FEUP

Optimization

Case Study
Course Scheduling in the Computer Science and Optimization Department of Monfort College

May 2022

Work developed by:

Clara Gadelho, up201806309

Flávia Carvalhido, up201806857

Leonor Gomes, up201806567

Index

Introduction

Optimization is used to solve many complex problems in many different areas, which means that the act of optimizing can have many different meanings given a specific context. Generally, optimization means “an act, process, or methodology of making something (such as a design, system, or decision) as fully perfect, functional, or effective as possible specifically: the mathematical procedures (such as finding the maximum of a function) involved in this.”
There are many problems that can be solved using traditional problem-solving techniques. However, optimization solves these same problems by aiming for the most efficient solution given the specific constraints of the setting. A very common set of issues that benefit from optimization procedures are the scheduling problems, present in almost every school, company, public transport network, etc. Having an optimized schedule can make a massive difference in the quality of service, the overall costs of running a company or a school, etc.

At Monfort College, the faculty schedule for Spring 2023 was still to be prepared. Dr. Pedro needed to come up with a schedule in only three weeks, which is a huge challenge, given all the restrictions and requests received both by students and members of the faculty, not to mention all the stress around it. Core classes, elective courses, student-athletes, class capacities and availabilities, minimum class enrollment… those were all factors to take into account when making the schedules for that semester. How would Dr. Pedro manage to get the most efficient schedule for everyone involved that could satisfy every requirement?
Therefore, the goal of the project at hand was to formulate and solve a course scheduling problem, taking into account certain restrictions and maximizing the faculty preferences.

Problem Formulation

We tackled this problem by dividing the case study into two stages. We first allocated the professors to the classes. Then, with the results obtained in the first stage, we solved the problem by dividing the classes by professor preference in MWF or TuTh and allocating teacher-class to time schedules.

Decision Variables

First Part of the Problem

Matrix Assignments[Teachers][Classes]

	CS011	CS012	CS021	CS022	CS023	CS024	CS025	CS026	CS031	CS032	CS033	CS034	CS035	CS041	CS051	CS061	CS062	CS063	CS064	CS065	CS066	CS071	CS081
Barbosa	x1,1	x1,2	x1,3	x1,4	x1,5	x1,6	x1,7	x1,8	x1,9	x1,10	x1,11	x1,12	x1,13	x1,14	x1,15	x1,16	x1,17	x1,18	x1,19	x1,20	x1,21	x1,22	x1,23
Castro	x2,1	x2,2	x2,3	x2,4	x2,5	x2,6	x2,7	x2,8	x2,9	x2,10	x2,11	x2,12	x2,13	x2,14	x2,15	x2,16	x2,17	x2,18	x2,19	x2,20	x2,21	x2,22	x2,23
Gerardo	x3,1	x3,2	x3,3	x3,4	x3,5	x3,6	x3,7	x3,8	x3,9	x3,10	x3,11	x3,12	x3,13	x3,14	x3,15	x3,16	x3,17	x3,18	x3,19	x3,20	x3,21	x3,22	x3,23
Lameiras	x4,1	x4,2	x4,3	x4,4	x4,5	x4,6	x4,7	x4,8	x4,9	x4,10	x4,11	x4,12	x4,13	x4,14	x4,15	x4,16	x4,17	x4,18	x4,19	x4,20	x4,21	x4,22	x4,23
Machado	x5,1	x5,2	x5,3	x5,4	x5,5	x5,6	x5,7	x5,8	x5,9	x5,10	x5,11	x5,12	x5,13	x5,14	x5,15	x5,16	x5,17	x5,18	x5,19	x5,20	x5,21	x5,22	x5,23
Pedro	x6,1	x6,2	x6,3	x6,4	x6,5	x6,6	x6,7	x6,8	x6,9	x6,10	x6,11	x6,12	x6,13	x6,14	x6,15	x6,16	x6,17	x6,18	x6,19	x6,20	x6,21	x6,22	x6,23
Queirós	x7,1	x7,2	x7,3	x7,4	x7,5	x7,6	x7,7	x7,8	x7,9	x7,10	x7,11	x7,12	x7,13	x7,14	x7,15	x7,16	x7,17	x7,18	x7,19	x7,20	x7,21	x7,22	x7,23
Sooeiro	x8,1	x8,2	x8,3	x8,4	x8,5	x8,6	x8,7	x8,8	x8,9	x8,10	x8,11	x8,12	x8,13	x8,14	x8,15	x8,16	x8,17	x8,18	x8,19	x8,20	x8,21	x8,22	x8,23
Contingent	x9,1	x9,2	x9,3	x9,4	x9,5	x9,6	x9,7	x9,8	x9,9	x9,10	x9,11	x9,12	x9,13	x9,14	x9,15	x9,16	x9,17	x9,18	x9,19	x9,20	x9,21	x9,22	x9,23

Where: x[i][j] is **1** if class j is taught by teacher i and **0** if it’s not

Second Part of the Problem

MWF[Teachers][MWFtimes]
TuTh[Teachers][TuThtimes]

Where:
MWF[i][j] is **1** if class i is allocated to timeslot j and **0** if it's not
TuTh[i][j] is **1** if class i is allocated to timeslot j and **0** if it's not

Objective Function

Our goal is to maximize teacher preferences having in mind their Seniority.
The objective function used for the first part of the project was this one. So in order to assign teachers to classes, we just used their seniority and the class preferences.

$$\sum_{i=1}^{Teachers} \sum_{j=1}^{Classes} x[i][j] * Seniority[i] * ClassPreferences[i][j]$$

For the second part, in allocating the combination of teacher-class with scheduling time, we divided the teacher-class with MWFClasses and TuThClasses according to the teacher (each teacher only taught classes either in MWF or TuTh. For this function, we also used the seniority of teachers and the preferences for each class hour of the professors. We added the result of the sums below the result.

$$\sum_{i=1}^{MWFclasses} \sum_{j=1}^{MWFtimes} x[i][j] * Seniority[TeachersOfClasses[i]] * MWFPreferences[TeachersOfClasses[i]][j]$$

$$\sum_{i=1}^{TuThclasses} \sum_{j=1}^{TuThtimes} x[i][j] * Seniority[TeachersOfClasses[i]] * TuThPreferences[TeachersOfClasses[i]][j] displaystyle$$

Constraints

First Part of the Problem

Hard Constraints

Explicit

1. Each faculty is assigned to only the courses that they are qualified to teach;
2. Each full-time faculty member's actual course load must be equal to their required course load.

Implicit

3. Each course should be assigned to one and only one teacher.

Soft Constraints

4. The number of different courses that each faculty taught per semester for each full-time faculty should be no more than two;
5. No more than two classes of the course CSO3 should be assigned to any full-time faculty;

6. CS06 should be taught by full-time faculty.

Second Part of the Problem

Hard Constraints

Explicit

1. At most three course classes can be assigned to the same timeslot;
2. Each full-time faculty member's actual course load must be equal to their required course load.

Implicit

3. Each course should be assigned to one and only one timeslot.

Soft Constraints

4. No faculty should teach more than two consecutive classes;
5. Classes ideally should start after 9 am and should finish before 4 pm.

Problem Resolution in CPLEX

Formulation

First Part of the Problem

For the first part of the problem, the data was formulated in CPLEX in this format.

```
Classes = {"CS011","CS012","CS021","CS022","CS023","CS024","CS025",
"CS026","CS031", "CS032","CS033","CS034","CS035",
"CS041","CS051","CS061","CS062","CS063","CS064","CS065","CS066","CS071","CS081",};

Teachers= {"Barbosa","Castro", "Gerardo","Lameiras", "Machado", "Pedro",
"Queiros", "Soeiro", "Contingent"};

//teacher course preferences
CoursePreferences = [
[4,4,3,3,3,3,3,3,5,5,5,5,5,0,0,1,1,1,1,1,1,0,0],
[3,3,0,0,0,0,0,0,3,3,3,3,3,5,0,5,5,5,5,5,5,0,0],
[1,1,3,3,3,3,3,3,3,3,3,3,3,0,0,5,5,5,5,5,5,0,5],
[5,5,5,5,5,5,5,5,3,3,3,3,3,0,0,0,5,5,5,5,5,0,0],
[0,0,0,0,0,0,0,0,5,5,5,5,5,0,0,0,0,0,0,0,0,0,0],
[3,3,5,5,5,5,5,5,5,5,5,5,5,0,0,1,1,1,1,1,1,0,0],
[3,3,5,5,5,5,5,5,1,1,1,1,1,0,5,1,1,1,1,1,1,0,0],
[1,1,1,1,1,1,1,1,3,3,3,3,3,0,0,5,5,5,5,5,5,5,0],
[3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3]];

CourseLoad=[3,2,3,3,1,3,3,3,2];

Seniority=[2,4,3,2,10,6,6,5,1];

InstructorsQualifications=[
[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,1,0],
[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1],
[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1],
[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,1,1],
[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,1,0],
[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,1,1],
[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,1,1],
[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1],
[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,1,1]
];
```

Furthermore, the model was formulated like this.

```
{string} Classes=...;
{string} Teachers=...;

int CoursePreferences[Teachers][Classes]=...;
int CourseLoad[Teachers]=...;
int Seniority[Teachers]=...;
int InstructorsQualifications[Teachers][Classes]=...;

dvar boolean  x[Teachers][Classes]; //course-teacher associations

//aux dvars
dvar int CS03teaching;
dvar int Preparations[Teachers];
dvar int AuxPrep;
dvar int SumSeniority;

maximize SumSeniority - 10*CS03teaching - 10*AuxPrep;

subject to{
  forall (i in Teachers){
    forall (j in Classes){
      InstructorsQualifications[i][j]>=x[i][j];
    }
  }

  forall (i in Teachers){
    sum (j in Classes) x[i][j]==CourseLoad[i];
  }

  forall (j in Classes){
      sum (i in Teachers) x[i][j]==1;
  }

  CS03teaching == (sum (i in Teachers) (x[i]["CS031"]+x[i]["CS032"]+x[i]["CS033"]+x[i]["CS034"]+x[i]["CS035"]>=3));

  forall(i in Teachers){
    Preparations[i]==  ((x[i]["CS031"]+x[i]["CS032"]+x[i]["CS033"]+x[i]["CS034"]+x[i]["CS035"])>=1)+
     ((x[i]["CS021"]+x[i]["CS022"]+x[i]["CS023"]+x[i]["CS024"]+x[i]["CS025"]+x[i]["CS026"])>=1)+
     ((x[i]["CS011"]+x[i]["CS012"])>=1)+
     ((x[i]["CS021"]+x[i]["CS022"]+x[i]["CS023"]+x[i]["CS024"]+x[i]["CS025"]+x[i]["CS026"])>=1)+
     ((x[i]["CS041"]+x[i]["CS051"])>=1)+
     ((x[i]["CS061"]+x[i]["CS062"]+x[i]["CS063"]+x[i]["CS064"]+x[i]["CS065"]+x[i]["CS066"])>=1);
  }

   AuxPrep==sum(i in Teachers)(Preparations[i]>=3);
   SumSeniority ==sum(i in Teachers) sum(j in Classes)Seniority[i]*x[i][j]*CoursePreferences[i][j];
}
```

Second Part of the Problem

For the second part of the problem, this is how we represented the data in CPLEX.

```
Classes = {"CS011",
"CS012","CS021","CS022","CS023","CS024","CS025","CS026",
"CS031", "CS032","CS033","CS034","CS035","CS041","CS051","CS061","CS062",
"CS063","CS064","CS065","CS066","CS071","CS081"
};

MWFClasses={"CS012","CS021","CS022","CS023","CS024","CS025","CS026","CS033","CS034","CS041","CS051","CS061"};

TuThClasses = {"CS011","CS031", "CS032","CS035","CS062","CS063","CS064","CS065","CS066","CS071","CS081"};

TeachersOfClasses=["Lameiras","Barbosa","Queiros", "Queiros", "Pedro","Queiros", "Pedro","Pedro",
"Lameiras", "Lameiras","Barbosa","Barbosa","Machado", "Castro","Contingent","Castro","Gerardo","Gerardo",
"Gerardo","Soeiro","Soeiro","Soeiro", "Contingent"];

Teachers= {"Barbosa","Castro", "Gerardo","Lameiras", "Machado", "Pedro", "Queiros", "Soeiro", "Contingent"};

MWFteachers= {"Barbosa","Castro", "Pedro", "Queiros", "Contingent"};

TuThteachers={"Gerardo","Lameiras","Machado","Soeiro","Contingent"};

MWFtimes={"9:10","10,20","11:30","1:30","2:40","3:50","5:25"};

MWFpreferences=[
[1,3,3,5,5,5,0],
[1,1,1,1,1,5,5],
[0,0,0,0,0,0,0],
[0,0,0,0,0,0,0],
[0,0,0,0,0,0,0],
[1,5,5,5,3,3,0],
[3,5,5,5,1,3,3],
[0,0,0,0,0,0,0],
[1,1,1,1,1,0,0]];

TuThtimes={"8:15","9:50","11:25","1:00","2:35","4:10","6:00"};

TuThpreferences=[
[0,0,0,0,0,0,0],
[0,0,0,0,0,0,0],
[1,5,5,1,3,5,1],
[1,5,5,5,3,3,0],
[5,0,0,0,0,0,0],
[0,0,0,0,0,0,0],
[0,0,0,0,0,0,0],
[1,5,5,5,5,5,0],
[1,1,1,1,1,1,0]];

Seniority=[2,4,3,2,10,6,6,5,1];
```

Additionally, the model was formulated like this.

```
{string} Classes=...;
{string} MWFClasses=...;
{string} TuThClasses=...;
{string} Teachers=...;
{string} MWFtimes=...;
{string} TuThteachers=...;
{string} MWFteachers=...;
{string} TuThtimes=...;

string TeachersOfClasses[Classes]=...;

int MWFpreferences[Teachers][MWFtimes]=...;
int TuThpreferences[Teachers][TuThtimes]=...;

int Seniority[Teachers]=...;

dvar boolean MWF[MWFClasses][MWFtimes];
dvar boolean TuTh[TuThClasses][TuThtimes];

//aux dvars
dvar int TotalPreferences;

//maximization function: maximize teacher preferences by seniority
maximize TotalPreferences;
```

```
subject to{
    TotalPreferences== (sum (i in MWFClasses)
                        sum (j in MWFtimes) Seniority[TeachersOfClasses[i]]*MWF[i][j]*MWFpreferences[TeachersOfClasses[i]][j] )
    + (sum (i in TuThClasses) sum (j in TuThtimes) Seniority[TeachersOfClasses[i]]*TuTh[i][j]*TuThpreferences[TeachersOfClasses[i]][j] );

    //implicit constraints
    //only one slot per class
    forall (i in MWFClasses){
      sum (j in MWFtimes) MWF[i][j]==1;
    }

    forall (i in TuThClasses){
      sum (j in TuThtimes) TuTh[i][j]==1;
    }

    //same teacher can not be allocated to more than one class in the same slot
 forall(i in MWFtimes){
        MWF["CS012"][i]+MWF["CS033"][i]+MWF["CS034"][i]<=1;
        MWF["CS041"][i]+MWF["CS061"][i]<=1;
        MWF["CS023"][i]+MWF["CS025"][i]+MWF["CS026"][i]<=1;
        MWF["CS021"][i]+MWF["CS022"][i]+MWF["CS024"][i]<=1;
  }

  forall(i in TuThtimes){
        TuTh["CS011"][i]+TuTh["CS031"][i]+TuTh["CS032"][i]<=1;
        TuTh["CS065"][i]+TuTh["CS066"][i]+TuTh["CS071"][i]<=1;
        TuTh["CS062"][i]+TuTh["CS063"][i]+TuTh["CS064"][i]<=1;
  }

    //explicit constraints
    //only 3 classes per slot
    forall (j in TuThtimes){
      sum (i in TuThClasses) TuTh[i][j]<=3;
    }
    forall (j in MWFtimes){
      sum (i in MWFClasses) MWF[i][j]<=3;
    }

    //soft constraint starting before 9am or ending after 4pm
    SChours== sum(i in MWFClasses)( MWF[i]["3:50"]+ MWF[i]["5:25"]) +
    sum(i in TuThClasses) (TuTh[i]["8:15"]+ TuTh[i]["4:10"] + TuTh[i]["6:00"]);

}
```

Results

<u>First Part of the Problem</u>

1st Solution

The **first attempt** for the **first part of the problem** was made without considering the soft constraints.

The function used was the following:

maximize SumSeniority - 0*CS03teaching - 0*AuxPrep

Where CS03teaching and AuxPrep is the auxiliar values that holds the number of times the soft constraints 4 and 5 are broken, respectively.

Teachers (size 9)	Classes (size 23)																						
	"CS011"	"CS012"	"CS021"	"CS022"	"CS023"	"CS024"	"CS025"	"CS026"	"CS031"	"CS032"	"CS033"	"CS034"	"CS035"	"CS041"	"CS051"	"CS061"	"CS062"	"CS063"	"CS064"	"CS065"	"CS066"	"CS071"	"CS081"
"Barbosa"	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0
"Castro"	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
"Gerardo"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
"Lameiras"	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
"Machado"	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
"Pedro"	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
"Queiros"	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
"Soeiro"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
"Contingent"	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Decision variables (5)	
AuxPrep	3
CS03teaching	1
Preparations	[1 2 1 3 1 3 3 1 2]
SumSeniority	456

With the obtained results, we can observe that the optimal solution without regards to the soft constraints achieves the SumSeniority (total sum of Teacher's Preferences by their seniority) of 456. We will consider this value as our base for comparing with the solutions that include the soft constraints. Our goal is to lower the number of infractions of the soft constraints without considerably lowering the value of the SumSeniority. As we can see, the optimal solution here has AuxPrep=3, which means that there are 3 teachers who teach more than 2 classes from different courses and CS03teaching=1, which means that 1 teacher teaches more than 2 CS03 classes. We will now try different solutions where these values will be lower, and see how that affects the value of SumSeniority.

2nd Solution

For the **second attempt**, the soft constraints had a small penalty associated and the same weight and it was calculated like this:

maximize SumSeniority - 3*CS03teaching - 3*AuxPrep

The results are the following:

Teachers (size 9)	Classes (size 23)																						
	"CS011"	"CS012"	"CS021"	"CS022"	"CS023"	"CS024"	25"	"CS026"	"CS031"	"CS032"	"CS033"	"CS034"	"CS035"	"CS041"	"CS051"	"CS061"	"CS062"	"CS063"	"CS064"	"CS065	"CS066"	"CS071"	"CS081"
"Barbosa"	0	0	0	0	0	0		0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
"Castro"	0	0	0	0	0	0		0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
"Gerardo"	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
"Lameiras"	1	1	0	0	0	0		0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
"Machado"	0	0	0	0	0	0		0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
"Pedro"	0	0	0	0	1	1		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
"Queiros"	0	0	1	1	0	0		1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
"Soeiro"	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
"Contingent"	0	0	0	0	0	0		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Decision variables (5)	
AuxPrep	0
CS03teaching	1
Preparations	[1 2 1 2 1 2 2 1 1]
SumSeniority	452

As we can see, applying a small penalty to the infraction of the soft constraints gets us a solution where AuxPrep=0, meaning that the constraint 4 is entirely respected (as if it was a hard constraint) and CS03teaching=1, meaning that constraint 5 is still infringed once, as it was in the 1st solution. If we look at the value of SumSeniority we can see that it became smaller than before but only by 4 units, which means that we were able to respect the soft constraints without affecting the preferences of the teachers too much. Let's now attempt to ensure the fulfillment of both soft constraints.

3rd Solution

As a f**inal solution to the first part of the problem**, we considered that the soft constraints had a big penalty associated with them. The formula was the following:

maximize SumSeniority - 10*CS03teaching - 10*AuxPrep

The results are the following:

Teachers (size 9)	Classes (size 23)																						
	"CS011"	"CS012"	"CS021"	"CS022"	"CS023"	"CS024"	"CS025"	"CS026"	"CS031"	"CS032"	"CS033"	"CS034"	"CS035"	"CS041"	"CS051"	"CS061"	"CS062"	"CS063"	"CS064"	"CS065"	"CS066"	"CS071"	"CS081"
"Barbosa"	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
"Castro"	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
"Gerardo"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
"Lameiras"	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
"Machado"	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
"Pedro"	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
"Queiros"	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
"Soeiro"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
"Contingent"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Decision variables (5)

AuxPrep	0
CS03teaching	0
Preparations	[2 2 1 2 1 2 2 1 1]
SumSeniority	446

Here, the values of both AuxPrep and CS03teaching are 0, which means that both soft constraints are not broken even once. As we can see from the value of SumSeniority, the total sum of teachers' preferences was reduced by 10 units from the 1st solution. We consider this a small decrease given that it allowed for the fulfillment of both soft constraints- With that in mind we were satisfied with this solution and used it to complete the second part of the problem.

Second Part of the Problem

1st Solution

The **first solution** found didn't take into account the soft constraints. The first formula used was the following:

maximize TotalPreferences

The obtained results were:

MWFClasses (size 12)	MWFtimes (size 7)						
	"9:10"	"10,20"	"11:30"	"1:30"	"2:40"	"3:50"	"5:25"
"CS012"	0	0	0	1	0	0	0
"CS021"	0	1	0	0	0	0	0
"CS022"	0	0	1	0	0	0	0
"CS023"	0	0	0	1	0	0	0
"CS024"	0	0	0	1	0	0	0
"CS025"	0	1	0	0	0	0	0
"CS026"	0	0	1	0	0	0	0
"CS033"	0	0	0	0	1	0	0
"CS034"	0	0	0	0	0	1	0
"CS041"	0	0	0	0	0	0	1
"CS051"	1	0	0	0	0	0	0
"CS061"	0	0	0	0	0	1	0

TuThClasses (size 11)	TuThtimes (size 7)						
	"8:15"	"9:50"	"11:25"	"1:00"	"2:35"	"4:10"	"6:00"
"CS011"	0	0	1	0	0	0	0
"CS031"	0	0	0	1	0	0	0
"CS032"	0	1	0	0	0	0	0
"CS035"	1	0	0	0	0	0	0
"CS062"	0	0	0	0	0	1	0
"CS063"	0	1	0	0	0	0	0
"CS064"	0	0	1	0	0	0	0
"CS065"	0	0	0	0	1	0	0
"CS066"	0	0	0	1	0	0	0
"CS071"	0	0	0	0	0	1	0
"CS081"	0	0	0	0	1	0	0

SChours	6
TotalPreferences	452

With this primary solution we were able to identify the optimal solution when we don't consider the soft constraints. The value of the variable TotalPreferences accounts for the value we want to maximize: the schedule preferences of the teachers according to their seniority, which in this case was 452. We will now use this value as a standard for our next solutions, to check how much the soft constraints affect its magnitude.
With the variable SChours, we are counting the number of classes that start before 9 am or end after 4 pm. With its value being 6, we can say that there are 6 classes that do not obey that soft constraint, we will now try to minimize this number in the next solutions.

2nd Solution

For the **second attempt**, the soft constraint of course allocation before 9 am and after 4 pm had a great penalty associated and it was calculated like this:

maximize TotalPreferences -200*SChours

The solution obtained was:

MWFClasses (size 12)	MWFtimes (size 7)						
	"9:10"	"10,20"	"11:30"	"1:30"	"2:40"	"3:50"	"5:25"
"CS012"	0	0	0	1	0	0	0
"CS021"	0	1	0	0	0	0	0
"CS022"	0	0	1	0	0	0	0
"CS023"	0	0	0	1	0	0	0
"CS024"	0	0	0	1	0	0	0
"CS025"	0	1	0	0	0	0	0
"CS026"	0	0	1	0	0	0	0
"CS033"	0	0	0	0	1	0	0
"CS034"	0	1	0	0	0	0	0
"CS041"	1	0	0	0	0	0	0
"CS051"	1	0	0	0	0	0	0
"CS061"	0	0	0	0	1	0	0

TuThClasses (size 11)	TuThtimes (size 7)						
	"8:15"	"9:50"	"11:25"	"1:00"	"2:35"	"4:10"	"6:00"
"CS011"	0	0	0	1	0	0	0
"CS031"	0	0	1	0	0	0	0
"CS032"	0	1	0	0	0	0	0
"CS035"	0	0	0	0	1	0	0
"CS062"	0	0	1	0	0	0	0
"CS063"	0	0	0	0	1	0	0
"CS064"	0	1	0	0	0	0	0
"CS065"	0	0	0	1	0	0	0
"CS066"	0	0	1	0	0	0	0
"CS071"	0	0	0	0	1	0	0
"CS081"	0	1	0	0	0	0	0

SChours	0
TotalPreferences	360

With this attempt, we wanted to see if it was possible to obey the soft constraint completely. As we can see from the value of SChours, it was possible, however the TotalPreferences value suffered a great decrease from its previous value. We think that such a tradeoff between the soft constraint and the optimal solution is not desirable, so we discarded this solution and tried a smaller penalty to try to achieve a more balanced solution.

3rd Solution

For the **final solution to the problem**, we considered that the soft constraint had a small penalty associated with it. The formula was the following:

maximize TotalPreferences -10*SChours

The solution was as follows:

MWFClasses (size 12)	MWFtimes (size 7)						
	"9:10"	"10,20"	"11:30"	"1:30"	"2:40"	"3:50"	"5:25"
"CS012"	0	0	0	1	0	0	0
"CS021"	0	1	0	0	0	0	0
"CS022"	0	0	1	0	0	0	0
"CS023"	0	0	0	1	0	0	0
"CS024"	0	0	0	1	0	0	0
"CS025"	0	1	0	0	0	0	0
"CS026"	0	0	1	0	0	0	0
"CS033"	0	0	0	0	1	0	0
"CS034"	0	1	0	0	0	0	0
"CS041"	1	0	0	0	0	0	0
"CS051"	1	0	0	0	0	0	0
"CS061"	0	0	0	0	1	0	0

TuThClasses (size 11)	TuThtimes (size 7)						
	"8:15"	"9:50"	"11:25"	"1:00"	"2:35"	"4:10"	"6:00"
"CS011"	0	0	0	1	0	0	0
"CS031"	0	0	1	0	0	0	0
"CS032"	0	1	0	0	0	0	0
"CS035"	0	0	0	0	1	0	0
"CS062"	0	0	1	0	0	0	0
"CS063"	0	0	0	0	1	0	0
"CS064"	0	1	0	0	0	0	0
"CS065"	0	0	0	1	0	0	0
"CS066"	0	0	1	0	0	0	0
"CS071"	0	0	0	0	1	0	0
"CS081"	0	1	0	0	0	0	0

SChours	3
TotalPreferences	442

In this solution, we were able to find a compromise between obeying the soft constraint and the teachers' preferences. The amount of classes that broke the soft constraint were reduced to half of the base solution, while the TotalPreferences suffered a small loss. We believe that this is a much preferable solution to the problem.

Assumptions made

Throughout the entire project, we had to make some assumptions. Firstly, we considered the constraint of CS06 being taught by full-time faculty a hard constraint. Additionally, as the number of classes of each course is fixed, we considered each class its own variable. Thirdly, when giving value to the “Contingent” course preferences, we opted for 3 since it was a neutral value, although this has little to no impact on the solution. Finally, because of our chosen formulation, the constraint that limited the number of consecutive classes by each teacher was really difficult to implement, so we ended up not considering it.

Conclusion

After fully reading the problem description, it became clear that many of the mentioned details wouldn’t play an active role in the problem’s solution, simplifying the formulation from the very beginning. Therefore, dividing the whole problem into 2 separate stages made the problem formulation much easier and understandable, thus making the problem resolution inherently simple. Furthermore, by applying the soft constraints and comparing final results with and without them, it has become clear the effect that these have on the overall optimal solution. In conclusion, having solved this problem has provided us with much more awareness about the importance of optimization in scheduling problems, as well as given us more experience in dealing with complex issues, such as the one stated for this project.