Industrial Modelling Competition at CP 2015

B. Hurley, M. Mossige, H. Simonis

August 31, 2015

This information can also be found at the conference website with the URL `http://booleconferences.ucc.ie/indmodellingcomp`.

As part of Industrial Day on Thursday, we propose a modelling competition for the participants of CP2015 and ICLP2015. We will have a session on Thursday afternoon (13:45-15:00 in room G08) to discuss progress, and will add the problems and suggested models to the CSPLib on Saturday in the CSPLib sprint event.

The problem has kindly been provided by Morten Mossige from ABB in Norway.

The problem arises in the context of a testing facility. A number of tests have to be performed in minimal time. Each test has a given duration and needs to run on one machine. While the test is running on a machine, no other test can use that machine. Some tests can only be assigned to a subset of the machines, for others you can use any available machine. For some tests, additional, possibly more than one, global resources are needed. While those resources are used for a test, no other test can use the resource. The objective is to finish the set of all tests as quickly as possible, i.e. all start times should be non-negative, and makespan should be minimized. The makespan is the difference between the start of the earliest test, and the end of the latest finishing test.

1 To participate

Submission is by email, send result files to `cpmodellingcomp@gmail.com` with the problem name as the subject. Any solver is allowed, teams of up to three participants are allowed to work together. Nobody should participate in more than one team.

2 Details of format

The input data is given as a set of Prolog facts. An example is shown below:

```
%% ****  Testsuite  ****
% Number of tests                  : 40
% Number of machines               : 10
% Number of resources              : 3
% Number of families               : 1
% Prob that a test use a resource  : 30%
```

```
% Minimum test duration            : 1
% Maximim test duration            : 800
% MPri                             : 40%

test( 't1', 568, [], [], 'fam1', 1 ).
test( 't2', 669, [], ['r3'], 'fam1', 1 ).
test( 't3', 688, [], [], 'fam1', 1 ).
test( 't4', 709, ['m7'], [], 'fam1', 1 ).
test( 't5', 505, ['m3'], [], 'fam1', 1 ).
test( 't6', 683, [], [], 'fam1', 1 ).
test( 't7', 20, [], [], 'fam1', 1 ).
test( 't8', 250, ['m6'], [], 'fam1', 1 ).
test( 't9', 235, [], [], 'fam1', 1 ).
test( 't10', 651, [], [], 'fam1', 1 ).
test( 't11', 477, ['m10'], [], 'fam1', 1 ).
test( 't12', 227, [], [], 'fam1', 1 ).
test( 't13', 464, [], [], 'fam1', 1 ).
test( 't14', 257, [], [], 'fam1', 1 ).
test( 't15', 80, ['m2'], [], 'fam1', 1 ).
test( 't16', 592, [], [], 'fam1', 1 ).
test( 't17', 376, ['m9'], [], 'fam1', 1 ).
test( 't18', 541, [], [], 'fam1', 1 ).
test( 't19', 215, [], [], 'fam1', 1 ).
test( 't20', 645, [], [], 'fam1', 1 ).
test( 't21', 624, [], ['r2'], 'fam1', 1 ).
test( 't22', 463, ['m1'], [], 'fam1', 1 ).
test( 't23', 549, [], [], 'fam1', 1 ).
test( 't24', 647, [], ['r1'], 'fam1', 1 ).
test( 't25', 361, [], [], 'fam1', 1 ).
test( 't26', 57, [], ['r3'], 'fam1', 1 ).
test( 't27', 422, [], [], 'fam1', 1 ).
test( 't28', 530, ['m1'], [], 'fam1', 1 ).
test( 't29', 492, [], ['r2'], 'fam1', 1 ).
test( 't30', 306, ['m9','m4','m8'], ['r3'], 'fam1', 1 ).
test( 't31', 519, [], [], 'fam1', 1 ).
test( 't32', 176, [], [], 'fam1', 1 ).
test( 't33', 354, [], ['r1','r2','r3'], 'fam1', 1 ).
test( 't34', 682, [], [], 'fam1', 1 ).
test( 't35', 428, [], [], 'fam1', 1 ).
test( 't36', 340, [], [], 'fam1', 1 ).
test( 't37', 119, ['m1','m6','m4'], ['r3','r2'], 'fam1', 1 ).
test( 't38', 791, [], [], 'fam1', 1 ).
test( 't39', 167, ['m4','m1','m9'], [], 'fam1', 1 ).
test( 't40', 363, [], [], 'fam1', 1 ).

embedded_board( 'm1').
embedded_board( 'm2').
embedded_board( 'm3').
embedded_board( 'm4').
embedded_board( 'm5').
```

```
embedded_board( 'm6').
embedded_board( 'm7').
embedded_board( 'm8').
embedded_board( 'm9').
embedded_board( 'm10').

testsetup( 'fam1', 0 ).

resource( 'r1', 1).
resource( 'r2', 1).
resource( 'r3', 1).
```

The format is the following:
Comment lines starting with % are ignored.
Naming convention of a test suite (the files):

```
t<number of test cases>m<number of machines>r<number of resources>-<instance no>.pl
```

e.g.

```
t20m10r3-18.pl
```

20 test case, 10 machines, 3 resources, instance number 18
At the moment four test problem are given, more may be added depending on results.
Each test suite is currently encoded as Prolog predicates as:

```
test( <name>, <duration>,
        <possible machine (empty list = all machines)> ,
        <requested resources>,
        <family of test case (not used)>,
        <priority(not used)>
).
```

e.g.

```
test( 't13', 537, [], ['r2','r1'], 'fam1', 1 ).
```

Name:t13, duration 537, executable on any machine, requesting resource r2 and r1
An empty list for the machine indicates any machine can be used, an empty list for the resources indicates that no additional resources are required.
A machine:

```
embedded_board( <name of machine>).
```

e.g.

```
embedded_board( 'm8').
```

A resource:

```
resource( <name>,<capacity>).
```

e.g.

```
resource( 'r1', 1).
```

Name: r1, capacity: 1 (currently we only use 1 as capacity, to be changed in future.)

The results should be given in the form:

```
result(<name>,<start>,<machine>).
```

e.g.

```
result('t1',0,'m1').
```

Task t1 starts at time 0 on machine m1.

3 Example

An example problem and its solution is shown in the following Figure:

Test	Duration	Executable on	Use of global resource
t_1	2	m_1, m_2, m_3	-
t_2	4	m_1, m_2, m_3	r_1
t_3	3	m_1, m_2, m_3	r_1
t_4	4	m_1, m_2, m_3	r_1
t_5	3	m_1, m_2, m_3	-
t_6	2	m_1, m_2, m_3	-
t_7	1	m_1	-
t_8	2	m_2	-
t_9	3	m_3	-
t_{10}	5	m_1, m_3	-