Diana Gonzalez Santillan
dgonza27 1350682

<u>Optimization and the “Time = Money” mentality</u>

In this essay, I discuss the connection of the Stable Matching problem to two of its algorithmic solutions, one by David Gale and Lloyd Shapley (G&S) ("College Admissions and the Stability of Marriage", 2013), and another one by Robert Irving, Paul Leather, and Dan Gusfield (IL&G) (“An Efficient Algorithm for the ‘Optimal’ Stable Marriage”, 1987). In particular, I discuss how algorithm optimization can be defined in different ways depending on the algorithm designer’s underlying priorities. Furthermore, I claim that the perceived preference given to optimality as speed, as opposed to optimality as fairness, is due to the capitalist mentality of “time = money” in which most computer scientists develop. Throughout the essay, I draw on ideas from Rasheedah Phillips’ article “Dismantling the Master(s) Clock(work Universe)” (2016) in order to critique the practice of partitioning time into quantifiable, linear chunks that provide a platform for time-monetization. Finally, I will analyze the lack of diversity in the Computer Science field, and how this pulls algorithm design further away from objectivity and fairness.

Now, to talk about computer algorithms we must first talk about the science that produces them. Computer Science is defined to be the study of algorithms for logical problem solving (Wikipedia). Moreover, computer scientists are not simply concerned with the design of *any* algorithm. Instead, they dedicate their time to finding the *optimal* algorithmic solution to any given logical problem. However, to aim towards optimality we must first agree on what optimality means. On the one hand, it is common in Computer Science to find optimality as a synonym for speed, and optimization as the action of making an algorithm run as fast as possible. On the other hand, optimality could also be taken as a synonym for fairness, in which case optimization would mean to design an algorithm such that its results benefit all of the involved parties as equally as possible.

In the case of the Stable Matching problem, G&S first came up with an $O(n^2)$ [1] solution in 1962. This solution was thoroughly unfair because it gave the best possible results to one side (typically the men), and the worst possible results to the other side (typically the women) (G&S, p. 390). Some years later, in 1976 Donald Knuth inquired if there existed a more "egalitarian" solution to the problem (IL&G, p. 532). However it wasn't until 1987 that IL&G actually set themselves to find "a stable matching that is optimal under some more equitable or egalitarian criterion of optimality" (IL&G, p. 532). In their paper, IL&G describe an $O(n^4)$ solution that benefits both "men and women" equally at *the cost* of being slower than G&S's algorithm (IL&G, p. 542). In other words, IL&G prioritized fairness over speed when it came to defining optimality. Despite this, it seems that nowadays G&S's unfair-but-fast algorithm is preferred for everyday-life applications. Lloyd Shapley and the economist Alvin Roth were even awarded the Economics Nobel Prize in 2012 (Nobel Media, 2014, p. 1) due to their work in applying the "unfair" algorithm to everyday-life situations such as kidney transplants, and assigning hospitals to medical residents.

If we are speaking about kidney transplants, getting a speedy matching could very well be a matter of life or death. However, when it comes to other practical

[1] In Computer Science, the *O* notation indicates how fast the running time of an algorithm increases as the input size goes to infinity. Both $O(n^2)$ and $O(n^4)$ say that the running time increases exponentially, which is never good. However n^4 increases way faster than n^2, and therefore $O(n^4)$ is a worse running time.

applications, choosing to focus on optimality-as-speed instead of considering optimality-as-fairness could also be linked to the capitalistic idea of “time = money”. This idea is rooted in the perception of people as simple agents of production in a capitalist economy (cite). That is, the more time a person spends producing something, the more money *someone* gets from selling that something. Therefore, people’s time becomes directly proportional to their productivity, and hence becomes subject to monetization (i.e., “time = money”). As a matter of fact, this monetization is not new, as it has been around in the Americas ever since the arrival of the first african slaves. Walter Johnson points out that “one of the many things slaveholders thought they owned was their slaves' time” (quoted from Rasheedah Phillips’ “Dismantling the Master(s) Clock(work Universe)” p. 22). Thus, when making transactions concerning their slaves, slaveholders were not only selling a slave, (or a person, or a product,) they were also selling the production time this person (or slave, or product) could provide.

In order to monetize time, slaveholders did not only have imagine that they owned their slaves’ production time. They also had to split that time into linear, quantifiable, and thus sellable, chunks. Paraphrasing Mark M. Smith, Phillips writes that "white southern slave masters adapted a mechanical clock time and corresponding linear time construct [to impose over] nature-based timekeeping methods. [This] impacted the social order and reinforced values of discipline, economic gain, efficiency, and modernity" (p.19). Coincidentally, the values mentioned by Smith are still highly regarded in today’s capitalistic society. Furthermore, the act of associating “timekeeping” and “linear time” with “efficiency” and “economic gain” still drives the way in which computer scientists design algorithms today.

Computer scientists feel the pressure to produce the fastest algorithms for today’s fast paced way of life, and use time efficiency as the main criteria for optimality because our society monetizes time and values economic gain. Nevertheless, algorithm design is not only influenced by ideas of time monetization. The background assumptions and upbringing of each computer scientist is also reflected on the way in which they go about solving a problem, the criteria they choose for testing optimality, and even the metaphors they choose to illustrate the solution to a given problem. G&S most likely did not realize (or did not choose to care) that their algorithm and the marriage metaphor they used as illustration was unfair to women directly, and to gender non-conforming and homosexual people by erasure. After all both David Gale and Lloyd Shapley were white, straight[2], men born in the 1920’s, as probably were the majority of the colleagues who checked their work. Thus, no one was there to point out that the illustration chosen, as well as the algorithm itself, might result quite problematic.

During several lectures and sections this quarter, we as a class discussed the unattainability of objectivity. After reading articles by Helen Longino and Donna Haraway, among other authors, we concluded that due to every person’s background assumptions and expectations, it is impossible to ever be objective as an individual. Instead, it was suggested that the best way to asymptotically approach objectivity is through collaboration with people coming from diverse backgrounds. In this way, every

---

[2] I am assuming their sexuality from Gale’s biography (O’Connor, 2008), and Shapley’s obituary (Wiel, 2016), both of which mention that they married to women and had children with them.

member of a team would be able to point out and question other members' assumptions and omissions (cite).

One may think that Computer Science is exempt from this definition of objectivity. After all, things related to computers are most of the time binaries, and it is easy to tell whether a binary is what we expect it to be or not. The transistor is on or off, the boolean is true or false, the algorithm works or it doesn't, we do not need people from different backgrounds to answer these questions. However, I claim that objectivity can also be considered as how fair an algorithm (and its illustration) can be. Even though algorithms work or don't work, the discourse with which algorithms are presented also matters. In this sense, having a more diverse working team would lead to algorithmic solutions that are more fair to all the parties, and to illustrations that do not erase certain groups of people. Unfortunately, nowadays Computer Science is still not very diverse (cite). This lack of diversity can be mapped to many things, most of which are beyond the scope of this essay. Nevertheless, one of them is the same "time = money" idea that inclines algorithms to be unfair in the first place.

Due to time monetization, universities oftentimes enroll more students than they have capacity for, and make curriculums packed with too much complex information in order to get more student money and have them off graduating in less time (cite). This results in a very difficult learning environment for everyone, but it mostly affects people from underrepresented communities who might have not had the best high school education, or who might feel extra pressure from being the "only woman" or the "only black person" in the room (cite). These factors lead to a higher Computer Science dropout rate for people of color, LGBTQ people, and women (cite), which in turn causes a lack of diversity in the Computer Science field, which leads to less objectivity and fairness in algorithm design.

Overall, in this essay I was able to point out that the values of capitalism as well as the capitalist idea of “time = money” have been present in the Americas ever since the arrival of slavery. Moreover, I discussed how time monetization contributes to the lack of diversity in Computer Science, as well as to the priorities computer scientists take when deciding whether to optimize an algorithm for time efficiency as opposed to fairness. Furthermore, I have discussed how a diverse working team does not only take us closer to objectivity in the STEM fields, but also takes us closer to fairness in a science that otherwise seems to always be objective (Computer Science). Therefore, perhaps looking at time in an alternative way will help attain more diversity in the field and more fairness when finding algorithmic solutions to logical problems.

<u>Bibliography</u>:

Gale, David, and Lloyd Stowell **Shapley**. "College Admissions and the Stability of Marriage." *The American Mathematical Monthly* 120.5 (2013): 386-91. Web.

Irving, Robert W., Paul **Leather**, and Dan **Gusfield**. “An Efficient Algorithm for the ‘Optimal’ Stable Marriage.” *Journal of the ACM* (1987): 532-43. Web.

"The Prize in Economic Sciences 2012 - Popular Information". *Nobelprize.org.* **Nobel** Media AB 2014. Web. 24 Apr 2017.- Click on “Public Information” pdf.

Novak, Rebecca. “How Diversity in Computer Programming Will Benefit everyone”. The Huffington Post. Web. 19 Nov. 2013. Accessed: 13 June, 2017.

Phillips, Rasheedah. “Dismantling the Master(s) Clock(work Universe).” *Space-Time Collapse I: From the Congo to the Carolinas*. (2016): 15-31. In class pdf.

O’Connor, John J. “David Gale Biography.” *St. Andrew’s School Biographies*. (2008). Web: http://www-history.mcs.st-andrews.ac.uk/Biographies/Gale.html

Weil, Martin. “Lloyd Shapley, who shared the 2012 Nobel Prize in economics, dies at 92.” *The Washington Post*. Web. 14 Mar. 2016. Accessed. 12 Jun. 2017.

Wikipedia. “Computer Science.” Wikipedia: The free encyclopedia. Last edited: June 12, 2017. Accessed: June 13, 2017.