Chapter 1

Background

1.1 Mechanism Design

Traditionally, economists and game theorists most deeply engage with the set of problems involving the construction of systems that "fairly" (under various definitions of the word) address scenarios involving a host of participants with adverse or overlapping goals. Some examples of problems that fall under this classification are as follows: designing a voting system that successfully eliminates strategizing by any party [**voting**], auctioning search engine advertisement spots in a way that promotes honest bidding [**auctions**], assigning roommates from a pool of students such that each student ends up with a roommate they prefer [**irving**], and maximizing aid while allocating food donations to food banks [**foodbank**].

It turns out that many mechanism design problems can be solved algorithmically, and as such computer scientists have increasingly joined the economists in the study of mechanism design theory. There are two subcategories of mechanism design relevant to this paper: stable matching and fair division.

1.1.1 Stable Matching

Matching problems are highly applicable to many real-world scenarios, such as assigning medical students to hospitals for residency [**nrmp**] or the aforementioned roommates example. The canonical (and heteronormative) matching problem is known as the Stable Marriage Problem [**princeton-marriage**], in which the goal is to match an equal number of men and women such that no participant remains uncoupled. This is known as a bipartite matching problem, or one in which the participants are divided into disjoint sides and matched from one side to the other. The other important aspect of the Stable Marriage Problem is that the matching should prevent "cheating," which occurs when a man and a woman who weren't matched to each other prefer to be together over their assigned partners and will "elope," leaving the other two participants alone. This couple would be known as a blocking pair, because their elopement "blocks" the matching from being complete. Herein lies the stability aspect of stable matching: we wish to find an algorithm that outputs a stable matching of couples, with no blocking pairs.

In 1962, David Gale and Lloyd Shapley published a seminal paper for the field of stable matching in which they proposed the deferred acceptance (DA) algorithm and proved it to output a stable matching [**gale˙shapley**]. Since then, the algorithm has been adapted to fit many more scenarios with various types of considerations, including the residency problem. However, there remain subsets of problems that do not fall under the same paradigm as the DA algorithm: for instance, problems with a one-sided market. The stable roommate problem is an example of one such problem, in which participants are to be matched from the same pool rather than two distinct categories of participant and stability retains the same meaning as in the bipartite setting. Over 20 years after Gale and Shapley's paper, in 1985 Robert Irving proved stability and efficiency for an algorithm that solves the roommate problem [**irving**]. Irving's algorithm also now serves as a jumping-off point for the majority of further research on one-sided matching problems.

1.1.2 Fair Division

The difference between matching and fair division often lies in the primary metric used to evaluate the effectiveness of the solution. Matching is most predominately studied under stability, and fair division under analysis of envy. When dividing a chocolate-vanilla cake, a person who prefers vanilla but receives chocolate will be envious of those who receive a vanilla slice. In a fair division problem, the goal is to minimize the amount of envy. The cake example is the canon example for divisible goods, but there is also a lot of study into the division of indivisible goods, which introduces a new set of considerations for maximizing welfare (the benefit participants yield from the goods they receive) and minimizing envy. The food bank example is one in which the goal pertains to fairly allocating indivisible goods [**foodbank**].

Formally, envy fair division will often be evaluated under pareto-optimality. An allocation is considered pareto-optimal if there is no other allocation that is strictly better for at least one participant while retaining the same level of welfare for the other participants [**pareto**]. As more factors are added to these problems— such as considering the order in which goods arrive for distribution, scaling the number of participants, and more specific definitions of welfare—envy freeness and pareto-optimality become more difficult to ascertain.

1.2 Adding constraints

What's known as the "vanilla" version of these matching and division environments, which is to say the more simple or basic versions, have been heavily studied over years of computer science and economics literature. So, the problems are often made more difficult or made to more accurately reflect real-world problems by adding *constraints*, therefore introducing new facets of the classic problem to solve. Constraints can come in many forms, and can often make problems too difficult to solve efficiently or impossible to solve at all. In the scope of class assignment and related literature, constraints often come in the forms of quotas on class capacity and categories of

students.

1.2.1 Quotas and Diversity

Quotas in allocation and matching problems can be implemented as upper and lower bounds. Upper bound quotas implies a capacity constraint, such as a maximum number of open positions per hospital in the residency matching problem. The College Admissions problem, in which a college with n applicants can admit up to q students, is traditionally studied with these capacity constraints (where q is for quota). In fact, the College Admissions problem was introduced with Gale and Shapley's Stable Marriage DA algorithm [**gale˙shapley**]. In recent years, these quotas have been extended in a variety of ways, such as introducing lower bounds in which schools also need to accept at least m students of some type [**quotas**]. Because adding quotas can make the problem too difficult, quotas are often studied under "soft" constraints as well: for example allowing a quota to be a target range rather than a specific number [**santhini**].

There is also

1.2.2 Big-O and Complexity

1.2.3 Optimization

1.3 Related Works

1.3.1 Two-sided matching

1.3.2 One-sided matching