

intern_v7.2.mod

```

1 /*****
2  * OPL 12.8.0.0 Model
3  * Author: chirs + kipp
4  * Creation Date: 19 Mar 2019 at 5:03:48 pm
5  *****/
6
7 //restricted first MIC rotation to 1..25
8
9 range Week = 1..54;
10 range Intern = 1..11;
11 range Rotation = 1..17;
12
13 dvar boolean x[Intern][Rotation][Week];
14
15 dvar boolean D[Week];
16
17 dvar boolean y1[Intern][Week];
18 dvar boolean y2[Intern][Week];
19 dvar boolean y3[Intern][Week];
20 dvar boolean y4[Intern][Week];
21 dvar boolean y5[Intern][Week];
22 dvar boolean y6[Intern][Week];
23 dvar boolean y7[Intern][Week];
24 dvar boolean y8[Intern][Week];
25 dvar boolean y9[Intern][Week];
26 dvar boolean y11[Intern][Week];
27
28 dvar boolean L1[Week];
29 dvar boolean L2[Week];
30 dvar boolean L3[Week];
31
32 int M = 1000;
33
34 dexpr int z = sum(i in Intern, j in Rotation, k in Week) x[i][j][k];
35
36 minimize z;
37
38
39 subject to{
40
41 //let Rotation 17 be Weeks 1-4 when Interns 7-11 have not yet begun and
42 // Weeks 51-54 when Intern 1-6 have finished
43
44 forall(i in 1..6)
45 sum(k in 51..54)x[i][17][k] ==4;
46
47 forall(i in 7..11)
48 sum(k in 1..4)x[i][17][k] ==4;
49
50 forall(i in 1..6)
51 sum(k in 1..50)x[i][17][k] ==0;
52
53 forall(i in 7..11)
54 sum(k in 5..54)x[i][17][k] ==0;
55
56 //Orientation Constraints
57

```

```

58 forall(i in 1..6)
59   sum(k in 1..4)x[i][5][k] ==1;
60
61 forall(i in 7..11)
62   sum(k in 5..8)x[i][5][k] ==1;
63
64 forall(i in 1..6)
65   sum(k in 1..4)x[i][9][k] ==1;
66
67 forall(i in 7..11)
68   sum(k in 5..8)x[i][9][k] ==1;
69
70 forall(i in 1..6)
71   sum(k in 1..4)x[i][10][k] ==2;
72
73 forall(i in 7..11)
74   sum(k in 5..8)x[i][10][k] ==2;
75
76 ///Intern Physical Constraint (can only be in one place at a time)
77 forall(i in Intern, k in Week)
78   sum(j in Rotation)x[i][j][k] <= 1;
79
80 ///Intern Rotation Completion Constraint (everyone must do one of the rotations)
81
82 //CPD-G (j=1)
83 forall(i in Intern)
84   sum(k in Week)x[i][1][k] == 8;
85
86 //CPD-V (j=2)
87 forall(i in Intern)
88   sum(k in Week)x[i][2][k] == 4;
89
90 //AP (j=3)
91 forall(i in Intern)
92   sum(k in Week)x[i][3][k] == 4;
93
94 //MIC (j=4)
95 forall(i in Intern)
96   sum(k in 1..49)x[i][4][k] == 4;
97
98 //MCH (j=5)
99 forall(i in Intern)
100   sum(k in 1..30)x[i][5][k] == 3;
101
102 //CPCa (j=6)
103 forall(i in Intern)
104   sum(k in Week)x[i][6][k] == 3;
105
106 //CPM (j=7)
107 forall(i in Intern)
108   sum(k in Week)x[i][7][k] == 3;
109
110 //CPK (j=8)
111 forall(i in Intern)
112   sum(k in Week)x[i][8][k] == 3;
113
114 //IP (j=9)

```

```

115 forall(i in Intern)
116   sum(k in Week)x[i][9][k] == 5;
117
118 //DISP (j=10)
119 forall(i in Intern)
120   sum(k in Week)x[i][10][k] == 5;
121
122 //CPC (j=11)
123 forall(i in Intern)
124   sum(k in Week)x[i][11][k] == 4;
125
126 //QUM (j=12)
127 forall(i in Intern)
128   sum(k in 1..30)x[i][12][k] == 1;
129
130 //H (j=13)
131 forall(i in Intern)
132   sum(k in 1..30)x[i][13][k] == 1;
133
134
135 ///Intern Rotation Capacity Constraint
136
137 //CPD-G (j=1)
138 forall(k in Week)
139   sum(i in Intern)x[i][1][k] <= 2;
140
141 ///PAIRING OF GEN MED:
142 forall(k in Week)
143   2 - (sum(i in 1..10)x[i][1][k]) <= M*D[k];
144 forall(i in 1..10, k in Week)
145   x[i][1][k] <= M*(1-D[k]);
146
147 //CPD-V (j=2)
148 forall(k in Week)
149   sum(i in Intern)x[i][2][k] <= 1;
150
151 //AP (j=3)
152 forall(k in Week)
153   sum(i in Intern)x[i][3][k] <= 1;
154
155 //MIC (j=4)
156 forall(k in Week)
157   sum(i in Intern)x[i][4][k] <= 1;
158
159 //MCH (j=5)
160 forall(k in 1..8)
161   sum(i in Intern)x[i][5][k] <= 2;
162 forall(k in 9..30)
163   sum(i in Intern)x[i][5][k] <= 1;
164
165 //CPCa (j=6)
166 forall(k in Week)
167   sum(i in Intern)x[i][6][k] <= 2;
168
169 //CPM (j=7)
170 forall(k in Week)
171   sum(i in Intern)x[i][7][k] <= 2;

```

```

172
173 //CPK (j=8)
174 forall(k in Week)
175   sum(i in Intern)x[i][8][k] <= 2;
176
177 //IP (j=9)
178 forall(k in Week)
179   sum(i in Intern)x[i][9][k] <= 2;
180
181 //DISP (j=10)
182 forall(k in Week)
183   sum(i in Intern)x[i][10][k] <= 3;
184
185 //CPC (j=11)
186 forall(k in Week)
187   sum(i in Intern)x[i][11][k] <= 5;
188
189 //QUM (j=12)
190 forall(k in Week)
191   sum(i in Intern)x[i][12][k] <= 1;
192
193 //H (j=13)
194 forall(k in Week)
195   sum(i in Intern)x[i][13][k] <= 1;
196
197
198 //Intern Rotation Duration Constraint
199
200 //CPD-G (j=1)
201 forall(i in Intern)
202   sum(k in 1..47)y1[i][k] ==1;
203 forall(i in Intern, k in 1..47)
204   8 -(sum(a in 0..7)x[i][1][k + a]) <= M*(1-y1[i][k]);
205
206 //CPD-V (j=2)
207 forall(i in Intern)
208   sum(k in 1..51)y2[i][k] ==1;
209 forall(i in Intern, k in 1..51)
210   4 -(sum(a in 0..3)x[i][2][k + a]) <= M*(1-y2[i][k]);
211
212 //AP (j=3)
213 forall(i in Intern)
214   sum(k in 1..51)y3[i][k] ==1;
215 forall(i in Intern, k in 1..51)
216   4 -(sum(a in 0..3)x[i][3][k + a]) <= M*(1-y3[i][k]);
217
218 //MIC (j=4)
219 forall(i in Intern)
220   sum(k in 1..25)y4[i][k] ==1;
221 forall(i in Intern, k in 1..25)
222   2 -(sum(a in 0..1)x[i][4][k + a]) <= M*(1-y4[i][k]);
223 forall(i in Intern)
224   sum(k in 27..48)y4[i][k] ==1;
225 forall(i in Intern, k in 27..48)
226   2 -(sum(a in 0..1)x[i][4][k + a]) <= M*(1-y4[i][k]);
227
228 //MCH (j=5)

```

intern_v7.2.mod

```

229 forall(i in Intern)
230   sum(k in 1..53)y5[i][k] ==1;
231 forall(i in Intern, k in 1..53)
232   2 -(sum(a in 0..1)x[i][5][k + a]) <= M*(1-y5[i][k]);
233
234 //CPCa (j=6)
235 forall(i in Intern)
236   sum(k in 1..52)y6[i][k] ==1;
237 forall(i in Intern, k in 1..52)
238   3 -(sum(a in 0..2)x[i][6][k + a]) <= M*(1-y6[i][k]);
239
240 //CPM (j=7)
241 forall(i in Intern)
242   sum(k in 1..52)y7[i][k] ==1;
243 forall(i in Intern, k in 1..52)
244   3 -(sum(a in 0..2)x[i][7][k + a]) <= M*(1-y7[i][k]);
245
246 //CPK (j=8)
247 forall(i in Intern)
248   sum(k in 1..52)y8[i][k] ==1;
249 forall(i in Intern, k in 1..52)
250   3 -(sum(a in 0..2)x[i][8][k + a]) <= M*(1-y8[i][k]);
251
252 //IP (j=9)
253 forall(i in Intern)
254   sum(k in 1..51)y9[i][k] ==1;
255 forall(i in Intern, k in 1..51)
256   4 -(sum(a in 0..3)x[i][9][k + a]) <= M*(1-y9[i][k]);
257
258 //CPC (j=11)
259 forall(i in Intern)
260   sum(k in 1..51)y11[i][k] ==1;
261 forall(i in Intern, k in 1..51)
262   4 -(sum(a in 0..3)x[i][11][k + a]) <= M*(1-y11[i][k]);
263
264
265 //Intern Leave Constraints
266
267 //"A holiday around April and a holiday around August"
268
269 //Week 1
270 sum(k in 14..26)L1[k] ==1;
271 forall(k in 14..26)
272   sum(i in Intern)x[i][14][k] == 11*L1[k];
273
274 //Week 2
275 sum(k in 31..43)L2[k] ==1;
276 forall(k in 31..43)
277   sum(i in Intern)x[i][15][k] == 6*L2[k];
278 sum(k in 31..43)L3[k] ==1;
279 forall(k in 31..43)
280   sum(i in Intern)x[i][16][k] == 5*L3[k];
281
282
283 //Avoidance Constraint (no one week rotations during seminar weeks/public holidays) - up until
    Week 30
284

```

```

285 //Dec25, Dec26
286 forall(i in Intern, j in 12..13)
287   x[i][j][4] ==0;
288
289 //Jan1
290 forall(i in Intern, j in 12..13)
291   x[i][j][5] ==0;
292
293 //Jan27
294 forall(i in Intern, j in 12..13)
295   x[i][j][9] ==0;
296
297 //Seminar1
298 forall(i in Intern, j in 12..13)
299   x[i][j][11] ==0;
300
301 //Mar9(LabourDay)
302 forall(i in Intern, j in 12..13)
303   x[i][j][15] ==0;
304
305 //Apr10(GoodFriday)
306 forall(i in Intern, j in 12..13)
307   x[i][j][19] ==0;
308
309 //Apr13(EasterMonday)
310 forall(i in Intern, j in 12..13)
311   x[i][j][20] ==0;
312
313 //Seminar2
314 forall(i in Intern, j in 12..13)
315   x[i][j][26] ==0;
316
317 //Jun8(QueensBirthday)
318 forall(i in Intern, j in 12..13)
319   x[i][j][28] ==0;
320
321 //Seminar3
322 forall(i in Intern, j in 12..13)
323   x[i][j][33] ==0;
324
325 //Seminar4+PotentialGrandFinalFriday
326 forall(i in Intern, j in 12..13)
327   x[i][j][43] ==0;
328
329 //PotentialGrandFinalFriday
330 forall(i in Intern, j in 12..13)
331   x[i][j][44] ==0;
332
333 //Nov3(MelbCup)
334 forall(i in Intern, j in 12..13)
335   x[i][j][49] ==0;
336
337 }

```