EXERCISE 1
Design the dual of the following Linear Problem:

max c T x

such that

a T i x ≤ bi , 1 ≤ i ≤ m'
a T j x = bj , m' + 1 ≤ j ≤ m
xk ≥ 0, 1 ≤ k ≤ r

where x = (x1, x2 . . . xn), x ∈ Rn

This exercise will be solved by hand in order to follow the Lagrangian method used in lectures.

<u>EXERCISE 2</u>

<u>Show that all linear programs can be expressed as:</u>

min c T x

Ax = b

x ≥ 0

where x = (x1, x2 . . . xn), , x ∈ Rn.

We must demonstrate that any linear program can be converted into an equivalent form by following the next structure:

min c^T x
subject to
Ax = b
x ≥ 0

where A is an m x n matrix, b is an m-dimensional vector, and x is an n-dimensional vector.

—---

1)

We can begin by considering a linear program in standard form, which has the following form:

min c^T x
subject to
Ax = b
x ≥ 0

where A is an m x n matrix, b is an m-dimensional vector, and x is an n-dimensional vector.

By introducing slack variables, we can convert the linear program to standard form if it is not already. Take, for instance, a linear program of the form:

min c^T x
subject to
Ax ≤ b
x ≥ 0

where A is an m x n matrix, b is an m-dimensional vector, and x is an n-dimensional vector.

We can introduce slack variables such that any inequality constraint in the form of Ax <= b is transformed into Ax + si = b by introducing each slack variable si, which corresponds to the i-th constraint in A.

Then, the linear program becomes:

min c^T x
subject to
Ax + s = b
x, s ≥ 0

This is in standard form and can be expressed in the desired form as:

min c^T [x, s]^T
subject to
[A I] [x, s]^T = b
[x, s] ≥ 0

where I is the m x m identity matrix. Therefore, any linear program can be expressed in the next form:

min c T x → min c^T [x, s]^T
Ax = b → [A I] [x, s]^T = b
x ≥ 0 → [x, s] ≥ 0

—---

2)

The idea of duality in linear programming is another way to demonstrate that any linear program can be expressed in the given form.

Taking as an example the standard linear program:

minimize cTx
with respect to Ax = b and x 0,

where c is the linear objective function's coefficient vector
x is the variables' vector, A is the constraints' coefficient matrix
b is the constants' vector

The following is a definition of the linear program's Lagrangian function:

L(x, λ) = cTx + λ T(Ax - b)

λ where is a Lagrange multiplicative vector.

The maximization of the Lagrangian function in relation to the Lagrange multipliers is the definition of the dual problem of the linear program:

maximize $g(\lambda)$ = minimum $L(x, \lambda)$

where $g(\lambda)$ is the dual function

In optimization, the min-max theorem states that, provided certain conditions are met, the primal problem (the original linear program) and the dual problem both have the same optimal value.

The primal problem must be feasible (it has a solution that can be implemented) and the dual problem must be bounded (it has an upper bound) for the min-max theorem to hold in linear programming.

The dual problem can now be described as:

The dual problem can then be rewritten as:

maximize $g(\lambda) = -Tb$

subject to $\lambda >= 0$ and $AT\lambda <= c$

where AT is the transpose of the matrix A.

We can then rewrite the dual problem as:

minimize $-bT\lambda$

subject to $A^T\lambda - c \leq 0$ and $\lambda \geq 0$

where λ is a vector of Lagrange multipliers.

This is a linear program in the form:

minimize $dT\lambda$

subject to $E\lambda = f$ and $\lambda \geq 0$

where d is the vector -b, E is the matrix A^T, and f is the vector c.

Thus, we have shown that any linear program can be expressed in the form:

minimize cTx

subject to $Ax = b$ and $x \geq 0$

where $x = (x1, x2 . . . xn)$, $x \in Rn$

EXERCISE 3

We consider an attacker on your flow network. Consider a bipartite network N(E, d) where E = {e1, e2, · · · , em} . We denote your (the designer's) budget by BD. The flow design is represented by a vector f ∈ [0, 1]m with the requirement Pm i=1 f[i] = 1, where f[i] is the flow amount on edge ei . For each edge ei with flow amount f[i], the designer's cost is d[i]f[i]. We say a design f is within budget if Pm i=1 d[i]f[i] ≤ BD. An attack is a vector X ∈ [0, 1]m indicating the attack on edges, where X[i] denotes the attack, or attack level, on edge ei , the fraction of flow captured on ei . The adversary's benefit is the sum of flow she captures over the bipartition and defined as Pk j=1 f[j] · X[j]. Your goal is to minimize the attacker's benefit. Given the adversary's strategy X[j] as fixed and provided, describe a linear program to optimize the designer's strategy and find its dual.

The following formula will derive a linear program that can maximize the designer's strategy and reduce the attacker's advantage:

minimize Pk j=1 f[j] · X[j]

subject to:

Pm i=1 f[i] = 1 (flow conservation constraint)
Pm i=1 d[i]f[i] ≤ BD (budget constraint)
0 ≤ f[i] ≤ 1 for all i (non-negativity constraint)

This is a standard linear program in the form:

minimize cT x

subject to Ax = b and x ≥ 0

where the optimization vector c is given by c = X, the flow conservation, budget, and non-negativity constraints are combined in the constraint matrix A, and the flow conservation and budget constraints are combined in the constraint vector b.

By introducing Lagrange multipliers for the constraints and maximizing the resulting Lagrangian function, the dual of this linear program can be obtained.

This is the dual program:

maximize λ1

subject to λ1 + λ2d[i] ≤ X[i] for all i, and λ1, λ2 ≥ 0

where λ1 and λ2 are Lagrange multipliers for the flow conservation and budget constraints, respectively.

The dual program can be simplified to:

<u>Objective function:</u>
maximize λ1

<u>Constraints:</u>
subject to λ1 + λ2d[i] ≤ X[i] for all i

where λ1 and λ2 are non-negative Lagrange multipliers.

A lower bound on the primal program's optimal value can be found in the dual program's optimal value. The dual program can be viewed as the attacker's problem of maximizing his benefit within the confines of her attack not exceeding the designer's budget and flow conservation requirements.

Solving the primal and dual programs will give the optimal flow design and the optimal attack strategy, respectively. The solution to the dual program can also provide insights into the sensitivity of the optimal flow design to changes in the attack strategy, which can be useful in designing robust flow networks.

EXERCISE 4

Set up a linear program for determining a shortest path in a weighted directed graph.

A variant of the shortest path algorithm known as the Bellman-Ford algorithm can be used to create a linear program for locating the shortest path in a weighted directed graph. The algorithm works on a graph with a set of vertices and edges, each of which has a weight that represents how much it costs.

From an initial node, the Bellman-Ford algorithm determines the shortest route to all other nodes in the graph. In order to accomplish this, it keeps a set of distances d(v) between the initial node and each vertex v of the network. These distances are initialized to infinity for all vertices except the initial node, which is set to 0. After that, the algorithm goes through the shortest route from the initial node to each network vertex v.

Next, the algorithm iterates through all edges (u, v) of the network and updates the distance d(v) if the distance through u is less than the current distance to v. The update rule is given by:

$d(v) = \min(d(v), d(u) + w(u, v))$

where w(u, v) is the weight of the edge (u, v).

After n-1 iterations, where n is the number of vertices in the graph, the algorithm terminates and the set of distances d(v) gives the shortest path from the initial node to all other nodes in the graph.

To formulate this algorithm as a linear program, we can introduce a variable d(v) for each vertex v of the network, which represents the shortest distance from the initial node to v. Then, the linear program can be written as

Objective function:
minimize d(t)

Constraints:
$d(s) = 0$
$d(v) \leq d(u) + w(u, v)$ for all edges (u, v) of the graph

where s is the initial node, t is the target node and w(u, v) is the weight of edge (u, v).

Find the dual of this LP.

This is a standard form of linear program, in which we seek to minimize the objective function d(t), subject to linear constraints. The dual of this linear program can be obtained by introducing dual variables y(s), y(v) for each vertex v of the graph and formulating the dual problem as:

Objective function:
maximize $y(s)$

Constraints:
$y(v) - y(u) \leq w(u, v)$ for all edges (u, v) of the network
$y(v) \geq 0$ for all vertices v of the graph.

The dual problem seeks to maximize the dual objective function $y(s)$, subject to constraints that are dual to the original constraints. The dual problem constraints are obtained by transposing the coefficient matrix and exchanging the roles of the primal variables d and the dual variables y. The duality theorem guarantees that the optimal solution of the dual problem is equal to the optimal value of the primal problem, and the solutions of both problems satisfy a set of conditions known as duality constraints.

The dual problem can be reached by using the steps learned in class with the Lagrangian approach.

We first introduce a Lagrange multiplier $\lambda(u, v)$ for each edge (u, v) and then form the Lagrangian function as follows:

$L(d, \lambda) = d(t) + \sum\lambda(u, v)[d(v) - d(u) - w(u, v)]$.

The dual problem is obtained by minimizing $L(d, \lambda)$ with respect to d, and maximizing it with respect to λ.

Minimizing $L(d, \lambda)$ with respect to d yields:

$\partial L/\partial d(u) = \lambda(u, v) - \lambda(v, u) = 0$ for all nodes u and v except for the source node s and the destination node t.

This implies that $\lambda(u, v) = \lambda(v, u)$ for all nodes u and v except for s and t, which can be written as:

$\lambda(u, v) = -\lambda(v, u)$ for all edges (u, v) except for edges incident to s and t.

Substituting this back into $L(d, \lambda)$ and simplifying, we obtain:

$L(d, \lambda) = d(t) - \sum[\lambda(u, v) * w(u, v)]$.

Maximizing $L(d, \lambda)$ with respect to λ gives the dual problem as follows:

Maximize $L(d, \lambda) = d(t) - \sum[\lambda(u, v) * w(u, v)]$

subject to:

$\lambda(u, v) = -\lambda(v, u)$ for all edges (u, v) except for edges incident to s and t.

$\lambda(v) \geq 0$ for all nodes v.

PRIMARY PROBLEM
Objective function:
minimize d(t)

Constraints:
d(s) = 0
d(v) ≤ d(u) + w(u, v) for all edges (u, v) of the graph

λ(u, v) = -λ(v, u) for all edges (u, v) except for edges incident to s and t.

λ(v) ≥ 0 for all nodes v.

DUAL PROBLEM
Objective function:
maximize y(s)

Constraints:
y(v) - y(u) ≤ w(u, v) for all edges (u, v) of the network
y(v) ≥ 0 for all vertices v of the graph.

EXERCISE 5

Complete the proof from Class: Compute the dual of linear program for the maximum flow problem using path structures instead of flow conservation at vertices. Use variables for flows on s-t paths, where a path is the sequence of edges from s to t. Explain how this can be used to prove the strong duality of the maximum flow problem.

Will be proved by hand in order to follow with the class proof.

EXERCISE 6

A system $Ax \leq b$, $x \in Rn$, $A \in Rm \times n$ may be infeasible, i.e. has no solution. Show that this is true if there exists a $y \in Rm$ such that $AT\ y = 0$, $bT\ y < 0$, and $y \geq 0$.

To show that a system $Ax \leq b$, $x \in Rn$, $A \in Rm \times n$ is infeasible if and only if there exists a $y \in Rm$ such that $AT\ y = 0$, $bT\ y < 0$, and $y \geq 0$, we need to prove both implications separately.

First, suppose that the system $Ax \leq b$, $x \in Rn$, $A \in Rm \times n$ is infeasible, i.e., there does not exist x satisfying all constraints.

We can now use Farkas' Lemma, which states that exactly one of the following two statements is satisfied:

- ➢ There exists $x \in Rn$ such that $Ax = b$ and $x \geq 0$.
- ➢ There exists $y \in Rm$ such that $AT\ y \leq 0$ and $bT\ y < 0$.

Since the system $Ax \leq b$ is infeasible, we know that the first statement will not be the satisfied one, which leads us to think that the second statement must be satisfied. Therefore, there will exist a $y \in Rm$ such that $AT\ y \leq 0$ and $bT\ y < 0$.

Both sides of the equation can be multiplied $AT\ y \leq 0$ by -1 to obtain $AT\ y = 0$ and $y \geq 0$, which is the desired form.

Note that:

$-AT\ y \geq 0$, is the same as $AT\ y = 0$ if $y \geq 0$.

Now we need to prove the other direction, namely, that if there exists a $y \in Rm$ such that $AT\ y = 0$, $bT\ y < 0$, and $y \geq 0$, then the system $Ax \leq b$ is infeasible. Suppose by contradiction that there exists an $x \in Rn$ such that $Ax \leq b$. Then we have:

$AT\ Ax \leq AT\ b$

Since $AT\ y = 0$, we can multiply both sides by y to obtain:

$0 \leq yT\ AT\ Ax \leq yT\ AT\ b$

Since $y \geq 0$ and $bT\ y < 0$, we have $yT\ AT\ b < 0$. But this contradicts the inequality $yT\ AT\ Ax \geq 0$. Therefore, there can be no such x, which means that the system $Ax \leq b$ is infeasible.

Therefore, we have shown that the system $Ax \leq b$ is infeasible if and only if there exists a $y \in Rm$ such that $AT\ y = 0$, $bT\ y < 0$, and $y \geq 0$.

EXERCISE 1

By following the Lagrange method given in lectures.

Objective function: maximize $c^T x = g(x)$

Constraints:
$f_i(x) = b_i - a_i^T x \geq 0 \Rightarrow a_i^T x \leq b_i \;, \quad 1 \leq i \leq m'$
$h_i(x) = b_j - a_j^T x = 0 \Rightarrow a_j^T x = b_j \;, \quad m'+1 \leq j \leq m$
$x_k \geq 0 \quad 1 \leq k \leq r$

The LAGRANGE function: $L(x, \lambda, \mu) = g(x) + \sum_i \lambda_i f_i(x) + \sum_j \mu_j h_j(x) =$

$= c^T x + \lambda^T x + \lambda^T (b - Ax) + \mu^T (b - Ax) = b^T(\lambda + \mu) - \lambda^T (Ax)$
$- \mu^T (Ax) + \lambda^T x + c^T x$

$\Phi(\lambda, \mu) = \sup [b^T(\lambda+\mu) + c^T x + \lambda^T x - (\lambda+\mu)^T (Ax)] =$

$b^T(\lambda+\mu) + \sup [(c^T + \lambda^T - (\lambda+\mu)^T A) x]$

* Minimizing $\Phi(\lambda, \mu) \Rightarrow (c^T + \lambda^T - (\lambda+\mu)^T A) = 0$
because if $\underbrace{(c^T + \lambda^T - (\lambda+\mu)^T A}$ $\neq 0 \Rightarrow \Phi(\lambda, \mu) = \infty$
So cancelling the $\uparrow$ term, $\Rightarrow \Phi(\lambda, \mu) = b^T(\lambda + \mu)$ to be minimized.

DUAL PROBLEM

objective function: minimize $(b^T(\lambda+\mu))$

Constraints:
$\lambda(A - I) + \mu A = c$
$\mu, \lambda \geq 0$

Exercise 5

Continuing with the mathematical proof started in class we can get the DUAL linear program for the maximum flow problem using "path structures"

$c = (c_1, c_2, \dots, c_e)$ capacities of edges

Objective function: maximum $e^T f$

Constraints: $\begin{cases} f_i(x) = c - Hf \\ f \geq 0 \end{cases}$

The Lagrangian approach for DUAL:

$H = \{ \}$ paths

$$L(f, \lambda, \mu) = e^T f + \lambda^T (c - Hf) + \mu^T (f)$$

$$\Phi(\lambda, \mu) = \sup [e^T f + \lambda^T (c - Hf) + \mu^T (f)] = \lambda^T c +$$

$$\sup [(e^T - \lambda^T H + \mu^T) f] \quad \lambda \geq 0 \quad \mu \geq 0$$

Avoiding $\Phi(\lambda, \mu) = \infty \rightarrow \Phi(\lambda, \mu) = \lambda^T c$

Assuming $e^T + \lambda H - \mu^T = 0$

Objective function: minimum $\lambda^T c$

Constraints: $H^T \lambda = e + \mu^T \Rightarrow H^T \lambda \geq e$

$\lambda \geq 0, \ \mu \geq 0$

Min $c^T \lambda$
$H^T \lambda \geq e, \ \lambda \geq 0$ $\Rightarrow$ min $\sum c_e \lambda_e$ subject to $\sum \lambda_e \geq 1$

MIN CUT THEOREM

By strong duality LP solution $\rightarrow$ Dual LP solution

$\rightarrow$ Max-Flow $\rightarrow$ Min-Cut

* In order to prove the strength of the duality of the max-flow problem, we can compare the primar problem to the dual one, the min-cut one.

The dual linear program refers to the minimum s-t cut method to achieve the maximum flow. Hence, as the solution to one, is the optimal solution to the other, we can assert that the duality is strong.

- PRIMAL LINEAR PROGRAM $\rightsquigarrow g(f) = e^T f \Rightarrow$ max-flow
- DUAL LINEAR PROGRPM $\rightsquigarrow \Phi(\lambda, \mu) = \lambda^T c \Rightarrow$ min-cut

$\Phi(\lambda^*, \mu^*) = g(f^*)$ $\lambda^*, \mu^*, f^* \Rightarrow$ optimal solution

* As covered in lectures materials and confirmed on the optimization theories, in order to have a strong duality, next conditions to be met are necessary:
* The objective function must be differentiable
* $g(f)$ shoud be convex, and the optimization problem also,
* The constraint functions are differentiable
* Feasible set contains an inferior point: Slater's condition

Julen's comment: "Path structures"-based approach is equivalent to the standard one using flux conservation, but may be more convenient in certain situations. For the exact case of this exercise, it can facilitate the proof of the strong duality theorem as it has been shown.