Tianyi Li
1000929445

CSC373 Fall 2017 Assignment 3

1. In the same way that we can express LPs in standard form, we can express an IP and its dual in standard form. Show that weak duality holds for integer programs in standard form.
   - **An integer programming or IP is a linear programming problem with integer values.**
   - **The standard form for an integer program is:**

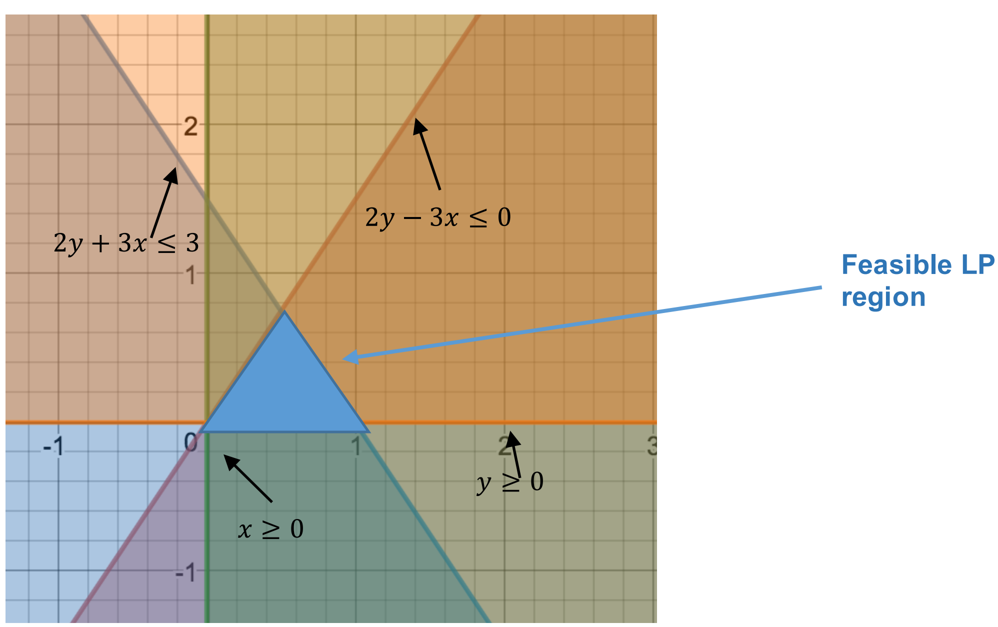
     Maximize $\boldsymbol{c^T x}$
     Subject to $\boldsymbol{Ax \le b}$
     $\boldsymbol{x \ge 0, x \in \mathbb{Z}^n}$
   - **The dual for an integer program is:**

     Minimize $\mathbf{y^T b}$
     Subject to $\boldsymbol{y^T \mathrm{A} \le c^T}$
     $\boldsymbol{y \ge 0, y \in \mathbb{Z}^n}$
   - **Let $\bar{x}$ be a feasible solution to the primal integer program,**
   - **Let $\bar{y}$ be a feasible solution to the dual integer program,**
   - **To show that the weak duality holds for integer programs in standard form, that means we need to prove $\boldsymbol{c^T x \le \mathrm{y}^T \mathrm{b}}$.**
   - **Since $\bar{x}$ is feasible, and it is primal,**
   - **Then** $\boldsymbol{A\bar{x} \le b}$**,**
     $\boldsymbol{\bar{y}^T A\bar{x} \le \bar{y}^T b}$**.**
   - **Since $\bar{y}$ is dual,**
   - **Then** $\boldsymbol{c^T \le \bar{y}^T A}$**,**
     $\boldsymbol{c^T\bar{x} \le \bar{y}^T A\bar{x}}$**.**
   - **From both above $\boldsymbol{c^T\bar{x} \le \bar{y}^T A\bar{x}}$, and $\boldsymbol{\bar{y}^T A\bar{x} \le \bar{y}^T b}$,**
   - **Then** $\boldsymbol{c^T\bar{x} \le \bar{y}^T A\bar{x} \le \bar{y}^T b}$**.**
   - **Therefore,** $\boldsymbol{c^T\bar{x} \le \bar{y}^T b}$**.**

2. Consider the following primal IP and LP in standard form:

   Maximize y
   Subject to $2y - 3x \le 0$
   $2y + 3x \le 3$
   $y, x \ge 0$

   For the IP, add the constraints that x and y are integral. Draw the feasible LP region.



a) What are the vertices of the feasible region of the primal LP?

- **Let** $2y - 3x = 0$ $\quad 2y + 3x = 3$ $\quad y = 0$ $\quad x = 0$
- **Then** $2y = 3x$ $\quad 2y = 3 - 3x$

  $y = \frac{3}{2}x$ $\quad y = \frac{3}{2} - \frac{3}{2}x$
- $\begin{cases} y = \frac{3}{2}x \\ y = \frac{3}{2} - \frac{3}{2}x \end{cases} \Rightarrow (\frac{1}{2}, \frac{3}{4})$
- $\begin{cases} y = \frac{3}{2}x \\ x = 0 \\ y = 0 \end{cases} \Rightarrow (0, 0)$
- $\begin{cases} y = \frac{3}{2} - \frac{3}{2}x \\ y = 0 \end{cases} \Rightarrow (1, 0)$
- **Therefore, the vertices of the feasible region are** $(\frac{1}{2}, \frac{3}{4})$**,** $(0, 0)$ **and** $(1, 0)$**.**

b) What is the optimal fractional solution?

- $y = \frac{3}{4}$

c) What is the optimal integral solution?

- $y = 1$

d) Provide the dual IP of the primal IP.

- $c = (0 \quad 1) \Rightarrow c^T = \binom{0}{1}$
- $A = \begin{pmatrix} -3 & 2 \\ 3 & 2 \end{pmatrix} \Rightarrow A^T = \begin{pmatrix} -3 & 3 \\ 2 & 2 \end{pmatrix}$
- $b = \binom{0}{3} \Rightarrow b^T = (0 \quad 3)$

- **Minimize** $3y$
- **Subject to** $3y - 3x \geq 0$

  $2y + 2x \geq 1$

  $y, x \geq 0$

e) Use this example to show that strong duality does not necessarily hold for integer programming.

- **Strong duality: the primal optimal solution equals to the dual optimal solution.**
- **In this case, the primal optimal solution is** $y = 1$**.**
- **To calculate the vertices of the feasible region of the dual LP:**
- **Let** $3y - 3x = 0$ $\quad 2y + 2x = 1$ $\quad y = 0$ $\quad x = 0$
- **Then** $3y = 3x$ $\quad 2y = 1 - 2x$

  $y = x$ $\quad y = \frac{1}{2} - x$
- $\begin{cases} y = x \\ y = \frac{1}{2} - x \end{cases} \Rightarrow (\frac{1}{4}, \frac{1}{4})$
- $\begin{cases} y = x \\ x = 0 \\ y = 0 \end{cases} \Rightarrow (0, 0)$
- $\begin{cases} y = \frac{1}{2} - x \\ y = 0 \end{cases} \Rightarrow (\frac{1}{2}, 0)$
- **Then the vertices of the feasible region are** $(\frac{1}{4}, \frac{1}{4})$**,** $(0, 0)$ **and** $(\frac{1}{2}, 0)$**.**
- **Then the optimal fractional solution is** $y = \frac{1}{4}$**.**

- **Then the optimal integral solution is $y = 0,\ 3y = 0$.**
- **The dual optimal solution is $y = 0$.**
- **The primal optimal solution does not equal to the dual optimal solution.**
- **Therefore, strong duality does not necessarily hold for integer programming.**

3. Consider the following weighted partial vertex cover problem (WPVC):
Given: A graph $G = (V, E)$ with vertex weights $w: V \to \mathbb{R}^{\geq 0}$ and edge costs $c: E \to \mathbb{R}^{\geq 0}$. In the standard vertex cover problem, we need to choose a subset $V' \subseteq V$ so that every edge $e = (u, v)$ has to be covered in the sense that at least one of its endpoints u or v must be in the cover V'. In the WPVC problem, not every edge has to be covered but an uncovered edge e costs $c(e)$. The objective it to minimize the weights of the partial cover plus the costs of uncovered edges.

   a) Provide a $\{0, 1\}$ IP for the WPVC problem.
   - **Let W_i be the weight of the vertex V_i**
   - **Minimize** $\sum_{i \in V} w_i\, x_i$
   - **Subject to** $x_i + x_j \geq 1$ **for every edge** $(i, j) \in E$
     $x_i \in \{0, 1\}$ **for every vertex** $i \in V$

   b) Provide an approximation guarantee for WPVC by the IP/LP rounding method.
   - **Here is an LP from the IP in part a):**
   - **Minimize** $\sum_{i \in V} w_i\, x_i^*$
   - **Subject to** $x_i^* + x_j^* \geq 1$ **for every edge** $(i, j) \in E$
     $0 \leq {x^*}_i \leq 1$ **for every vertex** $i \in V$
   - **Compute the LP optimum solution x^* to part a),**
   - **Need to make sure that part a) has to be a feasible solution to IP.**
   - **Let** $S = \{i \in V: x_i^* \geq \frac{1}{2}\}$
   - **Return S.**
   - **Consider an edge** $(i, j) \in E$
   - **Since** $x_i^* + x_j^*$
   - **and either** $x_i^* \geq \frac{1}{2}$ **, or** $x_j^* \geq \frac{1}{2}$
   - **Then** (i, j) **is covered**
   - **If S^* is an optimal vertex cover,** $W(S^*) \geq \sum_{i=1}^{n} w_i\, x_i^*$
     $\geq \sum_{i \in S} w_i\, x_i^*$ **since** $S \subset \{1, \ldots, n\}$
     $\geq \frac{1}{2} \sum_{i \in S} w_i$ **since for all** $i \in S$, $x_i^* \geq \frac{1}{2}$
     $= \frac{1}{2}\, w(S)$
   - **The IP/LP rounding method is a 2-approximation algorithm for WPVC.**

4. Consider a model of computation where arithmetic operations take $O(1)$ time and choosing a random number in the range $\{1, 2, \ldots, 2n\}$ takes time $O(1)$.

   Let A, B and C be $n \times n$ matrices over the ring of integers. We want to verify that $AB = C$ using $O(n^2)$ arithmetic operations.

   a) Provide a randomized algorithm ALG satisfying that: If $AB = C$, ALG will always say YES.
   - **If we are just multiplying two matrices and compare the product with C will take $O(n^3)$ runtime.**
   - **But what we can do is to compute it using the matric-vector multiplication method**
   - **We need to choose a random vector $r \in \{0, 1\}^n$, compute $A(Br)$ and Cr, and compare the 2 results.**
   - **This involves 3 matric-vector multiplication and 1 comparison, which yields $O(n^2)$ runtime.**

Tianyi Li
1000929445

- **If $AB = C$, then $A(Br) = Cr$.**
- **So if the algorithm will only return true if $A(Br) = Cr$.**
- **checking_equality(A, B, C):**

```
        select a random vector r
        x = B r
        y = A x
        z = C r
        if y == z:
              return True
        return False
```

b) Provide a randomized algorithm ALG satisfying that: If $AB \neq C$, ALG will say YES with some constant error probability $p < 1$.
   - **Continue with the idea from part a), in this case, we assume that $AB \neq C$.**
   - **Let $D = AB - C \neq 0$**
   - **Let d_{ij} be the elements in the ith row and jth column in the matrix D, and they do not all equal to zero.**
   - **For each d_{ij}, the possibility p of $d_{ij}r_j$ is not 0 is $\frac{1}{2}$.**
   - **Over n^2 times of multiplication and comparison, p will never get to 1.**
   - **checking_inequality(A, B, C):**

```
        select a random vector r
        x = B r
        y = A x
        z = C r
        D = y - z
        if D != 0:
              return True
        return False
```

c) Show how you can verify $AB = C$ with error probability $\frac{1}{n}$. What is the time complexity of your algorithm?
   - **I do not know the answer to this question.**

5. Consider the exact Max-3-Sat problem:
   Given a propositional CNF formula $F = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ where each clause C_i has exactly 3 literals. The objective is to compute a truth assignment so as to maximize the number of satisfied clauses. Provide a polynomial time randomized algorithm that in expectation will satisfy a $\frac{7}{8}$ fraction of the clauses.
   - **For each random assignment of the 3 literals $x_1, x_2, \ldots, x_m$ in each clause, *they are* independently assigned to either True or False with a probability of $\frac{1}{2}$ each.**
   - **Let *S* equals to the number of satisfied clauses.**
   - ***S* can be written as a sum of random indicator variables S_i for each clause.**
   - $S_i = \begin{cases} 1, & \text{if } C_i \text{ is satisfied} \\ 0, & \text{otherwise} \end{cases}$**, and $S = S_1 + S_2 + \ldots + S_n$**
   - **Then the expected value of S is $E[S] = E[S_1] + E[S_2] + \ldots + E[S_n]$.**
   - **By the definition of 3-SAT, the probability of a clause is not satisfied is $\frac{1}{2}^3 = \frac{1}{8}$.**
   - **Then, the probability of a clause is satisfied is $\frac{7}{8}$.**
   - **Then, $E[S] = \frac{7}{8}n$.**