CS 473 ✧ Fall 2024

Homework 10

Due Tuesday, **December 3**, 2024 at 9pm Central Time

This is the last graded homework.

---

1. Alex and Bo are playing ***Undercut***. Each player puts their right hand behind their back and raises some number of fingers; then both players reveal their right hands simultaneously. Thus, each player independently chooses an integer from 0 to 5.[1] If the two numbers do not differ by 1, each player adds their own number to their score. However, if the two numbers differ by 1, then the player with the lower number adds *both* numbers to their score, and the other player gets nothing. Both players want to maximize their score and minimize their opponent's score.

   Because Alex and Bo only care about the *difference* between their scores, we can reformulate the problem as follows. If Alex chooses the number i and Bo chooses the number j, then Alex gets M_{ij} points, where M is the following 6×6 matrix:

$$M = \begin{pmatrix} 0 & 1 & -2 & -3 & -4 & -5 \\ -1 & 0 & 3 & -2 & -3 & -4 \\ 2 & -3 & 0 & 5 & -2 & -3 \\ 3 & 2 & -5 & 0 & 7 & -2 \\ 4 & 3 & 2 & -7 & 0 & 9 \\ 5 & 4 & 3 & 2 & -9 & 0 \end{pmatrix}$$

   (In this formulation, Bo's score is always zero.) Alex wants to maximize Alex's score; Bo wants to minimize it.

   Neither player has a good *deterministic* strategy; for example, if Alex always plays 4, then Bo should always play 3. Exhausted from trying to out-double-think each other,[2] they both decide to adopt *randomized* strategies. These strategies can be described by two vectors $a = (a_0, a_1, a_2, a_3, a_4, a_5)^\top$ and $b = (b_0, b_1, b_2, b_3, b_4, b_5)^\top$, where a_i is the probability that Alex chooses i, and b_j is the probability that Bo chooses j. Because Alex and Bo's random choices are independent, Alex's expected score is $a^T M b = \sum_{i=0}^{5} \sum_{j=0}^{5} a_i M_{ij} b_j$.

   (a) Suppose Bo somehow learns Alex's strategy vector a. Describe a linear program whose solution is Bo's best possible strategy vector.

   (b) What is the dual of your linear program from part (b)?

   (c) So what *is* Bo's optimal strategy, as a function of the vector a? And what is Alex's resulting expected score? (You should be able to answer this part even without answering parts (a) and (b).)

---

[1] In Hofstadter's original game, players were not allowed to choose 0 for some reason.

[2] "They were both poisoned. I've spent the last several years building up an immunity to iocaine powder."

(d) Now suppose that Alex knows that Bo will discover Alex's strategy vector before they actually start playing. Describe a linear program whose solution is Alex's best possible strategy vector.

(e) What is the dual of your linear program from part (d)?

(f) **Extra credit:** So what *is* Alex's optimal Undercut strategy, if Alex knows that Bo will know that strategy?

(g) **Extra credit:** If Bo knows that Alex is going to use their optimal strategy from part (f), what is Bo's optimal Undercut strategy?

Please express your answers to parts (a)–(e) in terms of arbitrary $n \times n$ payoff matrices M, instead of this specific example. For each linear program, explain in English the meaning of each variable and each constraint. You may find a computer helpful for parts (f) and (g).

2. A ***three-dimensional matching*** in an undirected graph G is a collection of vertex-disjoint triangles (cycles of length 3) in G. A three-dimensional matching is ***maximal*** if it is not a proper subgraph of a larger three-dimensional matching in the same graph.

(a) Let M and M' be two arbitrary maximal three-dimensional matchings in the same underlying graph G. Prove that $|M| \leq 3 \cdot |M'|$.

(b) Finding the largest three-dimensional matching in a given graph is NP-hard. Describe and analyze a fast 3-approximation algorithm for this problem.

(c) Finding the smallest maximal three-dimensional matching in a given graph is NP-hard. Describe and analyze a fast 3-approximation algorithm for this problem.

3. You are designing a digital circuit where each component can operate in one of two states: ***high signal*** (logical 1) or ***low signal*** (logical 0). The circuit includes several directed connections (edges), each with an associated non-negative weight representing the importance or strength of the connection. For a connection to function properly, the signal must flow from a component in the high signal state to a component in the low signal state, and the total weight of such valid connections should be maximized.

You are given a directed, edge-weighted graph $G = (V, E, w)$ representing the circuit, where each vertex $i \in V$ corresponds to a circuit component, each directed edge $i{\to}j \in E$ corresponds to a connection between components, and the weight $w_{i\to j} \geq 0$ of each edge $i{\to}j$ indicates the importance of that edge.

Now suppose we assign some vertices to state ***high*** and the rest to state ***low***. The ***weight*** of this high/low assignment is the sum of the weights of all edges $i{\to}j$ such that vertex i is high and vertex j is low. Your task is to find a high/low assignment with the maximum possible weight.

(a) Describe a *simple*, self-contained, and efficient randomized algorithm for this problem that finds a high/low assignment whose weight is within a factor of 4 of optimal. No LPs are necessary!

(b) Prove that the following integer linear program (ILP) is an exact formulation of our assignment problem: every high/low assignment to the vertices gives an ILP solution whose objective value is at least the weight of the assignment, and every ILP solution gives an assignment whose weight is at least the objective value.

$$
\begin{array}{lll}
\text{maximize} & \displaystyle\sum_{i\to j\in E} w_{i\to j} z_{i\to j} & \\
\text{subject to} & z_{i\to j} \le x_i & \text{for each edge } i\to j \in E \\
& z_{i\to j} \le 1 - x_j & \text{for each edge } i\to j \in E \\
& z_{i\to j} \in \{0,1\} & \text{for each edge } i\to j \in E \\
& x_i \in \{0,1\} & \text{for each vertex } i \in V
\end{array}
$$

(c) The LP relaxation of the above ILP is obtained by replacing the integer constraints with $0 \le z_{i\to j} \le 1$ for each edge $i\to j$ and $0 \le x_i \le 1$ for each vertex i. Consider the following randomized rounding algorithm:

i. First solve the LP relaxation of the ILP from part (b). Let (z^*, x^*) denote the solution to this LP.

ii. Then independently assign each vertex i a state of **high** with probability $\frac{1}{4} + \frac{x_i^*}{2}$ and **low** otherwise.

Prove that in expectation, this algorithm yields a 2-approximation to the optimal value for our assignment problem.