

# Chapter 2

## Formalizing the Problem

### 2.1 Introduction

The existing research on school choice and course allocation would seem to suggest that for our problem, which only features one-sided preferences as opposed to preferences on both sides of the market, we follow a fair division approach similar to that of Budish's one-sided matching. However, given the practical application of our problem as a mechanism for assigning every student to one specific class, we determined that constructing a notion of stability makes more sense than evaluating envy in the matching. Our solution is driven by the goal of assigning HUM classes that are balanced both in size and student interests, so we want to minimize the ability of students to swap with each other post-assignment as much as possible so as not to disrupt the balance. If no student wishes to swap with another student, then we can consider the assignment stable. This is the framework with which we shall formalize our model.

### 2.2 The Model

The basic components of our model are the two sides of our market: a set of students,  $S$ , and a set of HUM class sections, each of which have a certain number of seats. We will call the set of all seats  $C$ . Though we are working in a bipartite setting, only one side of our market has preference over the other (i.e. students prefer certain classes, classes are inanimate and therefore indifferent to students as far as we know). Students also do not necessarily have ranked preference over every single available section; they likely equally do not want to take any sections that conflict with their other selected courses. So, preference lists will be incomplete, with those that students cannot take not being included at all. We will denote the preference list for student  $s \in S$  as  $P_s$ .

The last aspect of our problem we have to formalize is the diversity constraint. Students can be divided into three types based on the way Reed College divides

majors into groups: Group 1 for language, arts, and literature, Group 2 for social sciences, Group 3 for math and lab sciences. We will refer to these types as subsets of  $S$ :  $A, B$ , and  $C$ , where  $A \cup B \cup C = S$ . To implement the HUM philosophy, which is that the class provides an opportunity for students of all knowledge backgrounds and academic interests to discuss relevant texts, we want to ensure a balanced spread of each type of student between sections. This means avoiding scenarios such as all chem 101 students with the same lab being placed in the same section due to the overwhelming similarity in their schedules. This will come in the form of quotas or caps on the number of seats per section that can be assigned to each type of student, so each  $c \in C$  will be divided evenly into four types of available seats: A, B, C, and free seats to allow for some flexibility.

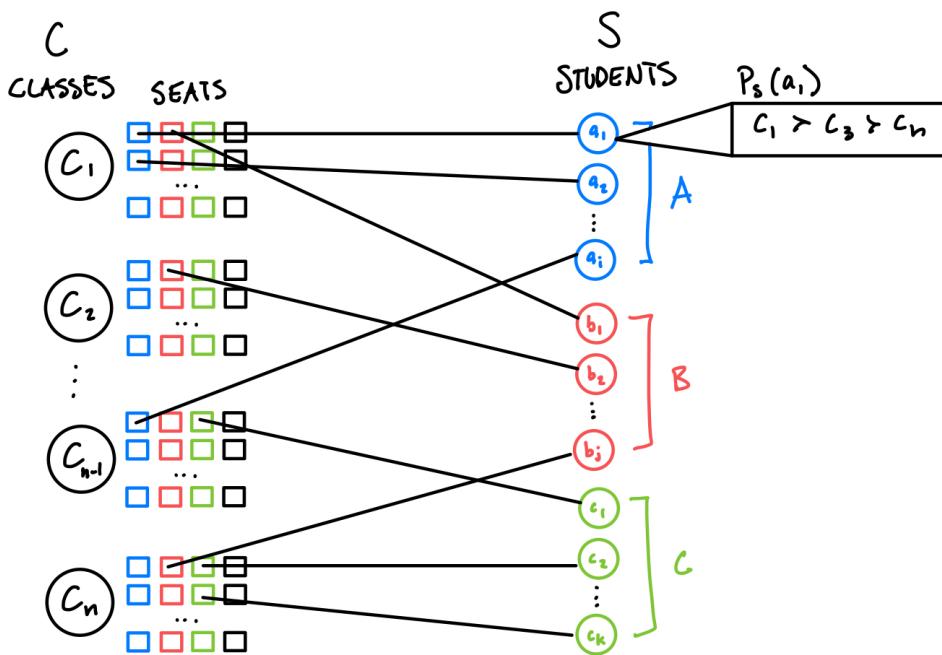


Figure 2.1: Graph illustrating generic model

## 2.3 The Challenge of Stability

Notably, “stability” in our case adopts a different meaning than in a traditional stable matching problem. Because we are working with one-sided preferences, we do not consider “cheating” as a violation of stability between students and classes, but rather between students themselves. So, stability can more formally be defined by saying there exists no blocking pair between student  $a$  and student  $b$  such that  $\mu(b) >_a \mu(a)$  and  $\mu(a) >_b \mu(b)$ .

**Theorem 2.1.** *There is not always a stable matching with quotas (caps) and incomplete preferences.*

*Proof.* Let  $C = \{c_1, c_2\}$ , where there are two types of student  $\alpha$  and  $\beta$  such that each  $c \in C$  can have at most one student of each type. Let  $S = \{A1, B1, A2, B2\}$  where students  $A1, A2$  are of type  $\alpha$  and  $B1, B2$  are of type  $\beta$ . Let the preference lists be as follows:

$$\begin{aligned} A1 : & c_1 \\ B1 : & c_2 \\ A2 : & c_1 \succ c_2 \\ B2 : & c_2 \succ c_1 \end{aligned}$$

In this case, regardless of the order of matching or algorithm, there is no stable matching. The  $A$  students will both wish to be matched to  $c_1$ , which only has one  $\alpha$  spot, and both  $B$  students wish to be matched to  $c_2$  which only has one  $\beta$  spot. So, when the first round of matching is done, that means that there is still an open  $\beta$  spot for  $c_1$  and an open  $\alpha$  spot for  $c_2$ , which the unhappy  $A$  and  $B$  students can then switch into on their own, respectively.  $\square$

Defining stability in this way is, then, too strict. Recall that even determining whether there is a stable matching or not is NP-complete in many scenarios similar to this [1]. So, we need to relax our constraints to allow for some notion of stability. For now, let's simplify the problem so we can build back up to a working definition for stability. At a basic level, before we even think about stability, we want to work in an environment where we can be sure an algorithm would respect student preferences, such that no student is assigned to a class with which they have a time conflict.

We can start by looking at an environment where we have 0/1 preferences and therefore no notion of stability. Let  $C$  denote the set of seats over all classes, and  $S$  the set of students. We will put lower bounds on the seats, such that the number of seats in each class will be reserved evenly among each type of student, with an extra group of seats that is free to any student. We will let  $C_r$  be the set of all reserved seats. In this case, we have an environment well-suited to the weighted graph matching problem. We can say  $G = (V, E)$  such that  $V = S \cup C$  and each edge goes from some  $s$  to some  $c$  if and only if  $c \in P_s$ , where  $P_s$  is the preference list of  $s$ . Each edge  $e$  has some weight  $w(e)$ . Let the weight be as follows:

$$w(e) = \mathbf{1}\{\text{if seat } c \text{ is reserved for } s\text{'s type}\} + \mathbf{1}\{\text{if student } s \text{ prefers class } c\}.$$

**Theorem 2.2.** *If a matching that respects student preferences and seat reservations exists, it will be the max-weight graph where  $w(\mu) = |S| + |C_r|$ .*

*Proof.* Let  $\mu$  be a matching that assigns every student to a seat they prefer while also filling reserved seats first, such that all reserved seats are filled before the overflow seats. By construction of the weights, we can determine a lower and upper bound for  $w(\mu)$ .

1. **Lower bound of  $\mu$**  Each edge exists only if the student prefers that seat, so every edge in  $\mu$  contributes at least 1. By our assumption of characteristics

of  $\mu$ , every student in  $\mu$  got a seat they prefer, so summing over all students contributes at least  $|S|$  to  $w(\mu)$ . We also assumed each reserved seat was assigned to a student in  $\mu$ , which adds an extra +1 to the weight for the edge from each of those reserved seats. Summing over all reserved seats gives an additional contribution of  $|C_r|$ , because the rest of the seats are still 1-weight edges. Therefore, we can bound  $\mu$ :

$$w(\mu) \geq |S| + |C_r|.$$

- 2. Upper bound of  $\mu$**  From our definition, we know each edge weight is at most 2. Similarly to the lower bound, we know  $\mu$  has  $|S|$  edges because each student was matched to one of their preferences. However, because only the reserved seats contribute an additional weight to an edge, we know that the number of 2-weight edges  $\leq |C_r|$ . So, the maximum summed weight of all edges must be  $2|C_r| + |S| - |C_r|$ , representing the maximum number of 2-weight edges plus all students assigned to an overflow seat. Note that is equivalent to  $|S| + |C_r|$ . Therefore, we can bound  $\mu$ :

$$w(\mu) \leq |S| + |C_r|.$$

So,  $\mu$  is a maximum-weight graph with a lower bound of  $|S| + |C_r|$  and an upper bound of  $|S| + |C_r|$ , so it must be that a matching that fully respects student preferences and seat reservations has a weight  $w(\mu) = |S| + |C_r|$ .  $\square$

Theorem 2.2 shows us that using the max-weight algorithm, we can assign each student to a class they prefer while respecting diversity if such a matching exists. From here, we can begin to reintroduce stability. While it may be too computationally difficult to find stability between every student, perhaps we can start by comparing just students of the same type. We are introducing a new notion of stability here, where we will consider a matching stable for our purposes if it exhibits *intra-group stability*. This means that for any type of student, that student does not form a blocking pair with another student of the same type.

Formally, a matching  $\mu$  is stable if, for any two students  $a_1$  and  $a_2$  of type  $A \subset S$ ,  $\mu(a_1) >_{a_1} \mu(a_2)$  or  $\mu(a_2) >_{a_2} \mu(a_1)$  such that there exist no intra-group blocking pairs.

For now, inter-group instability, or blocking pairs involving students of different types, does not interfere with our notion of stability. Though we would ideally like to guarantee stability between all students, looking at the problem through the lens of intra-group stability allows us to give some theoretical guarantee that helps minimize the amount of swapping that will occur in an assignment while remaining computationally feasible, as opposed to turning to less-reliable heuristic solutions.