**CE301 Challenge Week Background Reading**

The first reading relevant to my final year project was concerning the stable marriage problem / stable matching problem. The stable marriage problem tries to see if there is a general algorithm that matches men and women to each other based on their preferences in a way that ensures they are stable [1]. A matching is stable when there is not another match which both prefer each other to their current partner.

From this, I researched the classic algorithm used to solve the stable marriage problem, the Gale–Shapley algorithm. This algorithm proves, for an equal number of men and women, it is always able to solve the stable marriage problem, i.e., make all matches stable. The results from the algorithm are best for the ‘proposers’ and is ‘truthful’ from the point of view of the proposers (they cannot misrepresent their preferences in order to get a better matching). This algorithm was put forward by David Gale and Lloyd Shapley [2].

Similar to the stable marriage problem, is the stable-roommate problem. In contrast to the stable marriage problem, there are no distinct groups (e.g. men and women), anyone can be matched with anyone. Another difference with the stable marriage problem is the fact that, for some sets of people and preferences, there could be no stable match. An algorithm to check whether a stable matching exists, and if so, find the matching, was given by Robert Irving [3].

The next step of research dived into rental harmony. Rental harmony is a fair division problem where items and a fixed monetary cost must be divided together [4]. Crucially, there are several properties that any assignment should satisfy. Firstly, all prices should be 0 or more. In a typical rental house setting (where rooms are divided amongst people in a group), it is hard to image accepting a result where someone would be payed to get a room. Secondly, the result should be Envy-Free (EF). Put simply, EF means that all the participants prefer their allocated room at the given price. This idea was introduced by George Gamow and Marvin Stern in their book Puzzle-math. Lastly, we want the results to satisfy Pareto efficiency. This is an allocation where no one can be better off without making at least someone worse off.

I researched many algorithms that provide rental harmony. The first algorithm I researched was devised by Francis Su. I read about his algorithm in a New York Times article, that provided a brief summary of Francis Su’s work [5]. The algorithm asks people what room they prefer in a certain pricing scheme. The algorithm returns a sequence of allocations that meet at an envy-free allocation. The outcome is non-negative and envy-free. However, the algorithm has many assumptions. One is that people always prefer a free room, which could be problematic. Another problem is that is assumes each room has a capacity of 1. Additional critique is presented in Procaccia, Velez and Yu’s paper [13], where they concluded that the algorithm’s preference selection is cumbersome, as multiple interactions with the user is required, the algorithm makes it impossible to select between envy free allocations, and it is impossible to adapt the algorithm to a budget constraint case.

The next algorithm I researched expanded Su’s solution to include a situation where the capacity of each room may be larger than 1. This was put forward by Azriely and Shmaya [6]. Unlike past research, they create a model where an agent’s preferred room may be a function of the entire vector of prices. This means that a person’s preference can be dependent on the prices of other rooms. They prove envy-freeness using Shapley’s KKMS theorem and Hall’s marriage lemma. This algorithm also has similar assumptions as Su’s algorithm, including the assumption that people prefer free rooms over any other room.

The next set of algorithms I researched are all cardinal versions. In these algorithms, every member of the group must submit a bid to each room. The first aspect I needed to research is maxsum / utilitarian allocation. This is the result that minimizes the sum of bids. Finding this is know as the assignment problem. The algorithm developed by Harold Kuhn solves the assignment problem [7].

The first cardinal algorithm I researched came from Brams and Kilgour [8]. They suggest a procedure called the Gap Procedure. This ensures non-negative prices, Pareto efficiency, monotonicity, where obtaining a good is never harmed by higher bids, sincere bids, and partially independent (on the amounts bid) prices. The Gap Procedure first calculates a maxsum allocation (see above), then continues based on the maxsum results. If the maxsum is less than the total cost, it is an unsolvable problem (partners don’t want to pay total amount required). If it equals the total cost, rooms are allocated. If the maxsum is more than the total cost, the room prices are lowered based on the gap between those prices and the next lowest valuations. If there is a room that is more desirable, it will cost more. One crucial aspect of this procedure is that it is not envy-free. This algorithm is more marketlike, as competitive rooms are more expensive, at the cost of envy.

The second cardinal algorithm I research was presented by Haake, Raith, and Su [9]. This algorithm has a unique property called the Compensation Procedure. One requirement in this algorithm is that the sum of a person’s bids must be at least the total cost. As the previous algorithm, it first finds a maxsum allocation. The algorithm then differs from the previous one. A pool of money is formed by charging the participants the value of their allocations. Crucially, if there is money left over after paying the (house) cost, then the algorithm eliminates envy by compensating envious participants. The sum of compensations is the smallest sum required to eliminate envy, with any money remaining being divided in an envy-free way, e.g., equal amount to each person. Also, unlike the previous algorithm, this procedure is not non-negative, but envy-free, and Pareto-efficient. As the authors warn, negative payments can lead to problems and should be avoided.

Another algorithm that was researched was suggested by Abdulkadiroglu and Sonmez and Unver [10]. This approach differs from the last two as it is a strong market-based approach. It simply uses an auction to determine who gets what. All rooms start with the same price. The participant choses rooms they want at the given price. Rooms that are over-demanded have their price increased at a constant rate, while simultaneously, the prices of the other rooms decrease such that the total cost of the rooms equals the total cost of the house. Keep doing this until there are no over-demanded rooms. Hall’s marriage theorem is then used to allocate a participant their room. The result is always envy-free (and thus Pareto-efficient). The results however are also not non-negative, however, its is non-negative if there is an envy-free allocation that’s non-negative, unlike the previous procedures.

The next algorithm was devised by Sung and Vlach [11]. First, as with most of the previous algorithms, a maxsum allocation is found. Then, a minsum price-vector is found (that is envy-free). This vector is one where the sum of the prices is minimized. Based on the minsum vector, certain actions are performed. If the minsum equals the total cost, implement the maxsum allocation with the minsum prices. If it is less then the total cost, increase prices at a constant rate (until the sum reaches the total cost). If the minsum is over the total cost, then there is no solution that is non-negative and envy-free. When this happens all the prices could be decreased at a constant rate, which will result in some prices being negative, or only the positive prices could be decreased at a constant rate, which will result in envy, though the envious will get their room for free. This algorithm captures the desirable properties of all the previous algorithms and allows for allocation to be non-negative but not envy-free, or envy-free by with negative prices, allowing users to chose what properties they prefer.

Mash, Gal, Procaccia and Zick use data from the Spliddit website to build an algorithm that is envy-free and preferred by users [12]. The algorithm devised uses a maximin approach, which maximizes the minimum utility of a participant subject to envy-freeness. Their paper demonstrates that using their maximin algorithm, participants have a higher satisfaction rating than by just using an envy-free allocation. However, in this algorithm, results can be negative. Unlike other algorithms, this one considers satisfaction of actual users, and is thus much more relevant to real world application of rental harmony.

Not addressed by any of the previous algorithms is budget constraints. Procaccia, Velez and Yu [13] present an algorithm that tries to find an envy-free allocation that also considers budget constraints. They algorithm is a maximin solution similar to the Gal et al. algorithm. The paper shows that a budget constraint version of the Gal algorithm also produces desirable fairness features. In practice, this idea means that users would also need to be asked their budget constraints.

All the rental harmony algorithms I have researched so far are not stratergyproof, someone can gain by giving false evaluations. More generally, strategyproofness is incompatible with envy-freeness. This is addressed in different ways. The paper by Gal et al. [12] states that strategic behavior does not play a significant role in practice, due mainly to users not being aware of how the algorithm works. Procaccia et al. [13] agree with this and also go further to argue that even with manipulation, an envy-free allocation can still be obtained.

**References**

[1] W. Hunt, “The Stable Marriage Problem”, Lecture [Online], Dept. CompSci. and Elect. Eng., West Virginia Univ., Morgantown. Available: <http://community.wvu.edu/~krsubramani/courses/fa01/random/lecnotes/lecture5.pdf>

[2] D. Gale, L. Shapley, “College Admissions and the Stability of Marriage”, *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9-15, Jan 1962.

[3] R. W. Irving, “An Efficient Algorithm for the ‘Stable Roommates’ Problem”, Journal of Algorithms, vol. 6, no. 4, pp. 577-595, Dec 1985.

[4] D. Lesage. (2019). *The Rental Harmony Problem: How to Split Rent (and Everything Else) Fairly* [Online]. Available: <https://medium.com/betterism/the-rental-harmony-problem-how-to-split-rent-and-everything-else-fairly-3a7804a6c72f>.

[5] A. Sun. (2014). *To Divide the Rent, Start With a Triangle* [Online]. Available: <https://www.nytimes.com/2014/04/29/science/to-divide-the-rent-start-with-a-triangle.html?_r=0>

[6] Y. Azrieli, E. Shmaya, “Rental harmony with roommates”, Journal of Economic Theory, vol. 153, pp. 128-137, Sep 2014.

[7] Brilliant.org. (Retrieved 2020). Hungarian Maximum Matching Algorithm [Online]. Available: <https://brilliant.org/wiki/hungarian-matching/>

[8] S. J. Brams, D. M. Kilgour, “Competitive Fair Division”, Journal of Political Economy, vol. 109, no. 2, pp. 418, Apr 2001.

[9] C. Haake, M. G. Raith, F. E. Su, “Bidding for envy-freeness: A procedural approach to n-player fair-division problems”, Social Choice and Welfare, vol. 19, pp. 723-749, Oct 2002.

[10] A. Abdulkadiroğlu, T. Sönmez, M. U. Ünver, “Room assignment-rent division: A market approach”, Social Choice and Welfare, vol. 22, pp. 515-538, June 2004.

[11] S. C. Sung, M. Vlach, “Competitive envy-free division”, Social Choice and Welfare, vol. 23, pp. 103-111, Aug 2004.

[12] Y. K. Gal, M. Mash, A. D. Procaccia, Y. Zick, “Which Is the Fairest (Rent Division) of Them All?”, in *Proceedings of the ACM Conf. on Economics and Computation*, Maastricht, The Netherlands, 2016, pp. 67-84.

[13] A. Procaccia, R. Velez, D. Yu, “Fair Rent Division on a Budget”, in AAAI Conf. on Artificial Intelligence, New Orleans, Louisiana, USA, 2018, pp. 1177-1184.