```
/*  ************************************************************************
    OPER 527 Scheduling Problem Using GLPK
    Christopher Flippen, Richard Foster, Cindy Mitrovic - 10/22/22

    Our school has:
    3 classrooms
    6 teaching hours
    10 subjects
    4 teachers

    Each teacher has a set of classes they can teach:
    Teacher 1: {s1,s2,s3}
    Teacher 2: {s3,s4,s5}
    Teacher 3: {s5,s6,s7}
    Teacher 4: {s8,s9,10}

    Our constraints are:
    every teacher teaches all of their subjects
    every teacher teaches at most 1 class per hour
    every subject is taught
    can't have two classes in the same room at the same time
    every classroom-hour combination has at most 1 class
    can't have same subject in multiple rooms at same time

    Our goal is:
    to make the classes meet as early as possible and in as few rooms as possible
    ************************************************************************
*/

/* create a list of subjects */
set SUBJECT := {'SUB1', 'SUB2', 'SUB3', 'SUB4', 'SUB5', 'SUB6', 'SUB7', 'SUB8', 'SUB9',
'SUB10'};

/* create a list of teachers */
set TEACHER := {'BOONE', 'HURLBERT', 'BUSHAW', 'LARSON'};

/* create a list of class meeting times */
set TIME := 1..6;

/* create a list of classroom numbers */
set CLASSROOM := 1..3;

/* create a list of which teacher can teach which class */
/* if a teacher, subject pair is not in this list, then that teacher cannot teach that
class */
set PAIRS := {('BOONE', 'SUB1'), ('BOONE', 'SUB2'), ('BOONE', 'SUB3'), ('HURLBERT',
'SUB3'), ('HURLBERT', 'SUB4'), ('HURLBERT', 'SUB5'), ('BUSHAW', 'SUB5'), ('BUSHAW',
'SUB6'), ('BUSHAW', 'SUB7'), ('LARSON', 'SUB8'), ('LARSON', 'SUB9'), ('LARSON',
'SUB10')};

/* classes[t,s,h,c] = 1 if t teaches subject s at time h in classroom c , else
class[t,s,h,c] = 0 */
var classes{t in TEACHER, s in SUBJECT, h in TIME, c in CLASSROOM} binary;

/* want classes as early as possible and in room 1 as much as possible */
minimize obj: sum {t in TEACHER, s in SUBJECT, c in CLASSROOM, h in TIME}
c*h*classes[t,s,h,c];

/* every teacher teaches all their subjects (and only their subjects) */
/* we also want to ensure the teachers don't teach classes that they shouldn't */
/* for each teacher, subject pair, check all times and classrooms to make sure they
teach the class */
s.t. con1{(t,s) in PAIRS}: sum {c in CLASSROOM, h in TIME} classes[t,s,h,c] >= 1;

/* for each invalid teacher and subject pair, check all times and classrooms to make
sure they do not teach the class */
s.t. con2{t in TEACHER, s in SUBJECT: (t,s) not in PAIRS}: sum {c in CLASSROOM, h in
TIME} classes[t,s,h,c] = 0;

```

```
60  /* each teacher teaches at most 1 class per hour */
61  /* for each teacher and hour pair, check all rooms and subjects to make sure they teach
    at most once */
62  s.t. con3{t in TEACHER, h in TIME}: sum {c in CLASSROOM, s in SUBJECT} classes[t,s,h,c]
    <= 1;
63
64  /* every subject is taught */
65  /* for each subject, check the times, classrooms, and teachers to make sure it is taught
    */
66  s.t. con4{s in SUBJECT}: sum {c in CLASSROOM, t in TEACHER, h in TIME} classes[t,s,h,c]
    >= 1;
67
68  /* can't have two classes in the same room at the same time */
69  /* for each time and classroom, check the teachers and subjects to make sure there is no
    double booking */
70  s.t. con5{h in TIME, c in CLASSROOM}: sum {t in TEACHER, s in SUBJECT} classes[t,s,h,c]
    <= 1;
71
72  /* every classroom-hour combination has at most 1 class */
73  /* this constraint is already covered */
74
75  /* can't have same subject in multiple rooms at same time */
76  /* for each subject and time, check all rooms and teachers to make sure subject isn't
    taught more than once */
77  s.t. con6{s in SUBJECT, h in TIME}: sum {c in CLASSROOM, t in TEACHER} classes[t,s,h,c]
    <= 1;
78
79  solve;
80
81  printf{t in TEACHER, s in SUBJECT, c in CLASSROOM, h in TIME: classes[t,s,h,c] > 0} 'Dr.
    %s teaches %s in room %i at time %i:00\n', t,s,c,h;
82  printf 'The minimum value is %f\n', sum {t in TEACHER, s in SUBJECT, c in CLASSROOM, h
    in TIME} c*h*classes[t,s,h,c];
83
84  end;
85
```