Discrete Optimization Specialization: Workshop 7
Visiting Zhuge Liang

1   Introduction

Liu Bei is very keen to enlist Zhuge Liang's help in fighting Cao Cao. Unfortunately his first two visits to Zhuge Liang did not succeed. Now Liu Bei is desperate and hence decides to prepare very well for the next visit. Liu Bei's strategy is to meet all of Zhuge Liang's closest associates (including his friends, the governor and scholars Zhuge trusted, and even Zhuge's wife and father-in-law). The purposes of the meetings are (a) to get to know Zhuge's background and habits more and (b) to ask these people to help persuade Zhuge to join Liu Bei.

Scheduling a Visit to Zhuge Liang — `visit_zhuge.mzn`

Lie Bei must schedule each visit to the associates of Zhuge Liang and the final visit to Zhuge Liang. Each visit will take a number of days, and some of his associates will not accept visits over weekends, and visits cannot be interrupted. There are also some precedences. Liu Bei will need to have completed a visit to some people, in order to gain entrance to see other people. Each person to be visited also has a rank in the society.

   The aim is to complete all the visits as soon as possible, but also to smooth the visits Liu Bei should try to avoid visiting people of higher rank before people of lower rank; otherwise the higher ranked person may think that Liu Bei does not treat them properly according to their rank. The most important thing is to minimize the end time of the visits, and secondly to minimize the number of rank violations.

   The data and decisions for problem are hence

```
enum PERSON; % set of people to be visited

array[PERSON] of int: duration;    % duration of visit
array[PERSON] of bool: on_weekend; % can the person be visited over weekends
array[PERSON] of int: rank;        % rank of the person in society

array[int,1..2] of PERSON: prec;   % precedences between visits
set of int: PREC = index_set_1of2(prec);

0..6: starting_day; % which day of the week is day 0 in the schedule
                    % 0 Monday, 1 Tuesday, ..., 5 Saturday, 6 Sunday

% upper bound on the amount of time needed: total duration plus 1 week
% per person who cant be visited on the weekend
int: total = sum(duration) + 7*sum(p in PERSON)(on_weekend[p] = false);
set of int: TIME = 0..total;

array[PERSON] of var TIME: start;
```

```
var TIME: end;
```

An example data file is

```
PERSON = {CuiZhouping, MengJian, ShiTao, LiuBiao, SimaHui, PangDegong,
              HuangChengyan, HuangYueying, ZhugeLiang};

duration = [ 2, 4, 3, 5, 3, 2, 3, 4, 5 ];
on_weekend = [ true, true, false, false, false, true, true, false, false ];
rank = [ 1, 1, 1, 4, 3, 3, 3, 2, 5 ];

prec = [| LiuBiao, SimaHui
        | ShiTao, PangDegong
        | LiuBiao, HuangChengyan
        | HuangChengyan, HuangYueying
        | CuiZhouping, ZhugeLiang
        | MengJian, ZhugeLiang
        | SimaHui, ZhugeLiang
        | PangDegong, ZhugeLiang
        | HuangYueying, ZhugeLiang |];

starting_day = 3;
```

Notice how all people must be visited before Zhuge Liang, either by a direct or indirect (transitive) precedence relation, and the precedences may conflict with rank.

A solution that could be returned is

```
start = [16, 0, 25, 4, 18, 28, 9, 12, 32];
end = 37;
rank_violations = 12;
```

The schedule starts on Thursday, and respects the weekend constraints. An illustration of the schedule is shown below, showing the days of the week, the time according to the schedule, and which person number is being visited on each day, or "." for nobody. Notice how the weekend constraints force Liu Bei to waste some days.

```
Day:   M T W T F S S M T W T F S S M T W T F S S M T W T F S S
Time:                          1 1 1 1 1 1 1 1 1 1 2 2 2 2 2
             0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
Visit:       2222222|444444444|77777|8888888|111|555555|......|

M T W T F S S M T W T F S S
2 2 2 2 2 3 3 3 3 3 3 3 3 3
5 6 7 8 9 0 1 2 3 4 5 6 7 8
33333|66666|.|999999999|....
```

A better solution is

```
start = [4, 0, 6, 11, 25, 9, 16, 19, 32];
end = 37;
rank_violation = 5;
```

The end time is the same, but the rank violations are much reduced. The only rank violations in this schedule are Liu Biao (4) is visited before each of Sima Hui (5), Huang Chengyan (7) and Huang Yueying (8), and both Pang Degong (6) and Huang Chengyan (7) are visited before Huan Yueying (8).

```
Day:   M T W T F S S M T W T F S S M T W T F S S M T W T F S S
Time:                           1 1 1 1 1 1 1 1 1 1 2 2 2 2 2
           0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
Visit:       2222222|111|33333|666|444444444|77777|8888888|...|

M T W T F S S M T W T F S S
2 2 2 2 2 3 3 3 3 3 3 3 3 3
5 6 7 8 9 0 1 2 3 4 5 6 7 8
55555|.......|999999999|....
```

If you want an extra challenge, modify your model to print out the schedule as shown in the examples (on only 4 lines, since the line breaks are added just to fit the page width). But note this is probably more work than the rest of the workshop.

2   Technical Requirements

For completing the workshop you will need MiniZinc 2.2.x (http://www.minizinc.org/software.html).