Tutor-Tutee LP Formulation

Jeremy Wesevich

November 13, 2022

Given a set of n tutors and m tutees, we wish to make an optimal assignment of tutees to tutors. To do this, we create a variable $x_{i,j}$ for each potential pairing, so $x_{i,j} = 1$ if tutee j is assigned to tutor i and 0 if not.

To create an optimal assignment, we assign a "badness weight" $c_{i,j}$ to each pairing $x_{i,j}$ based on the following:

- If tutor i is not qualified to tutor the class that tutee j needs help with, then i and j are considered incompatible and we give their pairing a weight of effectively $+\infty$.
- If tutor i wants to be assigned to tutee j XOR tutee j wants to be assigned to tutor i, then $x_{i,j}$ is probably a good match, so it gets a weight of $c_{i,j} = -1.5n$.
- If tutor i and tutee j want to be assigned to one another, then $x_{i,j}$ is almost certainly a great match, so $x_{i,j}$ gets weight $c_{i,j} = -10n$.
- If tutor i doesn't want to be assigned to tutee j or vice versa, we wish to respect that decision and consider the pairing incompatible, so we give $x_{i,j}$ a weight of (effectively) $+\infty$.

Since we've got some assignments that we consider incompatible, we'd rather give the model other options so it doesn't just spit out garbage if it needs to make an incompatible pairing. To give it a way out, we create a "ghost tutor", a tutor which doesn't actually exist, but can take on tutees that would otherwise be given incompatible pairings. Thus, if a tutee is assigned to the ghost tutor, this is equivalent to them not being paired with anyone.

To model the ghost tutor, we create a binary variable g_j for each tutee j, and give it a "badness weight" of 50,000. This ensures that the ghost tutor won't be picked in place of any compatible pairing (assuming some reasonable size constraints on the problem), but will always be picked over any incompatible pairing.

We also wish to incentivize even distribution of tutees among tutors, that is, we don't want to give all the tutees to one tutor (unless they all really prefer that tutor). To model this, we define variables $d_{i,j}$ to be the absolute difference between the number of tutees assigned to tutor i and the number of tutees assigned to tutor j. If the sum of the $d_{i,j}$

values is low, then tutees are relatively evenly distributed, while if the sum is high, then the assignment is imbalanced.

Given all this, we can now write an objective function for the Gurobi solver to minimize:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} c_{i,j} x_{i,j} + 50000 \sum_{j=1}^{m} g_j + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{i,j}$$

To properly model the problem, we must also include some constraints. First, each tutee must be assigned to exactly 1 tutor (even if it's the ghost tutor):

$$g_j + \sum_{i=1}^{n} x_{i,j} = 1, \quad j \in \{1, ..., m\}$$

Second, each tutor must be assigned no more than t_i tutees, where t_i is the maximum number of tutees that tutor i can fit in their schedule:

$$\sum_{j=1}^{m} x_{i,j} \leq t_i, \quad i \in \{1, ..., n\}$$

Note that the ghost tutor has no such restriction.

Finally, we must force $d_{i,j}$ to actually take the value it's intended to have, namely the absolute value of the difference in number of tutees assigned to tutors i and j. This can be accomplished with the following two inequalities, noting that we generally wish to minimize the value of each $d_{i,j}$ based on the objective function:

$$\begin{aligned} \sum_{k=1}^{m} x_{i,k} - x_{j,k} &\leq d_{i,j} && \forall i, j : 1 \leq i < j \leq n \\ \sum_{k=1}^{m} x_{j,k} - x_{i,k} &\leq d_{i,j} && \forall i, j : 1 \leq i < j \leq n \end{aligned}$$

Finally, all $x_{i,j}$ and g_j variables must be binary, while all $d_{i,j}$ variables must be integer-valued. The $c_{i,j}$ values are constants calculated ahead of time, and the t_i's are just data given to us. This completes the description of the model.