Algorithm Design and Analysis
Assignment 5
Deadline: Jun 10, 2024

1. (25 points) **[Common System of Distinct Representatives]** Given a ground set $U = \{1, \ldots, n\}$ and a collection of k subsets $\mathcal{A} = \{A_1, \ldots, A_k\}$, a *system of distinct representatives* of $\mathcal{A}$ is a "representative" collection T of distinct elements from the sets in $\mathcal{A}$. Specifically, we have $|T| = k$, and the k *distinct* elements in T can be ordered as $u_1, \ldots, u_k$ such that $u_i \in A_i$ for each $i = 1, \ldots, k$. For example, $\{A_1 = \{2, 8\}, A_2 = \{8\}, A_3 = \{4, 5\}, A_4 = \{2, 4, 8\}\}$ has a system of distinct representatives $\{2, 4, 5, 8\}$ where $2 \in A_1, 4 \in A_4, 5 \in A_3, 8 \in A_2$, while $\{A_1 = \{2, 8\}, A_2 = \{8\}, A_3 = \{4, 8\}, A_4 = \{2, 4, 8\}\}$ does not have a system of distinct representatives.

   (a) (10 points) Design a polynomial time algorithm to decide if $\mathcal{A}$ has a system of distinct representatives.

   (b) (15 points) Given a ground set $U = \{1, \ldots, n\}$ and two collections of k subsets $\mathcal{A} = \{A_1, \ldots, A_k\}$ and $\mathcal{B} = \{B_1, \ldots, B_k\}$, a *common system of distinct representatives* is a collection T of k elements that is a system of distinct representatives of both $\mathcal{A}$ and $\mathcal{B}$. Design a polynomial time algorithm to decide if $\mathcal{A}$ and $\mathcal{B}$ have a common system of distinct representatives.

   For each part, prove the correctness of your algorithm, and analyze its time complexity.

   Answer:

   Part (a): System of Distinct Representatives for $\mathcal{A}$

   Algorithm Design

   To determine if $\mathcal{A}$ has a system of distinct representatives (SDR), we can use the concept of bipartite matching in graph theory.

   1. Construct the Bipartite Graph:

   - Let $G = (V, E)$ be a bipartite graph where $V = U \cup \{A_1, A_2, \ldots, A_k\}$.

   - There is an edge (A_i, u) if and only if $u \in A_i$.

   2. **Find a Maximum Matching:**

   - Use a maximum bipartite matching algorithm to find a matching in this bipartite graph. This can be done using the Hopcroft-Karp algorithm, which runs in $O(\sqrt{V}E)$.

   3. **Check for Perfect Matching:**

   - After finding the maximum matching, check if the size of the matching is k. If it is, then there is an SDR; otherwise, there is not.

   Algorithm Steps

1. Construct the bipartite graph G with vertex sets U and $\{A_1, \ldots, A_k\}$.

2. Apply the Hopcroft-Karp algorithm to find the maximum matching.

3. If the maximum matching size is k, return True (SDR exists). Otherwise, return False.

Correctness and Complexity Analysis

- Correctness: A perfect matching in the bipartite graph corresponds to selecting one distinct representative from each A_i such that all representatives are distinct.

- Time Complexity: Constructing the graph takes $O(nk)$ time (since each subset A_i can have at most n elements). The Hopcroft-Karp algorithm runs in $O(\sqrt{n+k} \cdot (nk)) = O(n^{3/2}k)$ since $|V| = n + k$ and $|E| = O(nk)$.

Thus, the overall time complexity is $O(nk + n^{3/2}k)$, which is polynomial.

Part (b): Common System of Distinct Representatives for $\mathcal{A}$ and $\mathcal{B}$

Algorithm Design

To determine if $\mathcal{A}$ and $\mathcal{B}$ have a common system of distinct representatives (CSDR), we can use a network flow approach.

1. Construct the Bipartite Graph:

- Create a bipartite graph G with vertex sets $\{A_1, \ldots, A_k, B_1, \ldots, B_k\}$ and U.

- Add edges (A_i, u) if $u \in A_i$ and (B_i, u) if $u \in B_i$.

2. Add Source and Sink:

- Add a source vertex s and a sink vertex t.

- Add edges from s to each A_i and B_i with capacity 1.

- Add edges from each $u \in U$ to t with capacity 1.

- Add edges from each A_i and B_i to u if $u \in A_i$ or $u \in B_i$ with capacity 1.

3. **Max-Flow Computation:**

- Apply a max-flow algorithm (such as the Edmonds-Karp algorithm) to find the maximum flow from s to t.

4. **Check for Required Flow:**

- If the maximum flow value is $2k$ (since each of the A_i and B_i must be matched), then a CSDR exists. Otherwise, it does not.

Algorithm Steps

1. Construct the bipartite graph with additional source s and sink t.

2. Add appropriate edges with capacities.

3. Compute the maximum flow using the Edmonds-Karp algorithm.

4. Check if the maximum flow value is $2k$. If so, return True (CSDR exists). Otherwise, return False.

Correctness and Complexity Analysis

- Correctness: The maximum flow algorithm ensures that we find a way to select k representatives from $\mathcal{A}$ and k from $\mathcal{B}$ such that all are distinct.

- Time Complexity: Constructing the graph takes $O(nk)$ time. The Edmonds-Karp algorithm runs in $O(VE^2)$. Here, $V = 2k + n + 2$ and $E = O(nk)$, so the complexity is $O((2k + n)^3)$.

Thus, the overall time complexity is $O((2k + n)^3)$, which is polynomial.

2. (25 points) **[Hall Marriage Theorem]** Given a bipartite graph $G = (A, B, E)$ with $|A| = |B| = n$, *Hall Marriage Theorem* states that G contains a perfect matching (i.e., a matching with size n) if and only if $|N(S)| \geq |S|$ for any $S \subseteq A$, where $N(S) = \{b \in B \mid \exists a \in S \subseteq A : (a, b) \in E\}$ is the set of all the neighbors of the vertices in S. In this question, we are going to prove Hall Marriage Theorem.

   (a) (5 points) Prove the necessity part: if G contains a perfect matching, then $|N(S)| \geq |S|$ for any $S \subseteq A$.

   (b) (10 points) Construct a directed weighted graph with vertex set $V = A \cup B \cup \{s\} \cup \{t\}$. The edge set is defined as follows. For each $a \in A$, construct a directed edge (s, a) with weight 1; for each $a \in A$ and $b \in B$, construct a directed edge (a, b) with weight ∞ if (a, b) is an edge in the original graph G; for each $b \in B$, construct a directed edge (b, t) with weight 1. Prove that, if $|N(S)| \geq |S|$ holds for all $S \subseteq A$, the minimum s-t cut has value n.

   (c) (10 points) Prove the sufficiency part: if $|N(S)| \geq |S|$ holds for all $S \subseteq A$, then G contains a perfect matching.

Answer:

Part (a):

Assume that G contains a perfect matching M. A perfect matching is a matching where every vertex in A and B is matched to exactly one vertex in the other set.

Consider any subset $S \subseteq A$. Let $N(S)$ denote the set of neighbors of S in B.

Since M is a perfect matching, every vertex in S is matched to a unique vertex in $N(S)$. Therefore, the vertices in S are all matched to distinct vertices in $N(S)$.

This implies that the number of distinct vertices in $N(S)$ must be at least the number of vertices in S. Hence, $|N(S)| \geq |S|$.

Part (b):

Assume that $|N(S)| \geq |S|$ for all $S \subseteq A$.

In the flow network G', we want to find the minimum s-t cut. According to the max-flow min-cut theorem, the value of the maximum flow from s to t is equal to the capacity of the minimum s-t cut.The capacity of any cut (S, T) where $s \in S$ and $t \in T$ is given by the sum of the capacities of the edges crossing the cut.

Consider the cut $S = \{s\} \cup A$ and $T = B \cup \{t\}$:

- The edges crossing this cut are (s, a) for all $a \in A$ and (b, t) for all $b \in B$.

- The capacity of each edge (s, a) is 1, and there are n such edges, so the total capacity from s to A is n.

- The capacity of each edge (b, t) is 1, and there are n such edges, so the total capacity from B to t is n.

Therefore, the total capacity of the cut (S, T) is n.

To prove this is the minimum cut, assume for contradiction that there exists a cut with capacity less than n: - Such a cut would need to separate s from t. - If S' is a subset of A that does not include all of A, then $N(S')$ in B must have at least $|S'|$ vertices by the assumption $|N(S')| \geq |S'|$. - Any cut separating s from t must account for at least n edges, each with capacity 1, as there are n nodes in A each contributing 1 to the cut capacity from s to A.

Hence, the minimum s-t cut must be at least n, proving that the minimum s-t cut in G' has value n.

Part (c):

Using the result from part (b), we constructed a flow network G' and showed that if $|N(S)| \geq |S|$ for all $S \subseteq A$, then the minimum s-t cut in G' has value n.

According to the max-flow min-cut theorem, the value of the maximum flow in G' is equal to the capacity of the minimum cut, which is n.

A flow of value n in G' corresponds to a situation where all vertices in A and B are matched uniquely. This is because:

- Each vertex $a \in A$ must send one unit of flow to a vertex $b \in B$.

- Each vertex $b \in B$ must receive one unit of flow from a vertex $a \in A$.

- Since the total flow value is n, each vertex in A and B is matched exactly once, forming a perfect matching.

Therefore, the existence of a maximum flow of value n implies the existence of a perfect matching in the original bipartite graph G.

Thus, if $|N(S)| \geq |S|$ for all $S \subseteq A$, then G contains a perfect matching, proving the sufficiency part.

3. (25 points) For the linear program

$$\begin{aligned} \text{maximize } & x_1 - 2x_3 \\ \text{subject to } & x_1 - x_2 \leq 1 \\ & 2x_2 - x_3 \leq 1 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

prove that the solution $(x_1, x_2, x_3) = (3/2, 1/2, 0)$ is optimal.

Answer:

To prove that the solution $(x_1, x_2, x_3) = \left(\frac{3}{2}, \frac{1}{2}, 0\right)$ is optimal for the given linear program, we will use the concept of duality in linear programming. We will show that the primal solution and the dual solution satisfy the complementary slackness conditions, which implies that the primal solution is optimal.

The dual linear program (LP) is:

$$\begin{aligned} \text{minimize } & y_1 + y_2 \\ \text{subject to } & y_1 \geq 1 \\ & -y_1 + 2y_2 \geq 0 \\ & -2y_2 \geq -2 \\ & y_1, y_2 \geq 0 \end{aligned}$$

Simplifying the dual constraints:

$$\begin{gathered} y_1 \geq 1 \\ -y_1 + 2y_2 \geq 0 \implies y_1 \leq 2y_2 \\ -2y_2 \geq -2 \implies y_2 \leq 1 \\ y_1, y_2 \geq 0 \end{gathered}$$

Now, we need to find a feasible solution to the dual LP and verify the complementary slackness conditions.

Consider $(y_1, y_2) = (1, 1)$. Check if this satisfies the dual constraints:

$$\begin{aligned} & y_1 = 1 \geq 1 \quad \text{(satisfied)} \\ & y_1 = 1 \leq 2 \cdot 1 = 2 \quad \text{(satisfied)} \\ & y_2 = 1 \leq 1 \quad \text{(satisfied)} \end{aligned}$$

This means that $(y_1, y_2) = (1, 1)$ is a feasible solution to the dual LP. The objective value of this dual solution is:

$$y_1 + y_2 = 1 + 1 = 2$$

Next, we calculate the objective value of the given primal solution $\left(\frac{3}{2}, \frac{1}{2}, 0\right)$:

$$x_1 - 2x_3 = \frac{3}{2} - 2 \cdot 0 = \frac{3}{2}$$

Notice that the dual objective value 2 is greater than the primal objective value $\frac{3}{2}$. According to the weak duality theorem, the primal objective value cannot exceed the dual objective value. Therefore, $\frac{3}{2} \leq 2$, which is consistent with weak duality.

To conclude the optimality, we verify the complementary slackness conditions: 1. For the primal constraint $x_1 - x_2 \leq 1$:

$$\left(\frac{3}{2} - \frac{1}{2} = 1\right) \implies \text{slack is } 0 \implies y_1 \cdot 0 = 0 \quad \text{(complementary slackness)}$$

2. For the primal constraint $2x_2 - x_3 \leq 1$:

$$\left(2 \cdot \frac{1}{2} - 0 = 1\right) \implies \text{slack is } 0 \implies y_2 \cdot 0 = 0 \quad \text{(complementary slackness)}$$

Both complementary slackness conditions are satisfied. Therefore, by the complementary slackness theorem, $\left(\frac{3}{2}, \frac{1}{2}, 0\right)$ is an optimal solution to the primal problem.

4. (25 points) **[König-Egerváry Theorem]** In the class, we have seen that the maximum matching problem can be formulated by the following linear program

$$\begin{aligned} \text{maximize} \; & \sum_{e \in E} x_e & \\ \text{subject to} \; & \sum_{e: e=(u,v)} x_e \leq 1 & (\forall v \in V) \\ & x_e \geq 0 & (\forall e \in E) \end{aligned}$$

and the minimum vertex cover problem can be formulated by the following linear program

$$\begin{aligned} \text{minimize} \; & \sum_{u \in V} x_u & \\ \text{subject to} \; & x_u + x_v \geq 1 & (\forall (u,v) \in E) \\ & x_u \geq 0 & (\forall u \in V) \end{aligned}$$

We have also mentioned that the second linear program is the dual program of the first (try to verify it on your own).

(a) (20 points) Prove that both linear programs have integral optimal solutions if the graph is bipartite.

(b) (5 points) Using the result in the first part, prove König-Egerváry Theorem, which states that the size of the maximum matching equals to the size of the minimum vertex cover in a bipartite graph.

Answer:

To prove that both the maximum matching and the minimum vertex cover linear programs have integral optimal solutions in bipartite graphs, we can use concepts from linear programming and combinatorial optimization. Specifically, we will use the fact that these linear programs have total unimodularity and the properties of bipartite graphs.

Part (a): Integral Optimal Solutions for Bipartite Graphs

Total Unimodularity

To show that these linear programs have integral optimal solutions in bipartite graphs, we need to show that their constraint matrices are totally unimodular. A matrix is totally unimodular if every square submatrix has a determinant of 0, +1, or -1. For bipartite graphs, the incidence matrix A (where $A_{ve} = 1$ if vertex v is incident to edge e, and 0 otherwise) is totally unimodular.

In bipartite graphs:

- Each vertex constraint in the maximum matching LP can be written as a row in the incidence matrix, and each edge corresponds to a variable.

- Each edge constraint in the minimum vertex cover LP can be written as a row in the incidence matrix, with vertex variables on each side of the bipartite partition.

Since the incidence matrix of a bipartite graph is totally unimodular, the constraint matrices of both linear programs are also totally unimodular.

Integrality of Solutions

By a fundamental theorem in linear programming, if a linear program has a totally unimodular constraint matrix and all the right-hand side coefficients are integers, then the vertices (extreme points) of the feasible region of the linear program are integral. Therefore, the linear programs for both maximum matching and minimum vertex cover have integral optimal solutions when the graph is bipartite.

Part (b): Proving König-Egerváry Theorem

Duality and Optimality

From linear programming duality, we know that the optimal value of a linear program equals the optimal value of its dual. Specifically, the maximum matching linear program is the primal, and the minimum vertex cover linear program is the dual.

Since both linear programs have integral optimal solutions in bipartite graphs, we have:

- The optimal value of the maximum matching linear program (an integer) equals the size of the maximum matching.

- The optimal value of the minimum vertex cover linear program (an integer) equals the size of the minimum vertex cover.

Equality of Sizes

Due to strong duality, the optimal values of the primal and dual programs are equal. Thus, the size of the maximum matching equals the size of the minimum vertex cover in bipartite graphs.

Therefore, we have proven the König-Egerváry Theorem using the integrality of the solutions to the linear programs and the duality theorem of linear programming.

5. How long does it take you to finish the assignment (including thinking and discussion)? Give a score (1,2,3,4,5) to the difficulty. Do you have any collaborators? Please write down their names here.

   It takes me about 8 hours to finish this assignment because of the unfamiliarity with NetworkFlow and LinearProgramming considering my previous courses.I would give 4 to the difficulty including some of my selfishness.