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 called ‘Managing a Business. As part of this course, students are partitioned into groups of around twenty-five students and given a single week to solve a large engineering or business related problem. In order to make this partitioning equitable, the students should be partitioned in a way that makes the groups as similar as possible across several factors. These include the gender and ethnic make-up of each group, as well as the number of students from each engineering discipline and the academic performance of the students in each group. A mixed integer programme was formulated to solve this problem and implemented using Excel, SolverStudio and PuLP. The model was validated against data from the 2013 cohort, and shown to improve all metrics. The Excel spreadsheet was given to the course organiser who used the model to partition the 571 students of the 2014 class into twenty-three 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, partitioning students, equitable groups

1 Introduction

Group work is an important component of many university courses that lets students learn how to work with a diverse range of people. Many students will have an ideal group in mind before beginning a group project, for example, they may wish to work with other students who are all excellent academic performers. It is also clear the people of different backgrounds bring different attitudes and skills to a group. For example, it is likely that a group of all females would tend to behave differently to a group of all males. The mix of people in a group can have a large impact on the performance of a group, and therefore, it is important for every group in a class to share the same characteristics in order to make the partitioning of students into groups equitable for all students.

We formulated an optimisation problem for the partitioning of students into equitable groups for a group project that all final year engineering students at the University of Auckland take part in. ENGGEN 403, or 'Managing a Business', runs a systems engineering group project that constitutes forty percent of each student's final course mark, and involves students being divided into groups of around twenty-five 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 and a proposal for the future of Auckland's transport network.

In 2014 the project was to write a tender to run a new public-private partnership investment agency, and for the first time, the groups were automatically allocated to groups using the solution described in this paper. We call our solution *Group-Allocator*.

This paper details the development, implementation and validation of Group-Allocator using PuLP, SolverStudio and Microsoft Excel that can automatically create equitable groups without the end user requiring knowledge of the underlying optimisation model.

1.1 Defining an equitable partition

There is no clear definition of an equitable partition, however, we can identify academic ability and demographic composition as being important factors in defining an equitable partition. 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 second is important to ensure that no one feels excluded in their group, and to ensure that each group has a diverse mix of people.

At the University of Auckland academic performance is measured quantitatively by grade point average (GPA). GPA's are measured on a 1-9 scale, where 1 represents a C- average, and 9 represents an A+ average. One way of measuring the overall academic ability of a group is by mean GPA; however, it is also important that the distribution of GPA for all the groups is as similar as possible, and so mean GPA cannot be considered alone. If only mean GPA were considered, then it may be possible to get a bi-modal split of high-GPA students and low-GPA students, which is likely to result in a very different group dynamic to one with all average GPA students. For this reason, in this paper we also seek to make the variance of GPA the same for each group.

When considering demographic composition, we considered gender, ethnicity and engineering discipline. ENGGEN 403 is a common paper to all the engineering disciplines at the University of Auckland. Each discipline has a unique set of skills and perspectives 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.

We consider this problem to be a multi-objective optimisation problem, however, we chose to formulate our model as a weighted multi-objective optimisation problem in order to create an interface for the course organiser that was simple and produced a "good enough" solution.

1.2 Previous Solution

In previous years, the job of partitioning the students into groups has been the course organisers job. The method can be considered to be a greedy heuristic, and works 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, ethnicity and mean 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: $E = \{$Pākehā/European, Māori, Asian, Other$\}$;
Genders: $K = \{$Male, Female$\}$;
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

γ_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.
α = priority given to allocating groups with equal mean compared to groups with equal variance. This should be between 0 and 1;
β^I = penalty weighting put on violating demographic constraints of type I, $I = D, E, K$;
$\hat{\mu}$ = mean GPA of all students;
$\hat{\sigma^2}$ = variance in the GPA of all students.

Decision variables

$x_{sg} \in \{0, 1\} = 1$ if student s allocated to group g, 0 otherwise;
ρ_{gi}^I = artificial penalty variable for group g, factor i, $\qquad I = \{K, D, E\}$;
μ_{max} = maximum mean GPA of any group;
μ_{min} = minimum mean GPA of any group;
σ_{max}^2 = maximum variance of GPA within any group;
σ_{min}^2 = minimum variance of GPA within any group.

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 β is necessary as some constraints are relaxed.

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

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.

$$\frac{1}{M_g}\sum_{s\in S} x_{sg}\gamma_s \geq \mu_{min} \qquad \forall g \in G \tag{4}$$

$$\frac{1}{M_g}\sum_{s\in S} x_{sg}\gamma_s \leq \mu_{max} \qquad \forall g \in G \tag{5}$$

$$\frac{1}{M_g}\sum_{s\in S} x_{sg}[\gamma_s^2-\hat{\mu}^2] \geq \sigma^2_{min} \qquad \forall g \in G \tag{6}$$

$$\frac{1}{M_g}\sum_{s\in S} x_{sg}[\gamma_s^2-\hat{\mu}^2] \leq \sigma^2_{max} \qquad \forall g \in G \tag{7}$$

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 ρ variables.

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

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

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

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 GPA variance of a group, one needs to know the square of the mean of the mean GPA of that group. 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 GPA of every group is equal to the mean GPA 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 GPA ($\hat{\mu}$). When we verified the model, we found this assumption to be valid.

2.2 The effect 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 of β are large relative to the rest of the objective function, then the effect 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 β value to zero. This may be desirable for smaller classes where a greater emphasis is placed on GPA.

3 Implementation

The 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, it was decided to use the PuLP modelling language. CBC (CBC 2014) was chosen for the solver as it is free to use so the model could be given to the course organiser without licensing issues.

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(A:A)-1, 1)
```

where the variable being created is in column A (so that the header is cell $A1$). A number of data checks are displayed to the user, for example, that there is no missing data for any students.

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 spreadsheet.

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.

3.1 Reporting

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.

Figure 1 shows an example of one of the plots produced by the Python code. In this case, boxplots of GPA are produced for each group. This enables a quick visual tool for users to understand the quality of the solution, ensuring there are no clear outliers.

In addition, 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.

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

4 Model Validation

In 2013, 586 students enrolled in ENGGEN 403. They were partitioned into 24 groups by the course organiser. To test the quality of the solutions provided by Group-Allocator, an anonymised version of the dataset was obtained from the course organiser.

There were 14 groups of 24 students, and 10 groups of 25 students. A value of 0.5 was used for α, and all β values were set to 1×10^4. The students had a mean GPA of 5.92, with a variance of 2.70. The model was solved for 120 seconds.

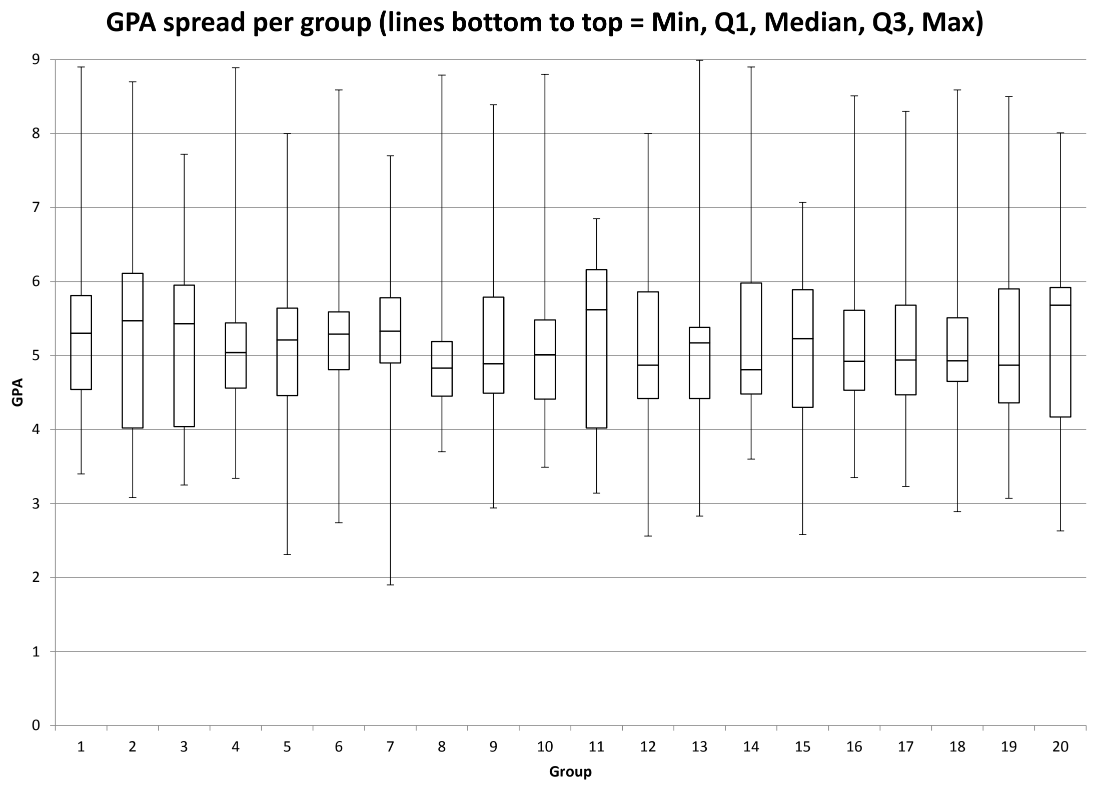

Figure 1: Example graph produced programmatically by the SolverStudio model.

There was a small number of students who did not have a recorded discipline, or were from a different degree program. They were classified in the *Other* category for discipline.

Table 1 shows the results of the partitioning for the 2013 ENGGEN 403 class. The *Class* column gives the corresponding value of all the students enrolled in the class. The *Manual* columns give the values for the groups as allocated by the course organiser, and the *Group-Allocator* columns give the values for the groups created by Group-Allocator.

Values in bold indicate that the corresponding method returned a solution with a value closer to the class value than the alternative method.

4.1 Grade Point Average

When compared to the manual partitioning, the Group-Allocator partitioning reduces the spread in mean GPA between the groups. The lowest mean GPA of any of the groups is 5.74 (compared to 5.42 for the manual solution), while the highest mean GPA is 6.05 (compared to 6.6 for the manual solution).

Similarly, Group-Allocator created groups with a larger minimum GPA variance (2.56 compared to 1.7), and a smaller maximum GPA variance (3.16 compared to 3.83) than the manual partitioning.

This improvement can also be seen visually in boxplots of the mean and variance of the GPAs of each group (Figure 2). In both cases Group-Allocator has a smaller spread than the manual solution, resulting in more similar groups.

		Class	Historic Min	Historic Max	SolverStudio Min	SolverStudio Max
GPA	Mean	5.92	5.42	6.60	**5.74**	**6.05**
	Variance	2.70	1.70	3.83	**2.56**	**3.16**
Discipline (%)	Biomedical	4.44	4.00	8.33	4.00	8.33
	Engineering Science	4.78	0.00	8.33	**4.00**	**8.00**
	Software	8.02	4.00	**12.00**	4.00	12.50
	Electrical	13.30	8.33	16.70	**12.00**	16.70
	Civil	28.70	25.00	33.30	**28.00**	**29.20**
	Mechanical	13.50	12.00	16.70	12.00	16.70
	Computer Systems	6.31	4.00	12.50	4.00	**12.00**
	Mechatronics	9.90	8.00	**12.50**	8.00	16.00
	Chemical & Materials	10.60	8.00	12.50	8.00	12.50
	Other	0.51	0.00	4.17	0.00	4.17
Gender (%)	Female	26.11	24.00	**28.00**	24.00	33.33
	Male	73.89	**72.00**	76.00	66.67	76.00
Ethnicity (%)	Pākehā / European	38.57	25.00	50.00	**36.00**	**44.00**
	Māori	3.75	0.00	12.50	0.00	**8.33**
	Asian	53.75	41.67	64.00	**52.00**	**58.33**
	Other	3.92	0.00	8.33	0.00	8.33

Table 1: Partitioned composition of the 2013 ENGGEN 403 class

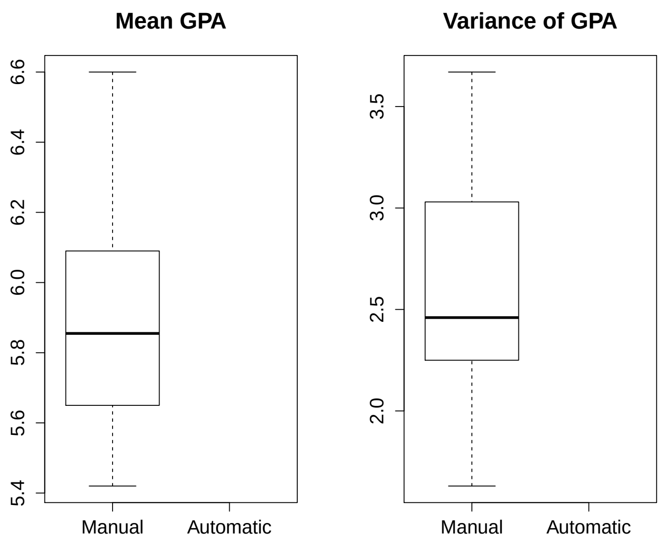

Figure 2: Boxplots of mean group GPA and variance within group GPA for manual and Group-Allocator partitioning.

4.2 Discipline

When compared to the manual partitioning, the SGroup-Allocator partition performed better in six metrics and poorer in two metrics. The manual partitioning had a smaller maximum number of software students (12% compared to 12.5%) and a smaller maximum number of Mechatronics students (12.5% compared to 16%).

In addition, there was one group in the manual partitioning with zero Engineering Science students. In the Group-Allocator solution there were at minimum 4% Engineering Science students.

4.3 Gender

When compared to the manual partitioning, Group-Allocator created a poorer partitioning of students based on gender. There was at most 28% females in a group in the manual partitioning compared to 33.3% in the Group-Allocator partitioning. There was a corresponding difference in the minimum percentage of male students.

4.4 Ethnicity

Group-Allocator generally produced both a higher minimum number of students and a smaller maximum number of students than the manual method.

5 Discussion

The results indicate that both the manual method and Group-Allocator produced balanced groups, but Group-Allocator clearly performed better with regards to mean GPA and GPA variance. The results also indicate the difficulty in comparing two solutions, as there are many different factors to consider and a single number cannot encapsulate a meaningful measure of solution quality. It could be said that a solution just needs to be "good enough", and that it is possible to judge this by checking for any outliers in the factors considered. For example, a group with all one ethnicity would clearly not be part of a good solution. The manual method and Group-Allocator both produced good solutions, but Group-Allocator provides an important advantage, in that it is fully automatic and takes 120 seconds to find a solution compared to two days with the manual method.

Group-Allocator improved almost all metrics than when compared to the manual solution, but it resulted in a poorer partitioning with respect to gender. The likely reason for this is that the course organiser placed more emphasis on balancing gender than other demographic factors. This could be overcome in future by changing the β^K value. This small decrease in gender equity between groups is offset by the more even distributions of students by discipline and ethnicity.

The model takes a long time to solve to optimality, and hence why a time limit is specified by the user. 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 CBC solver quickly (less than one minute) finds a good integer solution, it is unable to find an optimal solution. 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.

6 Conclusion

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.

This case study 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 spreadsheet 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 using the model to partition the ENGGEN 403 groups in 2014, providing the historical data used to validate the model, and for providing valuable feedback. 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].

CBC. 2014. CBC Home page. [Online; accessed 2014-10-29].

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].