Final Year Project Allocation For The Chemistry Department at the University of Bath

Robert Cobb

July 2018

1 Introduction

This document outlines the work done and the mathematical under pining for the Chemistry allocation system at the University of Bath. Chemistry students are required to take on a project in their final year, an aptly named "final year project". The student's projects are supervised by a supervisor provided by the department. This report studies and details the system that allocates student to their project supervisors.

2 Current System

Currently (2016), each student fills out a paper slip detailing their preferences over the supervisors they wish work with. Lectures from the department then sit down with all of these papers and, over a day, allocate the students. Supervisors make their preferences informally during the meeting. This system is a time confusing and, (the author has it on good sources) mentally taxing task for all involved.

3 Aims

The aim of this project is to change the system to achieve the following goals:

- Digitise the process
- Remove lectures preferences over students to make to process fairer for students
- Make the process more mathematically rigorous
- Ensure that no supervisor is allocated a full load of natural science students
- Try to produce an even distribution of students across the 4 topic areas of chemistry (Physical, Organic, Inorganic and Computational)

4 Phase 1

In 2017 phase 1 of the project was implemented. The output of phase 1 was a piece of software that would allocate students to supervisors based off of student choices. Phase 1 made changes to the data collection as well as the algorithm that performed the allocation.

4.1 Data collection

As part of phase 1, students were asked to fill out an online Google form. The students were asked to pick 4 ranked supervisor choices as well as to specify their specific course (This is used to determine if the student is a natural science student). The Google form software then generates a spreadsheet of the data gathered.

The Chemistry department comes up with a separate spreadsheet that details the supervisors, their capacities, and their topic areas.

4.2 Allocation

The phase 1 software looks at the 2 spreadsheets, interprets the data, creates a network flow diagram that is solved using a max flow-min cost algorithm and finally reinterpreted in the context of the spreadsheets.
Mathematically:

- A set of Students S.
- A set of Supervisors T.
- A set of Topic areas A.

$$A = \{"physical","computational","inorganic","organic"\}$$

- An integer N_p that denotes the number of supervisor choices made by each student $s \in S$. Each student makes the same number of choices. For phase 1:

$$N_p = 4$$

- A function, f_1, $f : (S, N) \to T$ that maps a student and a positive integer n, $n \leq N_p$ to a supervisor that is that student's given n^{th} ranked choice. It is useful to define another function related to this one $f_{1.2}$, $f : (S, T) \to N$ that maps a student and supervisor pair to the choice index for which the student picks the supervisor. The function is undefined for pairs where the student did not pick the supervisor.
- A function, f_2, $f : T \to Z^*$ that maps a supervisor to a non negative integer that is the capacity of the supervisor - the maximum number of students that can be allocated to that supervisor.

- A function, $f_3, f : T \to A$ that maps a supervisor to their topic area.
- A function, $f_4, f : S \to \{\text{true, false}\}$. This function describes whether a student is a natural science student or not. For a given student s, this function will return true if the student is a natural science student, false otherwise. (The program does this by looking at the student's course and comparing to a list of known natural science courses. If the student takes a natural science course they are a natural science student, otherwise they are not.)
- A function, f_5, $f : N \to Z^*$ that maps a given choice index, n, $n \leq N_p$ to the cost of that matching. The phase 1 software defines $f_5(x)$ as follows:

$$\begin{cases} 1 & x = 1 \\ 2 & x = 2 \\ 10 & x = 3 \\ 15 & x = 4 \end{cases}$$

- p, $p \in N, p \leq 100$ that denotes the maximum percentage of students (of the global population) to be allocated to a specific topic area. For example, if p = 50, no more than 50% of the student population will be allocated to any of the given topic areas $a \in A$. This aims to allow the end user performing the allocation to control the distribution of students across topic areas by limiting the maximum percentage of students allcoated to one specific area.

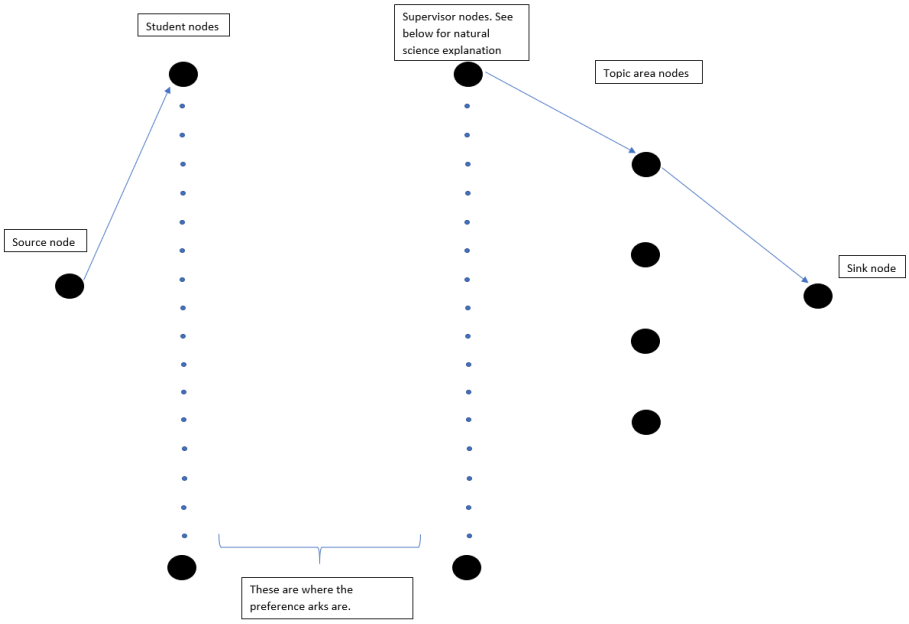
The network for phase 1 is defined as follows:

- A source node
- A sink node
- A student node $\forall s \in S$
- 2 supervisors nodes $\forall t \in T$. A normal node and a natural science node. For more explanation on this see 4.2.1
- A topic area node $\forall a \in A$
- An arc from the source node to every student node each with a capacity of 1 and cost of 0.
- An arc from every student node to every supervisor node iff that student chose that supervisor as one of their preference choices.

$$\begin{aligned} &\forall(\text{s, n}) \in S \times \{x | x \in Z, 0 < x \leq n_p\} && \text{(For phase 1, } S \times \{1,2,3,4\}\text{)} \\ &\qquad \text{let } t = f_1(s,n) && \text{(let t be the given student's n'th choice)} \\ &\qquad \text{Add arc } (s, t, f_5(n)) \end{aligned}$$

Note: If the student is a natural science student, the arc is from the student to the supervisor's natural science node, otherwise the arc is from the student to the supervisor's normal node.

- An arc from every natural science supervisor node to the normal supervisor node with capacity $f_2(t)$-1 and cost of 0. This is what ensures that a supervisor will never get a full load of natural science students. See section 4.2.1.
- An arc from each normal supervisor node to the relevant topic area node. $\forall t \in T$ draw an arc between normal node t and $f_3(t)$ with capacity $f_2(t)$ cost 0
- An arc from each topic area node to the sink node with capacity $round((p/100)\times |S|)$ and cost 0. This arc is what limits the number of students per topic area.



The flow in the network represents a matching and the cost represents the how good that matching is. The outcome of the match is a set of tuples, M

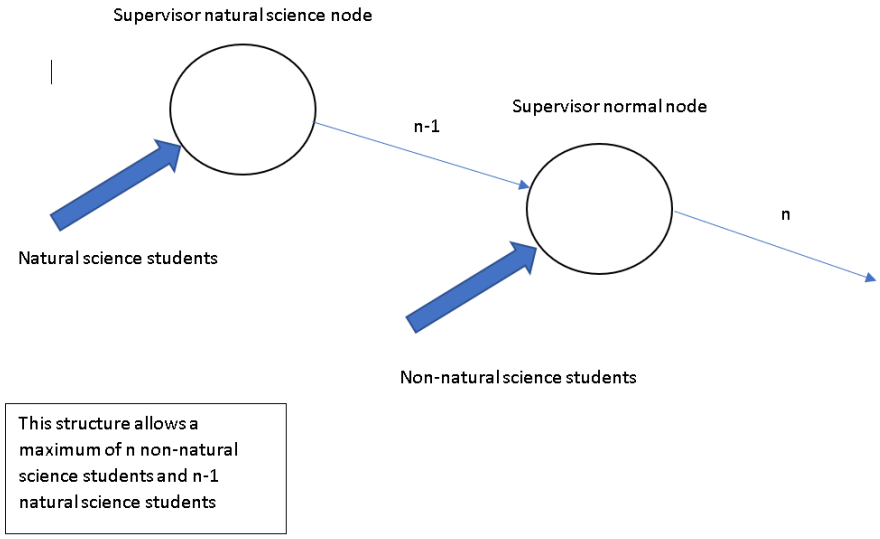
$$M = \{(student, supervisor) | \text{student is allocated to supervisor}\}$$

The max flow-min cost algorithm finds the allocation with the highest number of matches (largest cardinality of M), then reducing the cost function:

$$\sum_{(s,t)\in M} f_5(f_{1.2}(s,t))$$

4.2.1 Explanation of 2 supervisor nodes

The network description of the problem contains 2 nodes for each supervisor. A "normal" node and a "natural science" node. The reason for this is to satisfy the aim that no supervisor can be fully saturated with natural science students. For example, Professor Adam can take a total capacity of 4 students. The requirement stipulates that no more than 3 students can be natural science. This is achieved by splitting professor Adam's nodes into two and then linking them together with an arc from the natural science node to the normal node. The arc has no cost but has a capacity of 3. Students who want Adam to supervisor them are connected to one of the two nodes, if they are a natural science they are connected to the natural science node, else they are connected to the normal node. The splitting of the two nodes thus enables us to satisfy this aim.



4.3 Outcome of phase 1

Phase 1 successfully allocated over 80% of students to supervisors for the academic year 2017-2018.
However, the system did not allocate 100% why?
If you look at the data its easy to see that the data suffers from a "superstar economy effect".
Consider this scenario: A set of 30 supervisors with 3 places each offering an overall capacity of 90 places. 86 students make their choices over the 30 supervisors. However, 3 supervisors are either so unpopular or not known so they get no picks. As 3 supervisors did not get picked and the software only allocates students to their choices the 9 places offered by those 3 supervisors are effectively non existent. The effective capacity is now 84. As $86 > 84$ a 100%

allocation is now not possible under the current software. In phase 2 we look at means to work around this issue.

5 Phase 2

Phase 2 aims to improve on phase 1. The problem with phase 1 was the superstar economy of supervisors. Some, well known, supervisors get lots of picks, whilst others get few to none. Phase 2 software looks at heuristic measures to try and allocate students to supervisors. The hypothesis is that the software can get around the superstar economy effect by making allocations based on heuristic indicators of common interests between students and supervisors.

5.1 Data collection

To facilitate the heuristic measures of phase 2, more data is collected from supervisors and students.
From students: 6 ranked choices over supervisors, 3 ranked choices over topic areas, A, and a selection of 5 keywords against a fixed set of keywords K.

From supervisors: capacity, topic area and 5 keywords against the same fixed set of keywords, K, are picked.

With these attributes collected we have a feature vector of keywords, topic areas and ranked picks to allocate on.

5.2 Allocation

The allocation in phase 2 is similar to the allocation to phase 1 with a few changes. One way to think of the allocation is to describe the allocation using 2 functions.

The allocation of students to supervisors can be characterised by 2 functions:

- A function that determines the legality of a matching
- A function that determines the cost of a matching

5.2.1 legality

The legality of a matching means to answer whether a possible matching between 2 entities makes sense.

Phase 1 Under phase 1 a matching is legal iff a student chose that supervisor as one of their preferred supervisors.
A matching is legal between supervisor p and student s if

$$p \in C \text{ where C is the set of choices made by s}$$

Phase 2 With the new phase 2 software this is changed to make a match legal even when students do not explicitly pick supervisors.
The legality of a matching will depend on several things, including the number of keywords in common, the topic areas chosen by students as well as the topic area the supervisor belongs to.

For a given student s where:

- C is the set of choices made by the student
- T is the set of topic areas choices made by the student
- K_s is the set of keywords chosen by the student

and a given supervisor p where:

- a is the topic area that p belongs too
- K_p is the set of keywords the supervisor chose

A matching between s and p is legal if:

$$p \in C$$
or
$$n_T \text{ and } a \in T$$
or
$$n_K \text{ and } |K_S \cap K_P| > t$$

where:

- n_T is True if and only if topic area allocation is enabled
- n_K is True if and only if keyword allocation is enabled
- t is an arbitrary user defined constant ≥ 0

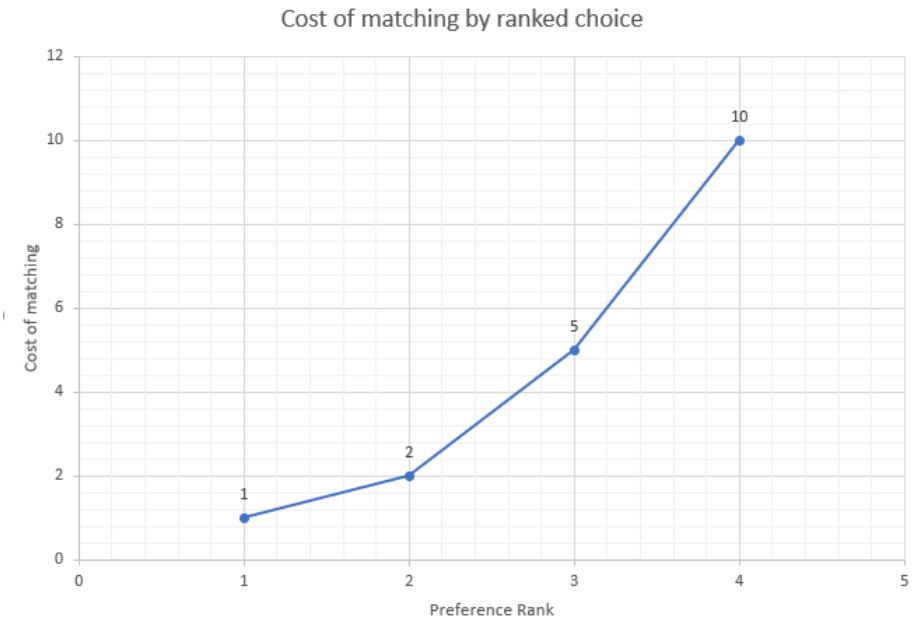
In plain English an allocation between student s and supervisor p is legal if any of the following are true:

- Student s chose p
- Topic area allocation is enabled and the student chose the supervisor's topic area as one of their topic areas choices
- Key word allocation is enabled and the student and supervisor share at least t keywords in common (where t is configurable)

5.2.2 Cost

Each potential matching is associated with some cost $\in N$. These costs allow the algorithm to weigh different potential matches against one another.

Phase 1 Under phase 1, the cost of a matching is described by a function $f(x)$ where x is the rank that the supervisor was given by a student. For example, for a student's first choice x=1, for second choice x=2
Below is a plot of $f(x)$



Phase 2 Under phase 2 the method for evaluating the cost of a matching depends not only on the ranked preference of the student to the supervisor but also the topic area chosen by the student, the topic area of the supervisor and the keywords of both the student and supervisor.
The cost of a matching is the weighted sum of 3 functions instead of 1.

Let $C_{(s,p)}$ be the cost of a matching from student s to supervisor p.

$$C_{(s,p)} = f(x) + n_T g(y) + n_K h(z)$$

where:

- n_T is 1 if and only if topic area allocation is enabled, otherwise 0
- n_K is 1 if and only if keyword allocation is enabled, otherwise 0
- x is the rank that student s gave the given supervisor p
- y is the rank that student s gave the topic area the supervisor p is in
- z is the number of keywords in common between the given student s and supervisor p.

- $g(y)$ is a user defined function similar to $f(x)$ that maps topic area ranks to costs
- $h(z)$ is a user defined function similar to $f(x)$ that maps keywords in common to costs. However typically topic area and preference functions (f and g) are increasing function, as the rank decreases the cost should increase. The keyword function should be a decreasing function. The more keywords in common, the better the matching (in theory), the lower the cost should be.

For example let $f(x) = g(y) = h(z) =$ The identity function i.e $x = f(x)$

- let student s pick supervisor, p as their second choice
- let student s pick topic area "organic" as their second choice
- let supervisor p be in topic area "organic"
- let student s and supervisor p share 2 keywords.
- keyword allocation is enabled
- topic area allocation is enabled

The cost of the matching:

$$C_{(s,p)} = f(2) + 1.g(2) + 1.h(2)$$

note that x=2 as student s picked supervisor p as their second choice, y=2 as student s picked the topic area "organic" as their second choice and supervisor p is in the organic topic area, z=2 as student s and supervisor p share 2 keywords in common.

$$C_{(s,p)} = 2 + 2 + 2 = 6$$

As the functions $f(x), g(y)$ and $h(z)$ are the identity function the cost of the matching is 6
As with phase 1, the algorithm finds a set, M, of the greatest number of student supervisor allocations then minimising the costs of the allocation.

$$Minimise(\sum_{(s,p)\in M} C_{(s,p)})$$

6 Outcome

At the end of phase 2, A complex piece of software has been created than can allocate based on a mix of heuristic and non-heuristic data. The software is flexible and can run under many configurations to find an optimal matching under a given set of rules. Whether the heuristic's chosen are successful in satisfying students is still to be seen as the phase 2 software is to be used for the first time in October 2018.