CSCI-570 Spring 2023

Exam 2

INSTRUCTIONS

- The duration of the exam is 140 minutes, closed book and notes.
- No space other than the pages on the exam booklet will be scanned for grading!
- If you require an additional page for a question, you can use the extra page provided at the end of this booklet. However please indicate clearly that you are continuing the solution on the additional page.

1. **True/False Questions (10 points)**

Mark the following statements as **T** or **F**. No need to provide any justification.

a) **(T/F)**. In the Ford-Fulkerson algorithm, the choice of augmenting paths can affect the number of iterations.

True

b) **(T/F)**. If all edges in a graph have capacity 1, then the Ford-Fulkerson algorithm runs in polynomial time.

True

c) **(T/F)**. If an LP problem has a solution at all, it will have a solution at some corner of the feasible region.

True. If an LP problem has an optimal solution, then it must occur at one of the corner points (or vertices) of the feasible region. In other words, the optimal solution of an LP problem always lies at the intersection of two or more constraint boundaries.

d) **(T/F)**. The weak duality theorem in LP implies that if the primal LP problem is infeasible, then the dual LP problem is also infeasible.

False. The weak duality theorem in LP does not make any statement about the feasibility of the primal or dual LP problems.

e) **(T/F)**. In LP, if the primal problem has a unique optimal solution, the dual problem also has a unique optimal solution.

False

f) **(T/F)**. If a primal LP has an unbounded solution, then the dual of that problem is infeasible.

True

g) **(T/F)**. If SAT $\leq_p$ 2-Coloring, then $P = NP$.

True

h) **T/F** If a problem is NP-hard, it is also NP-complete.

False. If the solution can be verified in polynomial time, only then

i) **T/F** A 1.01-approximation vertex cover algorithm must find the optimal solution for graph with vertices less than 100

True. $\text{ALG} \leq 1.01\text{OPT} < \text{OPT} + 1$

j) **T/F** To solve bin packing with a first-fit approach, items must be first sorted in order to get a constant approximation.

False. The first-fit approach is 2-approximation algorithm.

2. Multiple Choice Questions (20 points)

Please select the most appropriate choice. Each multiple choice question has a single correct answer.

a) What would be the running time of the Ford-Fulkerson algorithm if we always choose the augmenting path with the least number of edges?

a) $O(E)$
b) $O(EV)$
c) $O(E^2V)$
d) $O(EV^2)$

c)

b) Which of the following statements is true?

a) The capacity of a cut is the sum of the capacities of all the edges that cross the cut.
b) For every node in a network, the total flow into that node equals the total flow out of that node.
c) The max flow in a network is less than or equal to the total capacity of all edges directly connected to the sink.
d) The max flow in a network is less than the total capacity of the edges in the min cut.

c)

c) Which of the following statements is correct?

a) An edge connecting s to t is always saturated when maximum flow is reached.
b) If all capacities in a flow network are integers, then every maximum flow has integer flows on edges.
c) If in a flow network all edge capacities are distinct, then there exists a unique min-cut.

d) If f is a maximum $s - t$ flow in a flow network, then for all edges out of s are saturated.

a)

d) Let X be a problem that belongs to the class NP. Which one of the following is true?

a) X cannot be solved deterministically in polynomial time.
b) If X can be solved deterministically in polynomial time, then $P = NP$.
c) X may be undecidable.
d) X can be solved in nondeterministic polynomial time.

d)

e) Which of the following statements is true?

a) If X and Y are reducible to each other, then X and Y are NP-Complete.
b) If X and Y are NP-Complete, then X and Y are reducible to each other.
c) NP-Hard is a subset of NP-Complete.
d) 2-SAT problem is NP-Complete.
b)

f) Let X, Y, Z be the problems which can be solved in time $O(\log n)$, $O(n^3)$, $O(2^n)$ respectively (where n denotes the input size). Then, we can NOT conclude that

a) Y is reducible to X.
b) Y is reducible to Z.
c) Z is reducible to Y.
d) If Z is reducible to Y, then Z is reducible to X.

c)

g) Consider the linear program

$$\max(31x + 55y)$$

subject to:

$$\begin{aligned} 2x - y &\leq 80 \\ y &\geq x \\ x &\geq 0 \\ y &\geq 0 \end{aligned}$$

Which of the following statement(s) are true?

a) the linear program has a single optimal solution.
b) the linear program has infinitely many optimal solutions.
c) the linear program has no optimum solution.
c)

h) What is the feasible region in linear programming?

a) The set of all optimal solutions to the LP problem
b) The feasible region is the set of all feasible solutions that satisfy the constraints of the LP problem.
c) The set of all feasible solutions that satisfy the objective function
d) The set of all solutions that satisfy the objective function
b). The feasible region is the set of all feasible solutions that satisfy the constraints of the LP problem. It is the common region determined by all the constraints in the linear programming problem.

i) Consider a linear program having an objective $max\, c^T x$, and let its dual have the objective $min\, b^T y$. Let x and y be some feasible solutions of the primal and dual respectively. Then,

a) $c^T x \geq b^T y$.

b) $c^T x \leq b^T y$.
c) $c^T x = b^T y$.
d) Any definitive comparison for the objectives is not possible.

b)

j) A feasible solution of Linear Programming

a) Must satisfy all the constraints.
b) Does not need satisfy all the constraints, but only some of them.
c) Must be a corner point of the feasible region.
d) Must be the maximum of the objective function.

a)

3. Linear Programming (15 points)

A company produces three products, Product X, Product Y, and Product Z. The company has a limited amount of resources, including production time, labor, and materials. The goal is to maximize profits subject to resource constraints. The following information is available:

- Each unit of Product X requires 1 hour of production time, 1 hour of labor, and 2 units of raw materials. Each unit of Product Y requires 2 hours of production time, 2 hours of labor, and 3 units of raw materials.
- Each unit of Product Z requires 3 hours of production time, 4 hours of labor, and 2 units of raw materials.
- The company has 200 hours of production time, 250 hours of labor, and 150 units of raw materials available.
- The profit per unit of Product X is $50, the profit per unit of Product Y is $60, and the profit per unit of Product Z is $80.
- The company must produce at least 50 units of Product X, 75 units of Product Y, and 25 units of Product Z. The company can only produce a maximum of 125 units in total.

Formulate a linear programming model to help the company maximize profits. You do not have to solve the resulting LP.

a) Describe what your LP variables represent (3 points).

b) Show your objective function (3 points).

c) Show your constraints (9 points).

solution:

a) Let x be the number of units of Product X to produce, y be the number of units of Product Y to produce, and z be the number of units of Product Z to produce.

b) The objective function to maximize profits can be expressed as:

$$Maximize \qquad 50x + 60y + 80z$$

c) The constraints are as follows:

$$x + 2y + 3z \leq 200 \text{ (production time constraint)}$$
$$x + 2y + 4z \leq 250 \text{ (labor constraint)}$$
$$2x + 3y + 2z \leq 150 \text{ (raw materials constraint)}$$
$$x \geq 50 \text{ (minimum production of Product X)}$$
$$y \geq 75 \text{ (minimum production of Product Y)}$$
$$z \geq 25 \text{ (minimum production of Product Z)}$$
$$x + y + z \leq 125 \text{ (total production capacity constraint)}$$
$$x, y, z \geq 0 \text{ (non-negativity constraint)}$$

Rubrics (15 points):

- Define the correct LP variables (3 pts)
- Define the correct objective function (3 pts)
- Define the correct constraints (9 pts). -1 pt for each missing constraint.

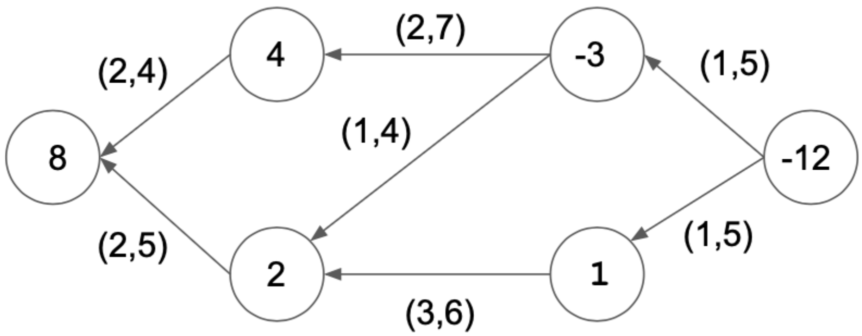


Figure 0.1: Question 4

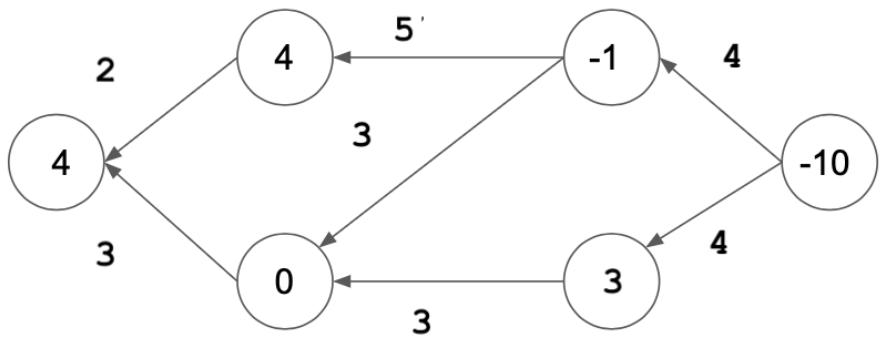
4. Network Flow (20 points)

In the network above, the demand values are shown on vertices (supply values are negative). Lower bounds on flow and edge capacities are shown as (lower bound, capacity) for each edge. Determine if there is a feasible circulation in this graph. Please complete the following three steps below.

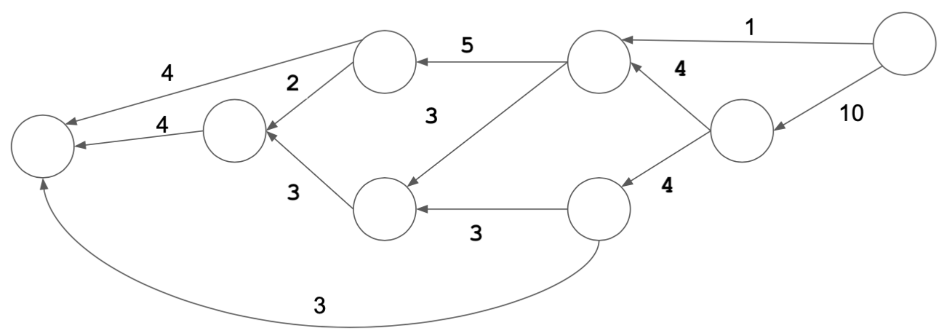
a) Turn the circulation with lower bounds problem into a circulation problem without lower bounds (8 points).

b) Turn the circulation with demands problem into the max-flow problem (7 points).

c) Does a feasible circulation exist? Explain your answer (5 points).



a)



b)

c) No, a feasible solution does not exist. Because the max-flow version of the problem has a max flow of 9, which is less than the sum of positive demands in the circulation problem w/o lower bounds.

5. NP Completeness (20 points)

An organization is arranging an event spread over n days, namely $D_1, D_2, \ldots, D_n$, and needs volunteers to manage the event. On day i, at least v_i volunteers are needed. A total of m people, namely $P_1, P_2, \ldots, P_m$, have applied to volunteer with each applicant indicates the availability. An applicant, if selected, must work on all the days that he or she has indicated. The organization wants to select as few volunteers as possible. Please complete the following five questions.

a) Phrase the above optimization problem as a decision problem and show that it belongs to NP. (5 points)

b) Show a polynomial time construction using a reduction from Vertex Cover.(6 points)

c) Write down the claim that the Vertex Cover problem is polynomially reducible to the original problem. (3 points)

d) Prove the claim in the direction from the reduced problem to the Vertex Cover problem. (3 points)

e) Prove the claim in the direction from the Vertex Cover problem to the reduced problem. (3 points)

a) Given the Event Management instance and a number k, is it possible to satisfy the complete volunteering requirement with at most k applicants?

To show in NP, the certificate is the set of at most k applicants to be selected. Assigning them to all the respective available days and checking if each day has v_i volunteers can be trivially done in polynomial time.

b) Construction: We have a graph $G = (V, E)$ as an input to the VC problem.
   - For each edge e in E, we construct a day D_e. (1 point)
   - For each node v in V, we construct an applicant P_v. (1 point)
   - Each applicant is available for all the days(edges) incident on the corresponding vertex (equivalently, for each day D_e where $e = (u, v)$, only the two applicants P_u and P_v are available). (1.5 points)
   - v_i for each day D_i is simply 1. (1.5 points)
   The construction is done trivially in polynomial time.

c) Claim: VC instance has a vertex cover of size at most$k$, if Event Management can be done with at most k volunteers.

d) If the event management can select at most k applicants, we pick the vertices corresponding to these applicants. Since for each edge at least 1 applicant available that day was chosen, the corresponding vertex covers that edge, thus, the k vertices cover all the selected nodes.

e) If the VC instance has a cover of size k, we simply select the applicants corresponding to these nodes. For each day, we have satisfied its requirement of 1 volunteer since the corresponding edge is covered by at least one of the selected k vertices.

a) 5 points for correct answer. 3 points if incorrect/missing NP argument or decision version of the problem.

b) 6 points for correct answer, -1.5 points for missing $v_i = 1$. 3 points for partially correct answer. 0 for incorrect answer.

c) 3 points for correct argument, 1.5 for partially correct answer or

d) 3 points if correct, **provided part b) is also correct**, else zero

e) 3 points if correct, **provided part b) is also correct**, else zero.

6. **Approximation Algorithm (15 points)**

There are N products with infinite supply available and the price of each one of product i is p_i, $1 \le p_i \le 100$. You have a \$100 gift card, and to better utilize it, you try to buy a bunch of products to maximize the total value, subject to the limit \$100. For example, if there are 3 products with prices $\$49.99, \$89.99, \$50$, then the optimal solution is to buy two product 3, and the total value is \$100.

a) Suppose every price is more than 50, that is, for every i we have $50 < p_i \le 100$. Prove that you can find the optimal solution in $O(N)$ time. (4 points)

b) Suppose there exists a price no more than 50, that is, there exists an i so that $1 \le p_i \le 50$. Prove that buying $\lfloor 100/p_i \rfloor$ product i alone is a $\frac{2}{3}$-approximation algorithm. Here $\lfloor x \rfloor$ is the greatest integer less than or equal to x, for example, when $p_i = 30$, you buy $\lfloor 100/30 \rfloor = 3$ product i. (6 points)

c) Without the assumptions in a) and b), design a $\frac{2}{3}$-approximation algorithm that runs in $O(N)$. Prove your results. (5 points)

a) Since every price is more than 50, we can only buy one product with amount 1.Thus, the optimal solution is buying one product with the highest price, which can be found in $O(N)$ time.

b) We consider two cases: $p_i > \frac{100}{3}$ or $p_i \le \frac{100}{3}$. If $p_i > \frac{100}{3}$, we have $\lfloor 100/p_i \rfloor = 2$ as $3p_i > 100$ and $2p_i \le 100$ by the assumption that $p_i \le 50$. Therefore, the approximation ratio is

$$\frac{2p_i}{\text{OPT}} \ge \frac{2p_i}{100} \ge \frac{200/3}{100} = \frac{2}{3}.$$

On the other hand, by the definition of the floor function, we have

$$p_i\left(\lfloor 100/p_i \rfloor + 1\right) \geq 100 \Rightarrow p_i \lfloor 100/p_i \rfloor \geq 100 - p_i,$$

so the approximation ratio is (recall that $p_i \leq \frac{100}{3}$ in this case)

$$\frac{p_i \lfloor 100/p_i \rfloor}{\text{OPT}} \geq \frac{p_i \lfloor 100/p_i \rfloor}{100} \geq \frac{100 - p_i}{100} \geq \frac{200/3}{100} = \frac{2}{3}.$$

c) Consider the following algorithm: if there exists an i so that $1 \leq p_i \leq 50$, buy $\lfloor 100/p_i \rfloor$ product i; otherwise, buy one product with the highest price. The algorithm runs in $O(N)$ as both finding i and the highest price can be done in linear time. From b) we know that this is a $\frac{2}{3}$ approximation if such an i exists; otherwise, we know that the solution is optimal from the solution of a). Therefore, we conclude the algorithm is a $\frac{2}{3}$ approximation overall.

- a) 4 points; 1 point for saying amount ≤ 1, 2 points for the algorithm and mentioning that buying one product with the highest price is optimal, 1 point for the runtime (no explanation is fine but must say the runtime is $O(N)$/ linear)
- b) 6 points; 3 points for $p_i > \frac{100}{3}$ and 3 points for $p_i \leq \frac{100}{3}$; in $p_i > \frac{100}{3}$, 1 point for $\lfloor 100/p_i \rfloor = 2$ and 2 points for the remaining inequalities; in $p_i \leq \frac{100}{3}$, 1 point for the total value $\geq 100 - p_i$ and 2 points for the remaining inequalities; give 3 points if identify the worst case is $\frac{100}{3}$ (or 34, 33.33) *and* correctly calculate the approximation ratio
- c) 5 points; 2 points for the algorithms, 1 point for the runtime, and 2 points for the proof; no more deduction on the algorithm

if the wrong algorithm in a) is used; no deduction for using the approximation in b) even if the proof in b) is missing or wrong

Additional space

Additional space