Grouping Medical Students

Introduction

This project was concerned with the automatic grouping of students in a Medical school in order to satisfy important constraints. Such groupings are performed several times per semester.

The basic problem is to minimize the number of times any pair of students is grouped together, so that each student works with different students as often as possible. In most cases, each group must also contain the same number of students. In some cases, each group must contain the same number of Dental students. And finally, gender plays a similar role.

The goal is to give each student the widest possible diversity of fellow group members, because these groups are problem-based learning (PBL) tutorials, rather than conventional lectures.

Every week a medical problem is posed. Each group meets several times during that week in order to "solve" it. By varying group membership in each block, students are exposed to the "real-world" conditions of medical practice, where they must consult wildly varying expertise.

The first two years at the school are divided into PBL blocks of 4 to 7 weeks. For each PBL block the students are partitioned into a small number of groups. For each block and group, one teacher is assigned to it. Typically, there about 8 students per group. This combination meets three times per week, for two hours at a time. Each block-week has a theme, to which all courses conform to varying degrees. That is, the lectures, labs, and groups for a given week focus on that week's theme (eg. Carbohydrate Metabolism). Block and week chairs are responsible for designing and scheduling courses to address the weekly themes, which collectively encompass the foundations of Medicine.

A conflict occurs if two students meet in the same group for a given block and they've already met in another group in a previous block. The objective of this project was to design and implement an algorithm which assigns each student to a group in each block so that the total number of conflicts is minimized and there are an equal number of students in each group.

Technical Description

A simplified version of an assignment will be defined, along with a typical mathematical restriction on when optimal assignments are even possible. Then a brief overview of the software will be presented.

- Assignment Problem

Mathematically, an assignment is the following:

Let

S = number of students

B = number of blocks

G = number of groups for each block (assumed to be the same for each block)

Then an assignment of students to groups, one per block for each student, is defined as an S X B matrix A containing only the values 1 to G, where

g = A(s,b) iff student s is assigned group g in block b

g = 1, …, G
s = 1, …, S
b = 1, …, B

and the number of students is divisible by the number of groups, and the number of students in each group is the same.

For example, 8 students assigned to 4 groups over 6 blocks might be represented by the following matrix A:

	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6
Student 1	1	2	3	4	1	2
Student 2	2	3	4	1	2	3
Student 3	3	4	1	2	3	4
Student 4	4	1	2	3	4	1
Student 5	1	2	3	4	1	2
Student 6	2	3	4	1	2	3
Student 7	3	4	1	2	3	4
Student 8	4	1	2	3	4	1

Here, the second student is assigned the fourth group in the third block.

More generally, the first student is assigned group 1 for block 1, group 2 for block 2 while the second student is assigned group 2 for block 1, group 3 for block 2, etc.

This is a very bad assignment because Student 1 is always in the same group as Student 5, Student 2 is always in the same group as Student 6, etc.

A better assignment would be the following:

	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6
Student 1	1	1	1	1	1	1
Student 2	2	2	2	2	2	2
Student 3	3	3	3	3	3	3
Student 4	4	4	4	4	4	4
Student 5	1	2	3	4	1	2
Student 6	2	3	4	1	2	3
Student 7	3	4	1	2	3	4
Student 8	4	1	2	3	4	1

Here, although the first 4 students never meet each other, nor do the last 4, each member of the first group meets every member of the second group at most twice.

Even better would be the following, where only Students 3, 5 and Students 4, 6 meet more than once.

	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6
Student 1	1	1	1	1	1	1
Student 2	2	2	2	2	2	2
Student 3	1	2	**3**	**4**	3	3
Student 4	**3**	**4**	1	2	4	4
Student 5	4	3	**3**	**4**	1	2
Student 6	**3**	**4**	4	1	2	3
Student 7	2	3	4	3	4	1
Student 8	4	1	2	3	3	4

By definition, an assignment is optimal if it minimizes the maximum number of times any two students meet.

The following rough inequality shows when optimal assignments are not possible if there are too many blocks.

Theorem

Suppose you assign S students evenly to G groups over B blocks, so that no two students are in the same group more than once. Then B must be at most G(S-1) / (S – G).

For example, if S = 14 and G = 7 (so that each of the 7 groups contains 2 students) then the number of blocks B is at most 13 for any optimal assignment.

Proof:

First recall that for any set of S objects, there are S(S-1)/2 collections of 2 objects from that set (elementary combinatorics).

For example, consider the set {1,2,3,4} consisting of S = 4 objects. The 6 collections of 2 objects from this set are:

{1,2}
{1,3}
{1,4}
{2,3}
{2,4}
{3,4}

Let's count all of the collections of 2 students who belong to the same group in the first block.

In each group there are S/G students so each group contributes (S/G)(S/G – 1)/2 to the final count. Since there are G blocks, the final count is G(S/G)(S/G – 1)/2.

For the second block we do the same thing, noting that we must produce new collections of 2 students, since no two students are in the same group more than once. So there are G(S/G)(S/G – 1)/2 new collections obtained from the second block.

The same reasoning holds for blocks 3, 4, ..., B.

So the total number of collections obtained from the B blocks is BG(S/G)(S/G – 1)/2.

But the total possible number of collections of 2 students from S students is S(S-1)/2.

Therefore,

$BG(S/G)(S/G - 1)/2 <= S(S-1)/2$
$B(S/G - 1) <= (S-1)$
$B <= (S-1)/(S/G - 1)$
$B <= G(S-1)/(S-G)$

Based on these and other mathematical conditions governing the best way for assigning students to minimize conflicts, a computer program called PBLGrouping was written.

The constraints addressed by this program are those described above, along with some additional restrictions (any one of which may be imposed on individual blocks):

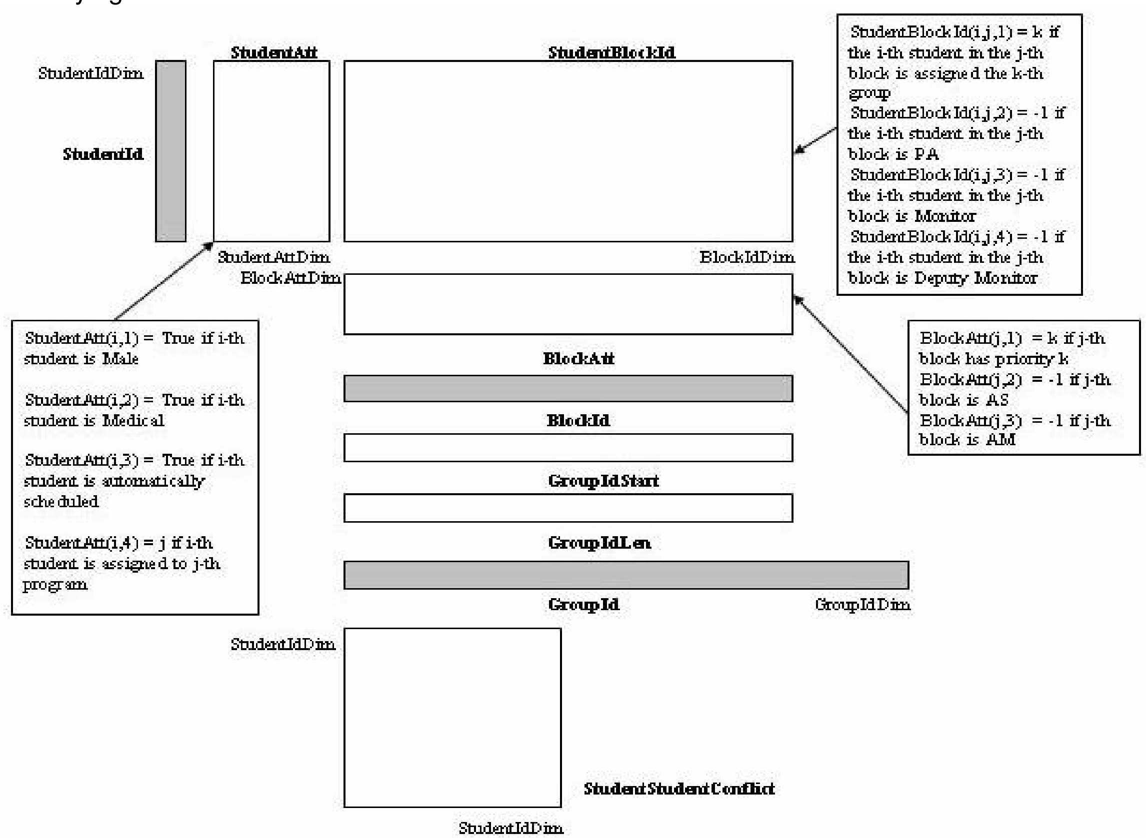
1. There must be equal numbers of Medical students in each group, equal numbers of Dental students in each group, and equal numbers of Males and Females in each group

2. Each group must be all Male or all Female and within each group there must be equal numbers of Medical students, and equal numbers of Dental students

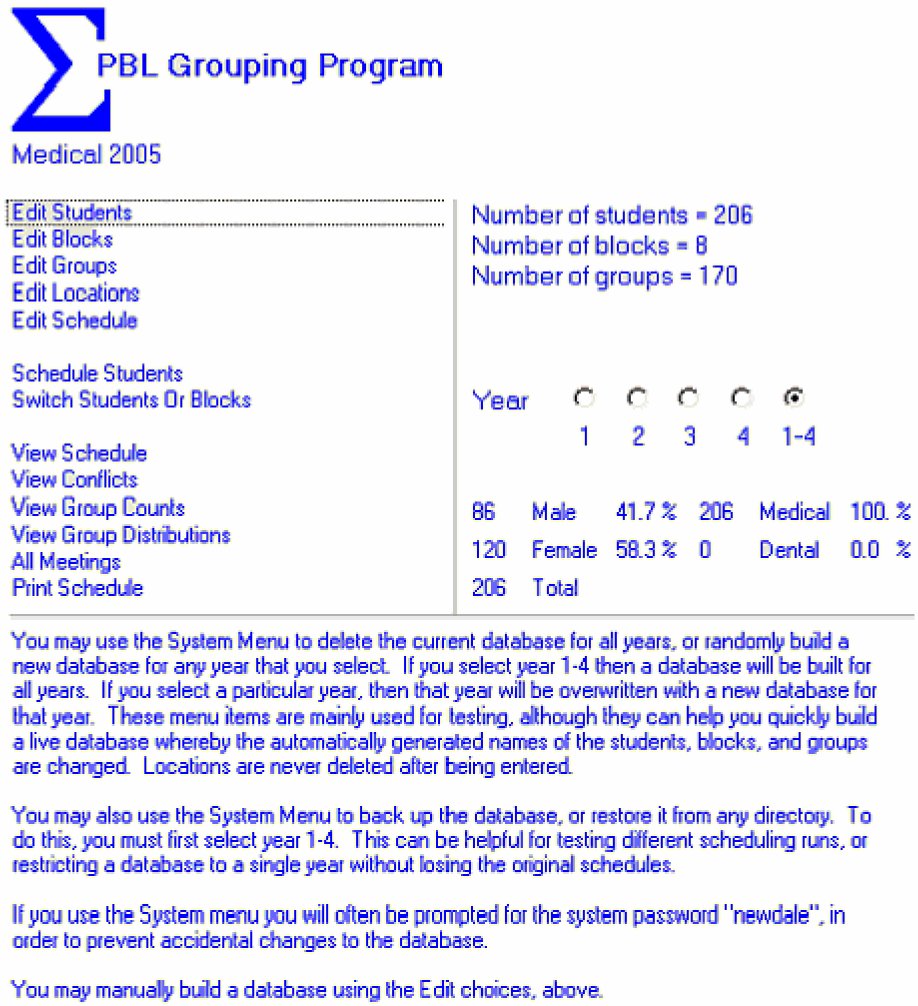
3. Each group contains Medical students only, and within each group there must be equal numbers of Male and Female students

Technical Description

A computer program called PBLGrouping was written automatically assigning students. The underlying data structures look like this:



This program allows the end user to input the names of the blocks, groups for each block, and students (along with their gender and medical/dental affiliation). The software then assigns all of the students to groups in each block while monitoring the effect on the total number of conflicts for each assignment. Subject to the mathematical constraints imposed on the number of blocks, groups and students, the program locates the proper assignments to minimize conflicts using a "greedy algorithm" type approach.



PBL Grouping Program

Users may define additional constraints for each block using a drop-down menu of choices.

A variety of reports are available showing where conflicts occur in the final schedule, along with the distribution of students by group.

The final schedule may be manually edited in any way, for fine-tuning.

The Windows program runs quickly, assigning several hundred students over ten blocks in about five minutes.

Output may be ported to Excel for further analysis, and the database may be attached to related applications using ODBC (eg. contact management system maintaining student demographics).