Computational Microeconomics, Practice Final Exam 2

Note: this is based on a **takehome** *exam that I gave previously during the pandemic, and some of the instructions – try out your code, etc. – and topics of questions reflect that.*

In this exam you are asked to provide several linear or (mixed) integer linear programs. You should write these in the MathProg (`.mod`) language you have been using in programming assignments. You are allowed and strongly encouraged to test out these programs on your own, making your own test cases. You do not need to submit your test cases. If you do test your code, you can facilitate grading by including your output for the problem instance given in this exam.

Please read instructions carefully. Be clear, precise, and concise in your answers. Rambling or "shotgun" answers where someone tries to say a bunch of different things, perhaps in the hope that something about it is right, will receive no points. For most questions, either you get it right or you don't, so please be careful and try to get it right.

You are allowed to ask clarification questions. However, we cannot help you with questions such as "I'm stuck" or "Can you tell me what is wrong with my formulation?"

Good luck!! –Vince

Problem 1: True or False.

Label each of the following statements as true or false. You are not required to give any explanation.

1. Weighted bipartite matching can be reduced to linear programming.
2. When preferences are single-peaked and the number of voters is odd, there is always a Condorcet winner, and always choosing the Condorcet winner as the winner of the election is strategy-proof.
3. Some people believe that the right answer to the Sleeping Beauty puzzle is 1/2, and others believe that it is 1/3.
4. A 2×2 game that has no weakly dominated strategies must have at least 2 Nash equilibria.
5. In a subgame perfect Nash equilibrium, there cannot be (what could reasonably be considered) a threat that is not credible.
6. The revenue equivalence theorem implies that Myerson's auction has the same expected revenue as the second-price sealed-bid auction (when bidders are risk neutral and valuations are drawn uniformly from $[0, 1]$).
7. With interdependent valuations, in a Vickrey auction, if you bid truthfully (your current estimate of the item's value to you), you will sometimes regret it.
8. The revelation principle implies that there does not exist a non-truthful auction that gives strictly more expected revenue than every truthful auction (assuming the solution concept is Bayes-Nash equilibrium in both cases).
9. In the generalized Vickrey auction (for combinatorial auctions), it is possible that if you add additional bids, the revenue will become (strictly) *lower*.
10. The Myerson-Satterthwaite impossibility result suggests that sometimes, a deal that would benefit both parties will not happen, due to strategic reasons.

Problem 2: Finding a Pareto improvement in assigning items.

In this problem, we will consider agents that have *additive* valuations. That just means that there are no substitutabilities or complementarities; the value of a bundle to an agent is simply the sum of the values of the individual items to the agent. We assume that there is no money; all the agents can do is trade items (a bartering setting). Nevertheless, we want to come up with the most efficient trade. That is, we want to rearrange all the items in such a way that each agent is better off, and moreover, we want to *maximize* the *minimum* improvement of any agent. That is, we evaluate the quality of a reallocation by how much utility the agent that gained the *least* gained. We assume all agents' utilities are common knowledge – so, they do not need to be reported.

For example, suppose agent 1 values item A at 2, item B at 2, and item C at 2, and currently owns item A; agent 2 values item A at 4, item B at 1, and item C at 0, and currently owns items B and C. Then, the optimal solution is to give items B and C to 1 (so that her utility is increased by $(2+2)-2=2$) and item A to agent 2 (so that his utility is increased by $4-(1+0)=3$). The quality of this solution is the minimum improvement, which is 2.

a. Give an example where in the optimal solution, an item ends up with an agent who does not value it the most (i.e., another agent has higher valuation for that item). You should do so by **changing** the above example, in particular by changing only **one** number.

b. Give an integer program for the problem of computing the optimal reallocation. Write it in the MathProg language, and **include** as the data portion the above example (the original one, not your modified one). It should start:

```
set AGENTS;
set ITEMS;
param value{i in AGENTS, j in ITEMS};
param currently_owns{i in AGENTS, j in ITEMS};
var obtains{i in AGENTS, j in ITEMS}, binary;
```

Here, `currently_owns[i,j]` denotes that i owns j before the reallocation and `obtains[i,j]` that i owns j after the reallocation.

Problem 3: Choosing bus routes.

A city decides to drastically reduce its bus routes. It used to have routes A, B, C, D, E, F, G, but it has decided to cut 3 of these routes, leaving only 4. To determine which 4 routes to keep, several agents are asked how much they value the routes. Agents generally want a *bundle* of bus routes, since sometimes they can only get where they want to go by transferring from one bus to another. Suppose the bids for bundles are the following (each bid belonging to a different agent, representing how much that agent values that bundle).

1. ({A,B,C}, 5)
2. ({A,D,E,F},10)
3. ({A,C,G},3)
4. ({B,G}, 3)
5. ({E,F,G},6)

Note that, unlike in a combinatorial auction, in this case goods are *nonrival*: if a bus route is chosen to be kept, *all* the agents who wanted that bus route will be able to use it. The difficulty here comes from only being able to choose 4. Note also that *all* the routes in an agent's bundle need to be included in order for the agent to get any positive value. The goal is to maximize the total value generated by the chosen bus route (maximize social welfare).

a. Find the optimal solution for the above example.

b. Compute the VCG (Clarke mechanism) payments for the above example.

c. Give an integer program for the problem of computing the optimal set of bus routes. Write it in the MathProg language, and **include** as the data portion the above example. (Your formulation does not need to calculate the VCG payments.) It should start:

```
set ROUTES;
set BIDS;
param max_number_of_routes_allowed;
param contained_in{i in ROUTES, j in BIDS};
param value{j in BIDS};
var kept{i in ROUTES}, binary;
```

You may wish to introduce an additional type of variable.

Problem 4: The distancing dilemma.

Two people are approaching each other on a narrow trail. They can't come close to each other. On both sides of the trail there is mud. If one person goes and stands in the mud, they can pass each other. (If they both go into the mud on their respective sides, they can also pass each other.) So, initially, both players have a choice between Mud (M) and Trail (T). If you choose T and the other M, you'll get a utility of 2. If you choose M (in which case you'll definitely get to pass), you'll get a utility of 1 – you'll pass but have muddy shoes.

However, if *both* choose T, the situation hasn't resolved yet – they'll still be stuck. Realizing that this has happened, they will get another chance where, again, they can choose M or T. The payoffs will be the same as above, except if they *again* both choose T, they'll have to give up and turn around, for a utility of 0. (Let's say otherwise an officer will come and tell both of them to go home, for causing trouble.) So there are 2 rounds of the game in this case, and never more.

a. Draw the above game in extensive form. Hint: While this is (at most) a 2-round game in which in both rounds the players choose simultaneously, only one person can move at a node in an extensive-form game. Thus, for each of the rounds, you'll have to sequentialize the moves by the players, but you can make them effectively simultaneous within each round by not letting the other person learn what the first person did in the same round (but after the first round completes, they'll know everything that happened up to that point). So the tree will be four moves deep on the side where in the first round they both choose T.

b. Give the normal form of this game. Note that it should be a 4×4 game because the strategy must specify an action at every information set (even if that information set is not reachable given the other part of the strategy). There should be a fair amount of repetition in the matrix.

c. Give a pure-strategy subgame-perfect Nash equilibrium in which player 1 obtains utility 2. The strategies should be ones from your game in **b**.

d. Solve for a symmetric subgame-perfect Nash equilibrium of this game (which will involve randomization). (An equilibrium is symmetric if the row and column player use the same strategy.) Hint: you can do this by a sort of backward induction: first solve for an equilibrium of the second round (after both choose T), and then replace this subgame by the values of that equilibrium, and solve the first round.

Problem 5: An approval voting Bayesian game.

There are three alternatives (a, b, and c) and two voters (1 and 2). The voters will use approval voting (approve or disapprove every alternative, alternative with most approvals wins) to determine which alternative wins. If there is a tie, then it is broken uniformly at random.

A voter's type is given by a vector (u_a, u_b, u_c) of utilities that that voter will get for the alternatives. Voter 1 has two possible types: $\theta_{11} = (1, 0.8, 0)$ and $\theta_{12} = (0, 0, 1)$. They occur with probability 0.8 and 0.2, respectively. Voter 2 has two possible types: $\theta_{21} = (0.8, 1, 0)$ and $\theta_{12} = (0, 0, 1)$. They occur with probability 0.8 and 0.2, respectively. So, most of the time player 1 prefers a (but with b close behind) and most of the time player 2 prefers b (but with a close behind), but for each of them, with some probability, he/she only likes c.

Two helpful insights are the following. For an alternative that gives you your maximum possible utility (in this case that is always 1), strategically, it never hurts you to approve it. Similarly, for an alternative that gives you your minimum possible utility (in this case that is always 0), strategically, it never hurts you to *not* approve it. This rules out many possible strategies; the only strategic question remaining is whether you should approve an alternative that gives you utility somewhere strictly between 0 and 1.

Thus, the key question becomes: if you have the first (more common) type, do you approve only your top choice, or also your second choice? Doing the former risks that c gets a larger probability (1/2 instead of 1/3 if the other player has the less common type), but doing the latter is more likely to give you only your second choice if the other player has the more common type.

a. Give the normal form of this game (*not* including strategies that do not make sense according to the above). The game should be a 2×2 game. Please work carefully; this is analogous to what we did in the slides, but a little trickier because the distributions here are not 50-50. Please express the utilities in the normal form as exact fractions, with 375 in the denominators. (You may want to use, e.g., Wolfram Alpha which will return answers in exact fractions if you input them as exact fractions. The numerators should be 273, 287, 291, and 305, not necessarily in that order.)

b. Solve for all Nash equilibria of the game and briefly **explain** why there are no others. (Is there dominance?)

Now consider a different game. It is no longer a Bayesian game (equivalently, you can think of it as each player having only one possible type), but a 3-player game. Specifically, voter 1 now definitely has utilities $(1, 0.8, 0)$ and voter 2 definitely has utilities $(0.8, 1, 0)$ (above, those were their first types). But now there is also a third voter with utilities $(0, 0, 1)$.

c. Give the normal form of this game (*not* including strategies that do not make sense according to the above). If a voter has only one strategy left, you don't need to add that voter as a player in the game (since it's clear what this voter will do anyway). By doing this, the game should be a 2×2 game.

d. Solve for all Nash equilibria of the game. (Is there dominance? Which game does this remind you of?)