Incentives in CS: Exercise Set #1

Due by 11:59 PM on Thursday, January 16, 2020

Instructions:

(1) You can work individually or in a pair. If you work in a pair, the two of you should submit a single write-up.

(2) Submit on Canvas. Only one person needs to submit the assignment. When submitting, please be sure to add your partner's name (if any). Be sure to add your partner to your group on Canvas.

(3) Please type your solutions. We encourage you to use the LaTeX template provided on the course home page.

(4) Write convincingly and clearly but not excessively. You should be able to fit all of your solutions into two pages.

(5) Except where otherwise noted, you may refer to anything accessible from the course Web page *only*.

(6) You can discuss the exercises verbally at a high level with other groups. And of course, you are encouraged to contact the course staff (via Piazza or office hours) for additional help.

(7) If you discuss solution approaches with anyone outside of your group, you must list their names on the front page of your write-up.

(8) You have a total of 5 late days throughout the quarter.

Exercise 1 (15 points)

Recall the allocation of offices to grad students that we studied in lecture 1. Consider the serial dictatorship mechanism (which does not require that each agent start off with an office).

Serial Dictatorship Mechanism

1. Each student submits a ranked list of all offices.
2. The students are ordered in some way. (E.g., randomly).
3. The students are considered in order. When student i is considered, she receives her top-ranked option that is still available.

Consider the following definition:

Definition 1 An allocation procedure is *group-strategyproof* if, even when groups of students are allowed to collude, honesty is the best policy.

Important: This is different from the notion of stability that we discussed in class for TTCA. Here the agents are still participating in the original mechanism, but are colluding on their reports in the first step.

(a) **(6 points)** Prove that a serial dictatorship is group strategyproof. Formally, we need to prove that: for every subset S of students, no matter how students are ordered in the second step, if some coordinated misreport of the subset's ranked lists makes at least one student in S strictly better off, then this misreport also makes at least one student in S strictly worse off.

(b) **(4 points)** Consider the even more general statement that *any* mechanism that is Pareto-optimal and truthful is group-strategyproof. This statement is false. Below is an incorrect proof.

> **Proof:** Let T be the truthful run of the mechanism and O_T denote the outcome of the truthful run. Now suppose there exists an arbitrary subset of lying students S, and let O_L denote the outcome when those students lie (the lies are also arbitrary). Suppose, for the sake of contradiction, that some student in S strictly prefers O_L to O_T (according to their *true* preferences). By the Pareto-optimality of O_T, in order to give someone a strictly better outcome, someone else must receive a strictly worse outcome. Now, since telling the truth is a dominant strategy, it can't be the case that someone who told the truth got a worse outcome in O_L than in O_T. Thus, it must be someone who lied, i.e. some $s \in S$, who got a worse outcome in O_L than in O_T.

Find and explain the error in this proof. *Hint:* the error is confined to a single sentence.

(c) **(5 points)** Now suppose that indeed each student starts with some office and then we run the serial dictatorship mechanism. Show that serial dictatorship is **not** stable. That is, show that it could benefit some subset of agents to refuse to participate in the mechanism, but rather go off and trade offices on their own.

To show this, it suffices to specify an order (for step 2) and provide an example of a set of preferences for which some subgroup of students is incentivized to break away and reallocate their offices amongst themselves only.

Exercise 2 (10 points)

Here's a bad alternative to a serial dictatorship, which unfortunately was used to assign kids to elementary schools in a number of major cities for many years.

> **A Bad Mechanism for One-Sided Markets**
>
> 1. Each student submits a ranked list (with no limit on the number of entries).
> 2. The students are ordered in some way. (E.g., by lottery numbers.)
> 3. The students are considered in this order. When student i is considered, if her top-ranked school is still available, then she is (permanently) assigned to that school. (Otherwise, she is not assigned in this phase.)
> 4. The still-unassigned students are considered in the same order as before. When student i is considered, if her second-ranked school is still available, then she is assigned to that school. (Otherwise, she is not assigned in this phase.)
> 5. And so on with the still-unassigned students' third choices, fourth choices, etc.

Discuss in detail what type of strategic behavior (i.e., gaming of the system) you would expect to see from the participants in this mechanism. Do you think the flaws of this mechanism would harm all students equally, or might some demographics be harmed more than others?

Exercise 3 (5 points)

Recall from lecture the stable matching problem, and that a *matching* is an assignment of students to hospitals. (If the capacity of hospital h is c_h, at most c_h students should be assigned to h.) A matching is *Pareto optimal* if every other matching that makes someone better off (i.e., matches them to a preferred student/hospital) also makes someone else worse off. Is every stable matching also Pareto optimal? Provide a proof or an explicit counterexample.

Exercise 4 (5 points)

Is every Pareto optimal matching also stable? Provide a proof or an explicit counterexample.

Exercise 5 (20 points)

In this problem, we'll consider the behavior of algorithms other than deferred acceptance for the standard two-sided stable matching problem. For each of the following two algorithms, prove each of the following statements or find a counterexample. You should assume that we are matching students and universities, there are n students and n universities, and each university has one slot.

i. The algorithm always outputs a Pareto-optimal matching.

ii. The algorithm always outputs a stable matching.

iii. It is in each student's best interest to report their preferences truthfully no matter what other students and universities report.

iv. It is in each university's best interest to report their preferences truthfully no matter what other universities and students report.

(a) Serial dictatorship (10 points)

1. Initialize a temporary matching $M := \emptyset$.
2. Pick the lexicographically next student s who is unmatched in M.
3. Match s to her favorite university that isn't matched in M.
4. Repeat from step 2 until all students are matched.

(b) Weighted matching (10 points)

Noticing the superficial similarities between the stable matching problem and bipartite matching, we might be tempted to turn the former into an instance of the latter. We might end up with an algorithm like this:

1. Define $R_s(u)$, the rank of u for s, to be the number of universities that s prefers to u. Similarly, define $R_u(s)$, the rank of s for u, to be the number of students that u prefers to s.
2. Define the weight w_{su} of the edge (s, u) to be $R_s(u) + R_u(s)$.
3. Output a minimum-weight perfect matching[1] in the complete bipartite graph with edge weights $\vec{w}$ (breaking ties lexicographically).

[1] This is a perfect matching such that the sum of the weights on the edges selected for the matching is minimum among all possible perfect matchings.