# Partitioning of students into equitable groups using SolverStudio

M. Fairley*, O. Dowson
Department of Engineering Science
University of Auckland
New Zealand
*mfai035@aucklanduni.ac.nz

## Abstract

Students in their final year of a Bachelor of Engineering at the University of Auckland are required to complete a course known as ENGGEN 403. As part of this course, students are partitioned into groups of around 25 students and given a single week to solve a large problem. In order to make this partitioning fair and equitable, the students should be partitioned in a way that makes the groups as similar as possible. This paper details the development, implementation, and results of an Excel based optimisation solution for this problem. A mixed integer programme was formulated to balance the number of students in each group by academic performance, gender, ethnicity and engineering discipline. This model was implemented in the PuLP modelling language using the SolverStudio modelling environment. The model was validated against data from the 2013 ENGGEN 403 class, and shown to improve all key metrics. The Excel spreadsheet was given to the course organiser who used the model to partition the 571 students of the 2014 ENGGEN 403 class into 23 groups. We believe this paper provides a good case study on how the combination of Excel, SolverStudio and PuLP enables the rapid development of practical optimisation solutions.

**Key words:** excel, pulp, solverstudio, group allocator

## 1 Introduction

Students in their final year of a Bachelor of Engineering at the University of Auckland are required to complete a course known as ENGGEN403. As part of the course, students are partitioned in to groups of around 25 students and given a single week to produce a large piece of work on a given topic. In recent years, these topics have included a rebuild plan for Christchurch, a proposal for the future of Aucklands transport network and a tender to run a new public-private partnership investment agency.

Although the stated aim of the project is to practice the skills taught throughout the course, a large component relates to team dynamics.

This paper details the development, implementation and results of an optimisation solution to the student partitioning problem.

### 1.1 Variables of Interest

There is no clear definition of fair and equitable, however we can identify two distinct types of variables that are of interest: academic ability and demographic composition. The first is important as it would be unfair to have one group composed of mainly high-achieving students, and another with low-achieving students.

The variable of most interest is grade point average (GPA). At the University of Auckland, GPA's are measured on a 1-9 scale, where 1 represents a C- average, and 9 represents an A+ average. There is anecdotal evidence from students and the course organiser of groups in previous years with large numbers of high GPA students. Although this does not necessarily lead to a better grade, there is a perception of inequity amongst students from other groups. A second cause for concern was groups with a bi-modal split of high-GPA students and low-GPA students.

Therefore, two key metrics in determining the quality of any partition are the mean GPA of a group, and the variance of student GPAs within a groups.

ENGGEN403 is a common paper to all the engineering disciplines at the University of Auckland. In addition, each discipline has a unique set of skills and perspective to contribute to group projects. To give all groups an equal balance of these skills, and to prevent a single discipline from dominating a group, students from each discipline should be distributed equally amongst the groups.

Two other demographic variables are gender and ethnicity. Although these have less of an impact on group performance than GPA or discipline, group demographics can have a large impact on the perceived fairness and enjoyment by students. Therefore, there should be an equal number of students identifying as each gender in every group. Balancing on ethnicity also contributes to the perceived fairness of the groups by students.

### 1.2 Previous Solution

In previous years, the job of partitioning the students into groups has been the course organisers job. The method for doing so was not rigorous, but can be considered to be a greedy heuristic. The basic method was as follows:

Starting with the smallest discipline, distribute the students across the groups aiming to balance the mean GPA of each group. Once complete, move onto the next smallest discipline and repeat. As more students are added, try to balance gender as well as GPA. Once all students are added, perform a series of swaps between pairs of students to improve the overall balance.

This process usually took on the order of two days to complete.

## 2 Mathematical model

In this section, we define the mixed integer mathematical programme that was formulated to solve the problem.

**Sets**

Students: $S = \{1, 2, \ldots, m\}$;
Groups: $G = \{1, 2, \ldots, n\}$;
Ethnicities: $E = \{\text{European, Māori, Pacific, Asian, Other}\}$;
Genders: $K = \{\text{Male, Female, Other}\}$;

Disciplines: D = {Civil, Chemical & Materials, Mechatronics,Mechanical, Electrical, Software, Computer Systems, Engineering Science, Biomedical};
Students who identify as gender $k$: $S_k^K \subset S, \quad \forall k \in K$;
Students who identify as ethnicity $e$: $S_e^E \subset S, \quad \forall e \in E$;
Students who study discipline $d$: $S_d^D \subset S, \quad \forall d \in D$.

**Parameters**

$\gamma_s$ = grade point average of student $s$;
$M_g$ = the number of students in group $g$;
**Note:** the values of $M_g$ are computed a priori. They should add up to the total number of students in the class.
$\alpha$ = priority given to allocating groups with equal mean compared to groups with equal variance. A good value for this is 0.5;
$\beta^I$ = penalty weighting put on violating demographic constraints of type I, $I = D, E, K$.

**Decision variables**

$x_{sg}$ = allocate student $s$ to group $g$;
$\rho_{gi}^I$ = artificial penalty variable for group $g$, factor $i$, $\quad I = \{K, D, E\}$;
$\mu_{max}$ = maximum mean GPA of group;
$\mu_{min}$ = minimum mean GPA of group;
$\sigma_{max}^2$ = maximum variance of group GPA;
$\sigma_{min}^2$ = minimum variance of group GPA;
$\hat{\mu}$ = mean GPA of all students;
$\hat{\sigma^2}$ = variance in the GPA of all students.

**Objective Function**

The objective function is to minimise a convex combination of the spread between the maximum and minimum mean group GPA, and the spread between the maximum and minimum variance of the GPAs within a group. Adding a penalty term with a large coefficient $\beta$ is necessary as some constraints are relaxed.

$$\begin{aligned} \text{Minimise } & \underbrace{\alpha \left( \frac{\mu_{max} - \mu_{min}}{\hat{\mu}} \right)}_{\text{normalised mean}} + \underbrace{(1-\alpha) \left( \frac{\sigma_{max}^2 - \sigma_{min}^2}{\hat{\sigma^2}} \right)}_{\text{normalised variance}} \\ & + \underbrace{\sum_{g \in G} \left( \beta^K \sum_{k \in K} \rho_{gk}^K + \beta^D \sum_{d \in D} \rho_{gd}^D + \beta^E \sum_{e \in E} \rho_{ge}^E \right)}_{\text{penalty variables}} \end{aligned} \tag{1}$$

**Constraints**

**Partitioning Constraint** These constraints ensure that all students are allocated to exactly one group.

$$\sum_{g=1}^{n} x_{sg} = 1 \qquad \forall s \in S \tag{2}$$

$$\sum_{s \in S} x_{sg} = M_g \qquad \forall g \in G \tag{3}$$

**Bounds on objective variables** These constraints set the values of the variables used in the objective function.

$$\sum_{s \in S} \frac{x_{sg} \times \gamma_s}{M_g} \geq \mu_{min} \qquad \forall g \in G \tag{4}$$

$$\sum_{s \in S} \frac{x_{sg} \times \gamma_s}{M_g} \leq \mu_{max} \qquad \forall g \in G \tag{5}$$

$$\sum_{s \in S} \frac{x_{sg} \times \gamma_s^2}{M_g} - \hat{\mu}^2 \geq \sigma_{min}^2 \qquad \forall g \in G \tag{6}$$

$$\sum_{s \in S} \frac{x_{sg} \times \gamma_s^2}{M_g} - \hat{\mu}^2 \leq \sigma_{max}^2 \qquad \forall g \in G \tag{7}$$

**Bounds on demographic variables** These constraints set the minimum number of students in each group from each demographic sub-group. These constraints are relaxed via the addition of $\rho$ variables.

$$\sum_{s \in S_k^K} x_{sg} + \rho_{gk}^K \geq \left\lfloor \frac{M_g \times |S_k^K|}{|S|} \right\rfloor \qquad \forall g \in G \quad \forall k \in K \tag{8}$$

$$\sum_{s \in S_d^D} x_{sg} + \rho_{gd}^D \geq \left\lfloor \frac{M_g \times |S_d^D|}{|S|} \right\rfloor \qquad \forall g \in G \quad \forall d \in D \tag{9}$$

$$\sum_{s \in S_e^E} x_{sg} + \rho_{ge}^E \geq \left\lfloor \frac{M_g \times |S_e^E|}{|S|} \right\rfloor \qquad \forall g \in G \quad \forall e \in E \tag{10}$$

**Variable Bounds** The allocation variables ($x$) are binary, all other variables are non-negative.

$$x_{sg} \in \{0, 1\}$$

$$\mu_{min}, \mu_{max}, \sigma_{min}^2, \sigma_{max}^2, \rho_{gi}^I \geq 0.$$

### 2.1 Estimating the variance of a group

In order to calculate the variance of a sample, one needs to know the square of the mean of that sample. This is a non-linear function of a variable in the model. Therefore, to prevent the problem from being non-linear, we make an assumption that the mean of every group is equal to the mean of all the students. This changes the square from acting on a variable, to acting on a parameter. We can make this assumption as constraints (4) and (5) act to force the mean GPA of each group towards the global mean.

### 2.2 The affect of relaxing the demographic constraints

It is noticed that the demographic constraints take the form of

$$\sum_i x_i + y \geq z \tag{11}$$

where $x_i$ and $z$ are both integer values. Since the objective is to minimise, and the objective coefficients of the $y$ variables are positive, they will optimise to their lower bound. This implies that the $y$ variables will also take integer values in an optimal solution.

If the values are $\beta$ are large relative to the rest of the objective function, then the affect of this is to optimise the demographics of the groups first, and then proceed optimising the GPAs. When allocating small numbers of students, this is likely to lead to poor quality solutions with respect to GPA metrics. However, in ENGGEN 403 there are around 600 students every year. This means that there are likely to be multiple students with the same gender, ethnicity and discipline. Therefore, there is still enough freedom in the problem to produce a high quality solution with respect to the GPA metrics.

Any of these constraints can be removed from the model by setting the corresponding $\beta$ value to zero. This may be desirable for smaller classes where a greater emphasis is placed on GPA.

## 3 Implementation

The above model was implemented in the SolverStudio (Mason 2013) modelling environment. SolverStudio is an add-in for Microsoft Excel. It acts as an interface between Excel and a variety of modelling languages including AMPL (AMPL 2014), GAMS (GAMS 2014), PuLP (PuLP 2014) and Gurobi (Gurobi 2014). In order to leverage the ability of SolverStudio to interface with Excel (see *Reporting*), it was decided to use the PuLP modelling language. CBC was chosen was the solver as it is the default solver for SolverStudio.

Model data is typed into cells in Excel and *named ranges* are created using SolverStudio's *Edit Data* function.

These ranges are dynamic, so that the user does not need to interact with this functionality. Adding a new row to the table will automatically adjust the named range in SolverStudio. This was done by using the following formula in the *Cell Range* field in the SolverStudio Data Items Editor instead of hard coding a fixed range:

$$= OFFSET(A2, 0, 0, COUNTA(A2 : A10000), 1) \tag{12}$$

where the variable being created is in column $A$ (so that the header is cell $A1$).

When the model is solved, SolverStudio loads the data from the named ranges into standard Python dictionaries that can be accessed in the usual way. This provides seamless interaction between the PuLP model object and the data. When the model has finished solving, the reverse occurs, and SolverStudio writes the solution into a named range on the sheet. In this manner, the user does not need to understand any python or optimisation modelling in order to use the programme.

Rather than solve to optimality, the user specifies a time limit for the solver.

The programme has been tested in Excel 2010 and Excel 2013 with SolverStudio version >0.6.x.

| UPI | Name | Gender | Ethnicity | GPA | Specialisation |
|---|---|---|---|---|---|
| | | | | | |

Table 1: Example layout of data in Excel. UPI (Unique Public Identifier) is a unique identifier for each student and forms the basis for emails at the University of Auckland.

### 3.1 Reporting

In this section we discuss the visual aids produced to help users understand the solution.

SolverStudio provides access to Excel's API (Application Programming Interface), allowing the Python code to not only create and solve the model via PuLP, but also leverage Excel's graphing functions to create a number of visualisations of the solution.

#### 3.1.1 Graphing the solution

Bar charts showing the number of students in each group were produced for each gender, discipline and specialisation. This allowed a visual means of checking the solution quality.

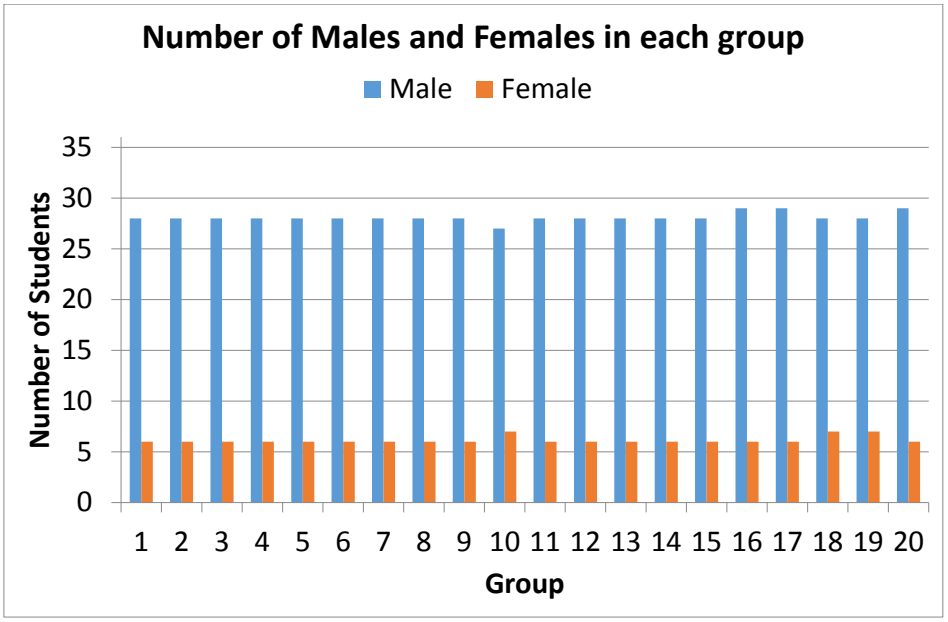


Figure 1: An example graph produced by the python model.

### 3.1.2 Creating reports

Excel's ability to save single worksheets as a PDF (portable document format) was utilised to create a final report that could be distributed to students showing their allocated group.

## 4 Results

In 2013, **500** students enrolled in ENGGEN403. There were partitioned into **20** groups.

| | | Historic | | Optimisation | |
|---|---|---|---|---|---|
| GPA | Actual | min | max | min | max |
| Group Mean | | | | | |
| Variance within group | | | | | |

Table 2: GPA metrics.

| | | Historic | | Optimisation | |
|---|---|---|---|---|---|
| Specialisation | Actual % | min | max | min | max |
| Biomedical | | | | | |
| Civil | | | | | |
| Chemical & Materials | | | | | |
| Computer Systems | | | | | |
| Electrical | | | | | |
| Engineering Science | | | | | |
| Mechanical | | | | | |
| Mechatronics | | | | | |
| Software | | | | | |

Table 3: Breakdown of the 2013 ENGGEN403 class by specialisation.

| | | Historic | | Optimisation | |
|---|---|---|---|---|---|
| Gender | Actual % | min | max | min | max |
| Male | | | | | |
| Female | | | | | |

Table 4: Breakdown of the 2013 ENGGEN403 class by gender.

| | | Historic | | Optimisation | |
|---|---|---|---|---|---|
| Ethnicity | Actual % | min | max | min | max |
| Pākehā/European | | | | | |
| Māori | | | | | |
| Pacific | | | | | |
| Asian | | | | | |
| Other | | | | | |

Table 5: Breakdown of the 2013 ENGGEN403 class by ethnicity.

## 5 Discussion

One issue with the model was that it took a long time to solve to optimality. This is due to a variety of factors, but chiefly the large number of binary variables (number of students $\times$ number of groups). Although the solver quickly (less than one minute) finds a good integer solution, it spends a long period of time the lower bound.

One reason for this is that the solution to the relaxed problem has the trivial solution where a fraction of each student of each student is allocated to each group, enabling a perfectly balanced partition. This creates a large bound gap (the distance between the relaxed problem and the optimal integer solution) for the solver to overcome.

*Another reason is the nature of the binary variables. A branch and bound approach is inefficient at increasing the lower bound at it requires near complete enumeration of the branch and bound tree (?).*

Other approaches, such as a set partitioning formulation, may improve the solution time and quality. However, given that students are unaware of the GPAs of other students in their groups, they are unlikely to perceive a small improvement in solution quality. It was for this reason, and the tight development timeframe, that other approaches were not investigated.

This paper provides a case study on how the combination of Excel, SolverStudio and PuLP enables the rapid development of non-trivial optimisation solutions. The total development timeframe, from conception to delivery of a finished product, took two weeks of part time work. The majority of this was spent creating the reporting and visualisation tools. Using this model as a guide for future problems will reduce this time even further.

It also highlights the ability for SolverStudio to abstract a large amount of complexity from the end user. The course co-ordinator needed only minutes of instruction to be able to successfully use the programme without supervision. Where the task of partitioning the students previously took two days, the Excel spreadsheet is able to achieve a more equitable result in the time it takes to make a cup of coffee.

Optimisation does not always need to be applied to large commercial problems to be beneficial. Instead, there exist a large number of problems where years of experience has culminated in a solution that is 'good enough'. Simple solutions to these problems can be created using SolverStudio in a short span of time that are user friendly and significantly reduce the work required by the user, freeing then up for more productive activities.

## Acknowledgments

The authors gratefully acknowledge Dr. Keith Adams of the Civil Engineering Department at the University of Auckland for providing valuable feedback, as well as the historical data used to validate the model. The final spreadsheet can be found on GitHub at http://www.github.com/odow/group-allocator.

## References

AMPL. 2014. A Modelling Language for Mathematical Programming. [Online; accessed 2014-10-10].

GAMS. 2014. GAMS Home page. [Online; accessed 2014-10-10].

Gurobi. 2014. Gurobi Optimisation. [Online; accessed 2014-10-10].

Mason, A. 2013. "SolverStudio: A New Tool for Better Optimisation and Simulation Modelling in Excel." *INFORMS Transactions on Education* 14 (1): 45–52.

PuLP. 2014. An LP modelling system written in Python. [Online; accessed 2014-10-10].