each activity. I.e. the set of all resources mastering a skill this activities requires.
- **POTENTIAL_ACT**: A $m \times 1$ array storing the set of potential activities for each resource. I.e. the set of all activities that each resource could feasibly be assigned.

Lastly, the instances of set 2 and 3 also store one extra piece of derived data:

- **SumOfsreq**: Sum of the skill requirements of all activities. This can be useful for doing analysis on the complexity of an instance.

We now give a simple example of a DataZinc file with 8 activities, 2 skills and 7 resources. Note, that the n value in the file name of the instance does not include the dummy start and end activities, whereas the n value within the dzn file (i.e. **nActs**) does include these 2 dummy activities. Hence, these numbers differ by 2.

File path = /instances/set-3/set-3a/

File name = inst_set3a_sf0_nc1.8_n6_l2_m7_00.dzn

Contents:

```
% seed = 0

mint = 14;
% maxt = 24;

nActs = 8;
dur = [0,2,4,2,8,4,4,0];

nSkills = 2;
sreq = [| 0,0,
        | 0,3,
        | 1,2,
        | 2,0,
        | 1,0,
        | 3,0,
        | 1,2,
        | 0,0, |];

% SumOfsreq = 15;

nResources = 7;
mastery = [| true,true,
            | false,true,
            | false,true,
            | true,false,
            | true,false,
            | true,true,
            | false,true, |];

nPrecs = 11;
```

```

pred = [1,2,2,2,3,3,4,4,5,6,7];
succ = [2,3,5,6,4,6,6,7,7,8,8];

nUnrels = 4;
unpred = [3,4,5,6];
unsucc = [5,5,6,7];

USEFUL_RES = [{},
  {1,2,3,6,7},
  {1,2,3,4,5,6,7},
  {1,4,5,6},
  {1,4,5,6},
  {1,4,5,6},
  {1,2,3,4,5,6,7},
  {}];

POTENTIAL_ACT = [{2,3,4,5,6,7},
  {2,3,7},
  {2,3,7},
  {3,4,5,6,7},
  {3,4,5,6,7},
  {2,3,4,5,6,7},
  {2,3,7}];

```