The Classroom Reallocation Problem: A Case Study

Lee Safranek, Ruobing Anita Wang, Colin Exley
Department of Mathematics, Simon Fraser University, Surrey BC

Abstract

Unlike typical academic scheduling problems, we will hold the time schedule constant and only optimize the reallocation of the rooms. We provide a background of research already published, the basic three-dimensional integer linear problem, the problem with reduced dimensions and a reformulation into a linear transportation program. We specifically look at the case study of Simon Fraser University Surrey.

1 Introduction

Linear Integer Programming models have been studied over various disciplines due to its wide variety of applications. Many scheduling problems can be formulated as integer programs and is one of the major topics of research in applied optimization. In particular, scheduling problems in an academic setting have received extensive attention from researchers and for good reason. Finding solutions that allow universities to improve allocation of resources could potentially increase student enrollment or serve existing students more efficiently is of great importance. However, due to the difficulty of scheduling problems, many schedules are merely feasible solutions, and may not be the best use of an academic institution's resources. Many research papers focus on incorporating both the time table along with implementing a feasible class room schedule. For example [4, 10] discuss many of the constraints that are relevant in class room scheduling problems. Their model minimizes the cost associated with having a class being assigned to a particular room while guaranteeing each class having a room, and ensuring each room is never double booked [4, 10]. These are the main constraints that any given scheduling problem must take into account. The article also modifies the objective function to force small classes to use small classrooms [4, 10]. The main difficulty with solving the time scheduling with the room schedule is the computational time [9]. With the time schedule being implemented with room schedule, the problem is not able to be treated with each day being represented as its own independent problem [2]. For a typical academic institution, this problem can be very complex, and difficult to solve [5]. For further research on university scheduling papers, we refer to [1, 3, 7]. Unlike the standard course scheduling problems studied in literature, we assume that a course schedule is already given and we want to make a reallocation of rooms while keeping the time schedule unaltered and minimizing a prescribed objective function. Section 2 describes the main ideas behind the room reallocation problem. Section 3 then describes different representations of the model which is then followed by the SFU Surrey's Case Study. Section 5 reformulates the integer program into a transportation problem with out the integer constraint with a proof that the minimum of the objective function remains preserved. Finally, section 6 will suggest future areas of possible research topics.

2 Problem Description

As an initial model, we will only look at reallocating classroom assignments and the resulting effects on the academic institution's resources. Minimizing the slack between room sizes and course enrollment is of particular interest to an academic institution because more space will be available to accompany more students and/or more courses. Slack represents the remaining space between enrollment and room capacity. Renting rooms to outside organizations provides the institution with revenue, consequently maximizing the institution's renting ability is also of interest. Minimizing the rental cost of used rooms for courses will allow for an increase in potential revenue from renting outside the institution. We will also look at grouping faculty courses by floor. For our case study, scheduling data is taken from Simon Fraser University (SFU) Surrey. We aim to minimize renting costs, slack between room size and course enrollment, and faculty grouping by formulating the problem as an integer linear program. The problem is solved for one week, with each day being divided into fifteen time slots representing the possible times the course can be offered. The time slots that a given course will be offered is fixed, thus results in strictly a classroom reallocation problem. We will use binary variables to indicate matching between rooms, courses and time slots. There will be constraints to ensure the capacity of the room is sufficient for the paired course. There will also be constraints ensuring only one course is paired with a room for each time slot. The final requirement is for each course to have a viable matching room.

3 Integer Programming Model

For simplicity, we first consider a three index model where classrooms are represented by i, time slots by j and course number by k. The decision variable X_{ijk} is defined as:

$$X_{ijk} = \begin{cases} 1 & \text{if course } k \text{ is assigned to room } i \text{ at time slot } j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Let the following given data be defined as follows:

α_k denote number of hours course k is offered per week.

e_k denote the enrollment size of course k.

R_i denote the room capacity of room i.

C_i denote the renting cost of room i for one time slot.

F_i denote the floor of room i.

F_k denote the desired floor of course k.

N denote the total number of courses.

M denote the total number of rooms.

λ_1, λ_2, λ_3 be non-negative weights.

3.1 General Classroom Assignment Problem

This gives the following integer programming formulation for the class scheduling problem.

Minimize:

$$\sum_{i=1}^{M} \sum_{j=1}^{75} \sum_{k=1}^{N} \left(\lambda_1 C_i + \lambda_2 \left(R_i - e_k\right) + \lambda_3 \left|F_i - F_k\right|\right) X_{ijk} \tag{2}$$

Subject to:

$$\sum_{i=1}^{M}\sum_{j=1}^{75} X_{ijk} = \alpha_k, \forall k \tag{3}$$

$$e_k X_{ijk} \leq R_i, \forall i, j, k \tag{4}$$

$$\sum_{k=1}^{N} X_{ijk} \leq 1, \forall i, j \tag{5}$$

$$X_{ijk} \in \{0, 1\} \tag{6}$$

The constraints are explained as follows:

1. Our objective function balances three terms. The first seeks to minimize the slack between room capacity and enrollment size. The second minimizes room costs which consequently maximizes potential renting profit from outside organizations. The third term is a soft constraint to group faculties on desired floors.

2. The number of hours a course is offered must be conserved for any feasible matching.

3. The course enrollment must be less than the room capacity for every used room.

4. No feasible matching can have more than one course in one room at one time.

5. X_{ijk} is a boolean variable.

Due to the inherit difficulty of scheduling problems, we keep the time slots fixed. This gives the following additional constraint:

Let J_k be the set of used time slots for course k.

$$\sum_{i=1}^{M} \sum_{j \notin J_k} X_{ijk} = 0, \forall k \tag{7}$$

This constraint ensures that unused time slots remain unused. Furthermore, courses that are more than one timeslot in length should remain in the same room. This gives:

$$X_{ijk} = X_{ij+1k} = \cdots = X_{ij+nk} \text{ if } J_k = [j, j+1, \ldots, j+n] \tag{8}$$

3.2 Reducing Dimension

As a result of fixing time slots, we are able to change the three-dimensional problem into one with only two dimensions. Since a course can be assigned to only one room per day, we are able to reformulate the problem into 5 independent problems, treating each day independently. The result allows the following decomposition:

$$X_{ijk} = T_{jk} Y_{ik} \tag{9}$$

$$T_{jk} = \begin{cases} 1 & \text{if course } k \text{ is offered during time slot } j \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$Y_{ik} = \begin{cases} 1 & \text{if course } k \text{ is assigned to room } i \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Since T_{jk} is fixed, the decomposition translates to the following reformulation:

Minimize:

$$\sum_{i=1}^{M}\sum_{j=1}^{15}\sum_{k=1}^{N}\left(\lambda_1 C_i + \lambda_2\left(R_i - e_k\right) + \lambda_3\left|F_i - F_k\right|\right)T_{jk}Y_{ik} \tag{12}$$

Subject to:

$$\sum_{i=1}^{M} Y_{ik} = 1, \forall k \tag{13}$$

$$e_k Y_{ik} \leq R_i, \forall i, k \tag{14}$$

$$\sum_{k=1}^{N} Y_{ik}T_{jk} \leq 1, \forall i, j \tag{15}$$

$$Y_{ik} \in \{0, 1\} \tag{16}$$

The constraints are explained as follows and enforce the same properties as the general problem:

12. Our objective function balances the same three terms as the original model.

13. The course can only be in one room per day and forces courses over an hour long to remain in one room.

14. The course enrollment must be less than the room capacity for every

used room.

15. No feasible matching can have more than one course in one room at one time.

16. Y_{ik} is a boolean variable.

With the implementation of the fixed values T_{jk}, the resulting model has far fewer variables and is much more practical to implement solvers.

4 Case Study

We are given a feasible course schedule from SFU Surrey campus. Our task is to improve on the schedule under certain assumptions and constraints. There is freedom in choosing the weights depending on which terms are deemed more valuable. The model will have the basic scheduling constraints as well as additional constraints tailored to the classroom reallocation problem. We will assume that the courses of which require computer labs will be fixed as all the computer labs in SFU Surrey have the same capacity, the same cost for renting and are located on the same floor. Therefore, there will be no possible improvement be swapping between the labs. The data and model will be implemented in Microsoft Excel. The initial schedule is mined by an algorithm written in Visual Basic to format the scheduling data to be consistent with our model. Open Solver, a plugin for Excel that allows for larger problem sizes, will use the simplex algorithm and as well as branch and bound technique for optimizing the schedule [6]. Currently the time and room schedule at SFU Surrey is created by hand by a full time employee. Their goal is simply to create a feasible class room location schedule. A one week snap shot of the current schedule has been provided by SFU Surrey and was added into excel to create an electronic version. Not including the computer labs, there are 31 classrooms, and including tutorials, there are 524 courses. There are only 3 floors which contain classrooms. Typically, science faculty reside on the 2 and 3 floor, and art and social sciences are found on the 5 floor. The majority of rooms hold between 24-30 students. There are additional rooms that have a capacity of 100, 150 and 200 students. As a rough estimate, for each day there are approximately 100 courses, 31 rooms resulting in roughly 3100 variables. This is coupled with approximately 3800 constraints.

4.1 Choosing Weights in the Objective Function

The objective function has three undetermined weights: $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]$. The choice of these weights determine the importance of each term in the objective function. Choosing weights admits freedom in how SFU Surrey would prefer to allocate their resources. Since the runtime was under five seconds for each day, we were able to run the model with various sets of weights. The tables display the results in the following manor. It looks at each day, and the resulting objective function of the given data before and after optimization. The final three columns show non-scaled values of the three terms of the objective function after optimization.

Table 1: $\boldsymbol{\lambda}$ = [1,0,0]

Day	Before	After	λ_1 Term	λ_2 Term	λ_3 Term
Monday	31446	20671	20671	5531	128
Tuesday	29052	19502	19502	4900	81
Wednesday	34167	21767	21767	5777	112
Thursday	26102	17177	17177	4301	61
Friday	31675	18325	18325	4674	93

Table 2: $\boldsymbol{\lambda}$ = [0,1,0]

Day	Before	After	λ_1 Term	λ_2 Term	λ_3 Term
Monday	7601	5464	21171	5464	115
Tuesday	6890	4869	19752	4869	81
Wednesday	8164	5691	21867	5691	115
Thursday	6312	4256	17477	4256	64
Friday	7490	4604	18475	4604	91

Table 3: $\boldsymbol{\lambda}$ = [0,0,1]

Day	Before	After	λ_1 Term	λ_2 Term	λ_3 Term
Monday	148	43	30921	7282	43
Tuesday	122	25	27252	6510	25
Wednesday	140	49	29517	7112	49
Thursday	128	31	20677	5035	31
Friday	140	38	28075	6559	38

Table 4: $\boldsymbol{\lambda}$ = [1,1,1]

Day	Before	After	λ_1 Term	λ_2 Term	λ_3 Term
Monday	39195	26261	20671	5474	116
Tuesday	36064	24447	19502	4876	69
Wednesday	42471	27568	21767	5701	100
Thursday	32542	21500	17177	4266	57
Friday	39305	23020	18325	4609	86

Table 5: $\boldsymbol{\lambda}$ = [0.01,1,5]

Day	Before	After	λ_1 Term	λ_2 Term	λ_3 Term
Monday	8655.5	6110.5	22246	5533	71
Tuesday	7790.5	5369.3	19927	4900	54
Wednesday	9205.7	6364.7	22667	5733	81
Thursday	7213	4705.3	17427	4276	51
Friday	8506.8	5161	1900	4626	69

Tables 1 - 5 show that, regardless of the weights, there is great possibility for improvement. We first looked at only one term of the objective function in order to see which terms were closely related. Typically larger classrooms are also more costly and Tables 1 and 2 show the terms scaled by λ_1 and λ_2 are related as expected. If we only optimize clustering faculty, Table 3 shows the cost and slack in rooms become large. Having equal weights results in first optimizing cost, then slack, and lastly faculty groupings which is displayed in Table 4. We modified the weights as having a course on the undesired floor should not be equal to having one student slack or one dollar change in room cost. We chose to have the floors weighted heavier by a factor of 5, and reduced the cost term by a factor of 100. Reducing cost is less important to a University as renting potential is insignificant compared to potential student enrollment. When a class is on a undesired floor, it affects more than one student, and hence requires a heavier weight. Comparing Tables 4 and 5 shows that there is minor relative difference between the λ_1 and λ_2 terms yet a more significant change in the λ_3 term.

4.2 Course Movement

Once we decided on the weights for $\boldsymbol{\lambda}$, we analyzed the movement of the courses. The movement of courses with $\boldsymbol{\lambda}$ = [0.01,1,5] was widespread. Out of the 524 courses, only 37 courses remained in the room allocated by the original schedule. This indicates the original schedule was a poor allocation of school resources. There were two main types of courses that remained stationary. Firstly, were the large courses, which had enrollment of over 100 students. These remained stationary since there are only a few large classrooms available during peak time slots. With the restriction of the number of classrooms available, there are no other feasible locations to be reallocated to. The other group of courses that remained the same were the courses with small enrollment, typically 12 students or fewer. These courses were also already located on the desired faculty grouping. Since these courses are already in the rooms with the smallest capacity, and on the desired floor, moving these courses would not improve the objective function. The majority of the courses that were reallocated where courses with enrollment between 20-50 students. This is due to the fact that the majority of the rooms at SFU Surrey have a capacity between 24 and 60 students. The movements were either between rooms of the same capacity but different floors, or the original schedule had placed the course in too large of a room when other smaller rooms were unused. In both cases, classroom reallocation has greatly increased SFU Surrey's allocation of resources.

5 Relaxation to LP

One of the difficulties of dealing with integer programs is the inherit computational complexity, NP-Hard [8]. Due to the special structure of our integer program, it can be inflated and reformulated in such a way that the integer constraint can be relaxed. The first step is to inflate the number of variables so that individual rooms have states according to time slots. Let i be an inflated index from before to denote a room and time slot.

Let S_i denote a supply node with supply 1.

Let D_j denote a demand node. The demand of each node is 1.

Let A_{ik} be the same constant as the objective function in the original model.

The resulting model is: Minimize

$$\sum_{i=1}^{15M} \sum_{k=1}^{N} A_{ik} X_{ik} \tag{17}$$

Subject to:

$$\sum_{i=1}^{15M} X_{ik} = D_k, \forall k \tag{18}$$

$$\sum_{k=1}^{N} X_{ik} \leq S_i, \forall i \tag{19}$$

$$X_{ik} \geq 0, \forall i, k \tag{20}$$

The classroom reallocation problem is now an unbalanced transportation problem. Balancing the problem requires creating a fictitious demand node for the excess supply at no extra cost. This gives:

$$D_{k+1} = \sum_{i=1}^{15M} S_I - \sum_{k=1}^{N} D_k \tag{21}$$

$$C_{ik+1} = 0, \forall i \tag{22}$$

Since classroom times are fixed, many possible matchings between supply and demand nodes must be eliminated. This is overcome by assigning the cost of using these edges to be higher than our objective function evaluated at the given feasible solution. This ensures that an optimal solution will not use these edges and preserves the minimizer between formulations.

Proof. Let X_0 be given initial feasible solution.

Let $f(X)$ be the objective function of transportation formulation.

Let $g(X)$ be the objective function of the reduced dimension formulation.

Let S_0 be the feasible matchings from reduced dimension formulation.

Let S_1 be the feasible matchings from transportation formulation.

The transportation formulation contraints ensure that each course is offered and there is no overlap between rooms. Since the constraint set is a proper subset of the reduced dimension formulation it follows: $S_0 \subset S_1$ and $S_1/S_0 \neq \emptyset$.

$\forall X \in S_1/S_0$ let the associated constant of each violated matching, course offered at wrong time or in too small of a room, in the objective function $g(X)$:

$A_{ik} = g(X_0) + \epsilon$ where $\epsilon > 0$

$\Rightarrow \forall x \in S_1/S_0, g(X) > g(X_0)$

In the reduced dimension formulation the objective function is given by:

$$f(Y) = \sum_{i=1}^{M} \sum_{j=1}^{15} \sum_{k=1}^{N} \left(\lambda_1 \left(R_i - e_k\right) + \lambda_2 C_i + \lambda_3 \left|F_i - F_k\right|\right) T_{jk} Y_{ik} \quad (23)$$

Since for $x \in S_0$ credit hours must be preserved,

$$\sum_{j=1}^{15} T_{jk} = \alpha_k \quad (24)$$

$\Rightarrow$ Summing over j gives

$$f(Y) = \sum_{i=1}^{M} \sum_{k=1}^{N} \alpha_k \left(\lambda_1 \left(R_i - e_k\right) + \lambda_2 C_i + \lambda_3 \left|F_i - F_k\right|\right) Y_{ik} \quad (25)$$

Coefficients in $g(X)$ are the same for every room. For $X \in S_0$, coefficients are repeated α_k times for every course . Deflating the index i in $g(X)$ gives:

$$\forall X \in S_0, g(X) = \sum_{i=1}^{M} \sum_{k=1}^{N} \alpha_k \left(\lambda_1 \left(R_i - e_k\right) + \lambda_2 C_i + \lambda_3 \left|F_i - F_k\right|\right) X_{ik}$$

(26)

Since S_1 is nonempty there exists an optimal solution X^*

Since X^* optimal and $X_0 \in S_1$, $g(X^*) \leq g(X_0) < g(X_0) + \epsilon$

$\Rightarrow X^* \notin S_1/S_0 \Rightarrow x^* \in S_0$

Since $f(X) = g(X)$ for $x \in S_0$ and $X^* \in S_0 \Rightarrow X^*$ is a minimizer of $f(X)$

$\Rightarrow$ Minimizer is preserved between formulations $\square$

The basic idea of the proof is to take advantage of the feasibility of the problem. In the optimal case, the objective function will be less than or equal to the given feasible solution. By inflating the value of the objective function for undesired matchings, we are able to reduce the number of constraints, specifically the integer constraint.

Although the formulation into a transportation problem gives 15 times more variables, it's main advantage is that all extreme points of the feasible set are integer valued given that row and column sums are integers, see [8] for a proof. This allows the relaxation of the integer constraint for suitable choices of $\boldsymbol{\lambda}$. Due to the small run-times for solving the classroom reallocation problem with the integer constraint, this transportation formulation was not explored further to the small possible decrease in runtime. However, for larger problems the transportation formulation may cut down the amount of runtime significantly due to the nice properties and algorithms available for transportation problems.

6 Future Research

The existence of fast algorithms for the resulting transportation problem motivates separating the full classroom assignment problem into the time scheduling and classroom reallocation phases. Once a time schedule is created, the classroom reallocation phase will move rooms to improve use of resources. In theory the disjoint problem will not provide the optimal solution, however the solution of the disjoint problem is more practical due to the significant decrease in the number of variables. The large slack in the case study combined with the large number of reallocations motivates the assumption that feasibility should be conserved in the disjoint problem. A potential addition to the model is a change in objective coefficients to penalize movement of course locations between separate days. Although run times with the integer constraint were very low for the SFU Surrey Case Study, determining the size when the transportation reformulation is more efficient is of particular interest. More constraints can be added to the transportation formulation by making the associated cost very high. Adding constraints that involve course needs like technology or specific information about a room can be incorporated with relative ease making this formulation a very practical model.

7 Conclusion

The case study has shown that SFU Surrey has been poorly allocating their resources when it comes to class room scheduling. Much research has been focused on the classroom scheduling problem that incorporates both the time and room schedule as a whole. We looked at the second phase of the two-phase scheduling problem, where the time schedule is given. As a result of the SFU Surrey being a relatively small campus, we were able to run the solver with various objective functions. This allows for SFU Surrey to modify their resource allocation in a more optimal manor. Furthermore, the movement between rooms was widespread showing the freedom in the room allocation problem. Additional constraints can also be added to create a more accurate model. The small runtime, along with the flexibility of the model makes this a practical tool for academic institutions. The entire set of results, including initial solution and solver is available by email.

Acknowledgments:

We would like to thank Jane Hawkins for providing the current SFU Surrey classroom scheduling data. This research work was carried to fulfill partial requirements for the course Math402W - Industrial Mathematics Projects taught by Abraham Punnen.

References

[1] David W. Ashley. A spreadsheet optimization system for library staff scheduling. *Computers and Operations Research*, 22(6):615 – 624, 1995.

[2] Natashia Boland, Barry D. Hughes, Liam T.G. Merlot, and Peter J. Stuckey. New integer linear programming approaches for course timetabling. *Computers and Operations Research*, 35(7):2209 – 2233, 2008. ¡ce:title¿Part Special Issue: Includes selected papers presented at the ECCO'04 European Conference on combinatorial Optimization¡/ce:title¿.

[3] John J. Dinkel, John Mote, and M. A. Venkataramanan. An efficient decision support system for academic course scheduling. *Operations Research*, 37(6):pp. 853–864, 1989.

[4] M. Ehrgott. Multiobjective optimization: Association for the advancement of artificial intelligence. pages 47 – 57, 2008.

[5] Perry Fizzano and Steven Swanson. Scheduling classes on a college campus. *Computational Optimization and Applications*, 16:279–294, 2000. 10.1023/A:1008720430012.

[6] Daniel Fylstra and Leon Lasdon. Design and use of the microsoft excel solver. *Interfaces*, 28(5):29 – 55, 1998.

[7] Rhydian Lewis. A survey of metaheuristic-based techniques for university timetabling problems. *OR Spectrum*, 30:167–190, 2008. 10.1007/s00291-007-0097-0.

[8] David Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley Publishing Company, Inc.

[9] S.A. MirHassani. Improving paper spread in examination timetables using integer programming. *Applied Mathematics and Computation*, 179(2):702 – 706, 2006.

[10] J. Nakasuwan, P. Srithip, and S. Komolavnij. Class scheduling optimization. *Thammasat International Journal of Science and Technology*, pages 88–98, 1999.