Optimizing the CS Match with Integer Linear Programming

Hugh Shanno, Batmend Batsaikhan, Mary Blanchard, Cecilia Ehrlichman, Andreas Miller, Petrichor Park

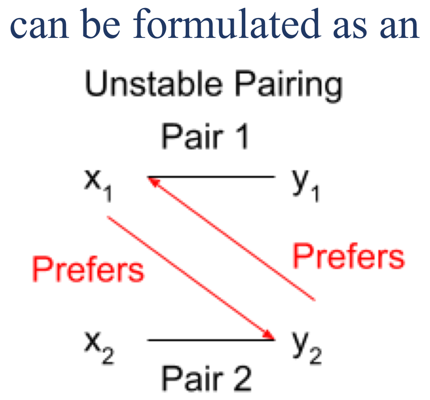
Carleton College Department of Computer Science, Northfield, Minnesota

What is Integer Linear Programming?

- Integer Linear Programming (ILP) is a subclass of problems in NP-space (a class of problems that cannot be solved efficiently).
- ILP problems are formulated as a series of linear inequalities, known as constraints, where all of the variables must be integers.
- ILPs also have an objective function that attempts to maximize or minimize a certain linear equation.
- ILPs are known to be solvable much faster by a computer than other NP problems.
- Formulating NP problems as ILPs should result in much quicker solutions to those problems.

Stable Matching with ILPs

- One of the main goals of this project is to build a system to solve the CS Match, a stable matching, using ILPs.
- The hospital resident matching problem allows for students and courses to have ties in their preference lists, while the Match does not, but is NP-Hard, so can be formulated as an ILP.
- A matching between students and courses is "stable" if it contains no unstable pairings.



To ensure that each student is matched to at most one class, we create the constraint

$$\sum_{j=1}^{c} x_{ij} \leq 1, 1 \leq i \leq s$$

Where x_{ij} represents whether student i is matched to course j. We can enforce course capacities by stating

$$\sum_{i=1}^{s} x_{ij} \leq c_j, 1 \leq j \leq c$$

Where c_j is the capacity of course j. We also want to ensure that if a student is not matched to a particular course, or any course it ranks at least as high, then that course should be full with students ranked at least as high, which can be formulated as

$$c_j\left(1 - \sum_{q} x_{iq}\right) \leq \sum_{p} x_{pj}, 1 \leq i \leq s, 1 \leq j \leq c$$

Where q represents courses that student i ranks at least as high as course j, and p represents students that course j ranks at least a high as student i. To maximize the number of students matched to courses:

$$max \sum_{(i,j)} x_{ij}, \quad 1 \leq i \leq s, 1 \leq j \leq c$$

Thus, we can formulate the CS match as an ILP allowing for ties in preference lists.

Optimizations of ILPs

Stable-Matching Specific:

- If certain pairings are not a part of any stable matching, then their values will always be 0.
- Thus, the variables corresponding to these pairings can be removed from a formulation, resulting in fewer variables and constraints to process.
- In the case of the CS match, students cannot be matched to courses they rank below the "no course" course, so these variables can all be removed from the formulation.

Branch and Bound Method:

- Runs the LP relaxation of an ILP (using the constraints of the ILP but allows variables to take real values).
- If a variable x is between two integers, the method runs the LP relaxation again, but with the added constraint that either $x \leq \lfloor x \rfloor$ or $x \geq \lceil x \rceil$ as seen in figure 1.
- The process is then continued until all variables are integers, and the optimal solution is chosen.

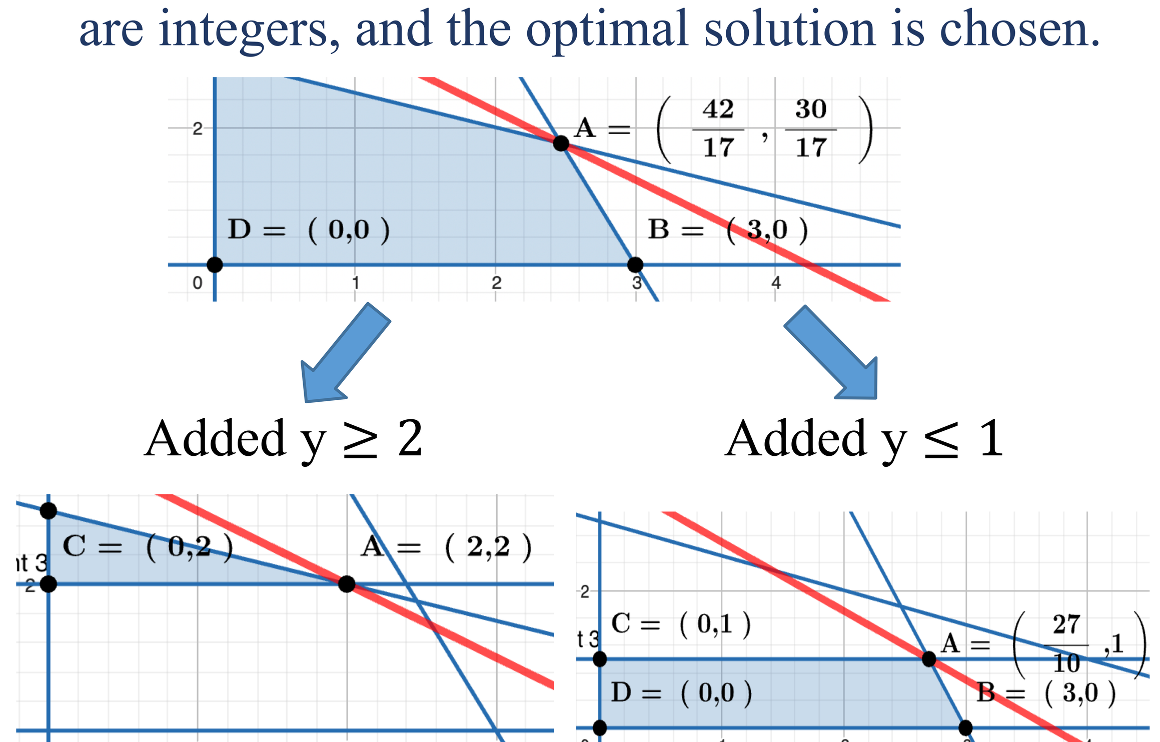


Figure 1

Branch-Cut Method:

- Runs the ILP solver on a subset of the constraints in the formulation.
- By removing constraints, the ILP will be able to run faster.
- Then we check if the held-out constraints were violated by the solution, and add any violated constraints to the solver.
- Repeat these steps until a valid solution has been reached.

System Architecture

Final system for solving the CS match using ILPs:

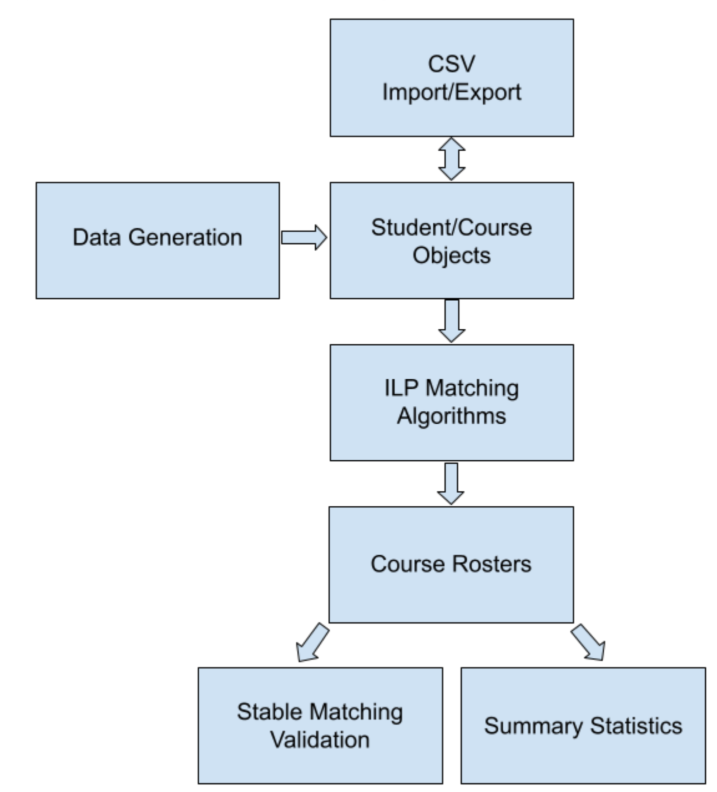


Figure 2

Results

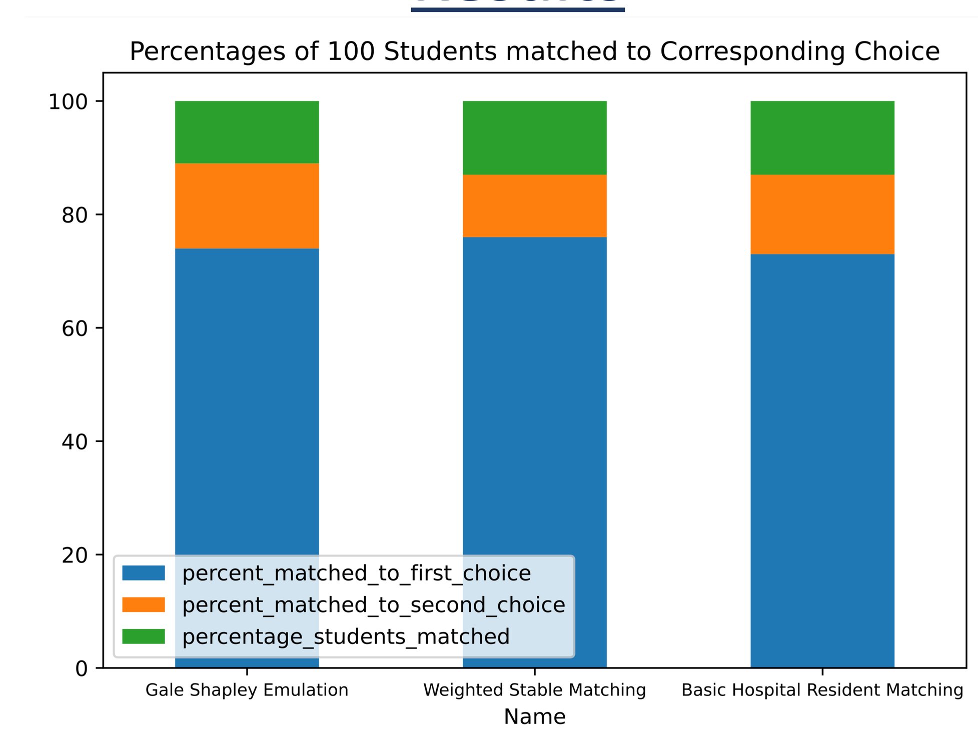


Figure 3

- The different stable matching algorithms all perform similarly (figure 3).
- All matchings performed by both the hospital resident matching problem and its version with reduced variables are stable and fast (figure 4).
- The two variations give roughly the same matching results (figure 4).

Number of students	Is Matching Stable?	Average Time to Match Students (seconds)	Distributions of Matched Students
100 Students	Stable	0.136	Equal
100 Students RV	Stable	0.038	
150 Students	Stable	0.336	Not Equal
150 Students RV	Stable	0.136	
200 Students	Stable	0.545	Equal
200 Students RV	Stable	0.173	
250 Students	Stable	0.835	Equal
250 Students RV	Stable	0.239	
300 Students	Stable	1.392	Equal
300 Students RV	Stable	0.400	

Figure 4
(RV = Reduced Variables)

Conclusions

- The ILP matching algorithm performed similarly to the CS match, with similar numbers of students getting matched to courses and similar number of students getting matched to their top two classes.
- In general, the ILP for the hospital resident matching problem is blisteringly quick, resulting in near-instant course matchings for students.
- The amount of time the solver takes per student grows at a low exponential rate.
- All algorithms consistently produced stable matchings.
- Simply removing the impossible matchings has a huge effect on the speed of the ILP solver, roughly thirding the runtime at lower student counts, and slightly increasing the ratio at higher student counts.
- ILPs are much faster methods of solving NP-Hard problems, and with optimizations can be made significantly faster.

References

Blanchard, J. D., & Chamberland, M; (2022). Salvaging college registrations during COVID-19 via integer programming. *Mathematics Magazine, preprint.* https://blanchard.math.grinnell.edu/home/Research/SRIP_preprint.pdf

Delorme, M., García, S., Gondzio, J., Kalcsics, J,. Manlove, D., & Pettersson, W. (2019). Mathematical models for stable matching problems with ties and incomplete lists, *European Journal of Operational Research*, *277*(2), 426-441. https://doi.org/10.1016/j.ejor.2019.03.017.

Gale, D. & Shapley, L. S.; (1962). College admissions and the stability of marriage. *The American Mathematical Monthly, 69*(1), 9-15. http://www.jstor.org/stable/2312726?origin=JSTOR-pdf

Gusfield, D. (2019). *Integer Linear Programming in Computational and Systems Biology: An Entry-Level Text and Course*. Cambridge: Cambridge University Press. doi:10.1017/9781108377737

Gusfield, D. & Irving, R. W. (1989). *The stable marriage problem : structure and algorithms*. MIT Press.

Pini, M. S., Rossi, F., Venable, K. B., & Walsh, T. (2011). Weights in stable marriage problems increase manipulation opportunities. *TARK XIII*. *Association for Computing Machinery*. 200–204. https://doi.org/10.1145/2000378.2000402

Slama, T. (2023, July 4). The Art of Linear Programming. YouTube. https://www.youtube.com/watch?v=E72DWgKP_1Y

Winston, W. L. (1996). Operations research: Applications and Algorithms. Brooks/Cole.

Acknowledgements

Layla Oesper, for advising us throughout this entire project.

Dave Musicant, for helping us to understand how the CS Match works and giving us its summary statistics.

My team for their hard work this term.