# Parental Consultation Scheduling Problem

Course 42133

March 7, 2011

In this project you should develop a metaheuristic for the Parental Consultation Scheduling Problem. Furthermore you also have the opportunity to compete with your fellow students. In the final phase of the course, you can submit your code to Thomas, and by the use of some extra datasets the best metaheuristic will be determined. Thomas will also provide a price for the best metaheuristic. The only requirement for participating in the competition is that you write your code in C#. This requirement is necessary to make a fair comparison.

It is important to mention two things:

- It is completely voluntarily if you want to participate in the competition!
- Your performance in the competition will not affect your grade!

So if you do not like to code C# or you do not want to compete, you can of course still select this project! This project has some nice features:

- You will be among the first in the world to research on this problem!
- The mathematical model is given.
- We provide a small program which calculates the objective and ensures feasibility.
- Datasets are given.
- Thomas' price....

## 1 Problem Description

Once or twice a year the high schools invite the students and their parents to participate in consultations between them and the teachers of a given course. The goal of these meetings is to allow the teachers to inform the parents of the students educational progression. The process for parental consultations is such that each student (in collaboration with his parents) makes prioritized requests for teachers he would like to meet for a consultation. The problem from a operational point of view is to allocate the requests to a schedule such that as many request as possible is granted, meanwhile the consultation plan for each individual is acceptable.

When assigning students and teacher to a time slot where a consultation should take place, it is necessary to take several parameters into consideration. First of all, a student or a teacher can not be assigned to a time slot if they are not available at the given time. Furthermore, when assigning consultations it is attempted to achieve a solution where the positions of the granted requests for a given individual is placed as close together as possible. This is to achieve a schedule for each individual with as little waiting time as possible. A break is a time slot with no allocated consultations. Three types of breaks exist; Forced, void and undesirable. A student is forced to have a break after each consultation. This is due to traveling time between meeting-rooms. A time slot is void for a teacher or student if the time slot is empty and no consultations are scheduled in neither earlier time slot or in later time slots. Void time slots must be distinguished from other empty time slots because they do not effect the density of a schedule. This is due to the fact that students and teachers are not obligated to stay at the school throughout the entire duration of all consultations. Figure 1 shows an example of a consultation schedule. The schedule contains 1 void time slot, 2 breaks, and 7 consultations.

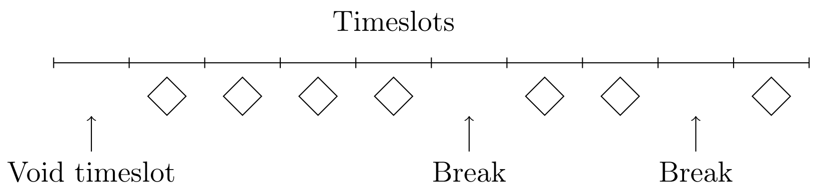


**Figure 1:** *Example of a feasible consultation schedule for a teacher.*

## 2 Mathematical Model

### Indices

- $t \in T$: Teachers
- $b \in B$: Blocks
- $s \in S$: Students

### Parameters

- $R_t^s = \begin{cases} 1 & \text{Student } s \text{ has requested teacher } t \text{ for a consultation} \\ 0 & \text{Otherwise.} \end{cases}$
- $E_b^s = \begin{cases} 1 & \text{Student } s \text{ is available in block } b \\ 0 & \text{Otherwise.} \end{cases}$
- $D_{t,b} = \begin{cases} 1 & \text{Teacher } t \text{ is available in block } b \\ 0 & \text{Otherwise.} \end{cases}$
- $\alpha_t^s \in \mathbb{R}_+$: The profit of granting a request from student $s$ for teacher $t$.
- $\beta_t \in \mathbb{R}_+$: The cost of a undesirable break for teacher $t$.
- $\gamma^s \in \mathbb{R}_+$: The cost of a undesirable break for student $s$.

### Variables

- $x_{t,b}^s = \begin{cases} 1 & \text{Student } s \text{ is given a consultation with teacher } t \text{ in block } b \\ 0 & \text{Otherwise.} \end{cases}$

**The model**

The following nonlinear function defines the number of undesirable breaks for a teacher $t$ with more than two consultations.

$$\mathcal{W}^t = \Delta\left(\mathfrak{B}_t^{\text{last}}, \mathfrak{B}_t^{\text{first}}\right) + 1 - \sum_{b,s} x_{t,b}^s \tag{1}$$

where $\mathfrak{B}_t^{\text{first}} \in B$ is the first time slot where teacher $t$ has a consultation; $\mathfrak{B}_t^{\text{last}} \in B$ is the last time slot where teacher $t$ has a consultations. The decision whether a student has a undesirable break is somewhat analogue.

$$\mathcal{Z}^s = \Delta\left(\mathfrak{B}_s^{\text{last}}, \mathfrak{B}_s^{\text{first}}\right) + 2 - 2 \cdot \sum_{t,b} x_{t,b}^s \tag{2}$$

Parental Consultation Scheduling IP Model (3)

$$
\begin{aligned}
\max \quad & \sum_{t,b,s} \alpha_t^s \cdot x_{t,b}^s - \sum_t \beta_t \cdot \mathcal{W}_t - \sum_s \gamma^s \cdot \mathcal{Z}^s && && \text{(3a)}\\
\text{s.t.} \quad & \sum_b x_{t,b}^s \leq R_t^s && \forall t, s && \text{(3b)}\\
& \sum_t x_{t,b}^s \leq E_b^s && \forall b, s && \text{(3c)}\\
& \sum_s x_{t,b}^s \leq D_{t,b} && \forall t, b && \text{(3d)}\\
& \sum_t \left(x_{t,b}^s + x_{t,b+1}^s\right) \leq 1 && \forall s, b \in B \setminus \{b_{|B|}\} && \text{(3e)}\\
& x_{t,b}^s \in \{0,1\} && && \text{(3f)}
\end{aligned}
$$

## 3 Setup

We provide you the datasets in .csv format. It is part of your task to implement a method which reads these .csv files. We also provide a small .exe file which can check the feasibility and calculate the objective of an output file. For the results of the competition, this .exe file will be used to calculate the value of your found solution. For the final tests, code will be run on the same machine, so it is important that your code is easily portable to another machine (usually not a problem).

In the following, all indexes are zero-based. E.g. if a dataset contains 3 teachers, these are denoted `t0`, `t1` and `t2`.

## 4 Data

10 datasets are given, each representing a real-life instance taken directly from the Lectio-database for the school-year 2010. A dataset consists of several .csv files, see Table 1.

## 5 Solution format

The solution format is a textfile representation of $x_{t,b}^s$, specified by the line-format [Student];[Teacher];[Block]. E.g.:

| Filename | Symbol | Line-format | Example |
|---|---|---|---|
| `basic.csv` | $\lvert S\rvert$, $\lvert T\rvert$, $\lvert B\rvert$ | Students:$\lvert S\rvert$ | `Teachers:156` |
| `requests.csv` | $R^s_t$, $\alpha^s_t$ | [Student];[Teacher];[Weight: $\alpha^s_t$] | `s17;t2;2.4` |
| `teachbreaks.csv` | $\beta_t$ | [Teacher];[Weight: $\beta_t$] | `t0;0.8` |
| `studbreaks.csv` | $\gamma^s$ | [Student];[Weight: $\gamma^s$] | `s328;0.6` |
| `teachavail.csv` | $D_{t,b}$ | [Teacher];[Block] | `t6;b15` |
| `studavail.csv` | $E^s_b$ | [Student];[Block] | `s17;b2` |

**Table 1:** *Files in a dataset*

```
s4;t55;b2
s18;t14;b15
s66;t4;b5
...
```

## 6 C# Hints

You should not spend too much time writing trivial I/O operations, so here are some hints for C#:

- File operations can be performed by the System.IO namespace. Use for instance the classes File, FileInfo, Directory, StreamReader....etc
- You can use the String.Split(new [] ';') method to split the lines you read from the files.