CSC373 Summer '22

Solutions to Assignment 3: Linear Programming

Due Date: July 21, 2022, 11:59pm ET

Instructions

1. Typed assignments are preferred (e.g., PDFs created using LaTeX or Word), especially if your handwriting is possibly illegible or if you do not have access to a good quality scanner. Either way, you need to submit a single PDF named "hwk3.pdf" on MarkUS at `https://markus.teach.cs.toronto.edu/2022-05`
2. You will receive 20% of the points for a (sub)question when you leave it blank (or cross off any written solution) and write "I do not know how to approach this problem." If you leave it blank but do not write this or a similar statement, you will receive 10%. This does not apply to any bonus (sub)questions.
3. You may receive partial credit for the work that is clearly on the right track. But if your answer is largely irrelevant, you will receive 0 points.

Q1 [30 Points] Max Flow with Losses

The *Maximum Flow with Losses* problem is similar to the maximum flow problem: you are given as input a directed graph $G = (V, E)$ with source s and sink t, except here, in addition to the graph, each vertex $u \in V - \{s, t\}$ has a real number called the *loss coefficient* $\varepsilon_u \in [0, 1]$ such that the total flow out of u must equal $(1 - \varepsilon_u)$ times the total flow into u. As before, we are looking for an assignment of flow values to every edge that maximizes the total flow out of s.

(a) [15 Points] Show how to solve the maximum flow with losses problem using linear programming. Give a detailed description of your linear program and justify clearly and carefully that it solves the problem.

(b) [15 Points] Convert the linear program above into the standard form, and describe how a solution to this modified linear program would lead you to a solution of your original linear program. More precisely, specify these quantities: n, m, an $n \times 1$ coefficient vector c, an $m \times n$ constraint matrix A, and an $m \times 1$ bound vector b (all numbers are real numbers) such that the linear program from part (a) can be "converted" to the linear program which maximizes $c^T x$ subject to the constraints $Ax \leq b$ and $x \geq 0$. Also, specify clearly which variable of your original linear program does each x_i (the i-th entry of x) play the role of.

Solution to Q1

(a) We adapt the linear program given at the bottom of page 860 of the text. This LP is associated with a max flow problem given by a directed graph $G = (V, E)$ in which each edge (u, v) in E has a nonnegatve capacity $c(u, v) \geq 0$. We set $c(u, v) = 0$ if (u, v) is not an edge, and in particular $c(u, u) = 0$ for all $u \in V$. A flow is a nonnegative real-valued function $f : V \times V \rightarrow \mathbb{R}$ that satisfies the capacity constraint and flow conservation. A maximum flow is a flow that satisfies these constraints and maximizes the flow value.

We use the variable f_{uv} to stand for $f(u,v)$, for each pair (u,v) of vertices. For this exercise, the linear program is the same as that in the text, except that the node constraint for a node $u \in V - \{s,t\}$ is modified so that the total flow out of u is equal to $(1-\epsilon_u)$ times the total flow into u.
Here is the resulting modified linear program:

- maximize: $\sum_{v\in V} f_{sv} - \sum_{v\in V} f_{vs}$
- subject to:

$$\begin{array}{rcll} f_{uv} & \leq & c(u,v) & \text{for each } u,v \in V, \\ (1-\epsilon_u)\sum_{v\in V} f_{vu} & = & \sum_{v\in V} f_{uv} & \text{for each } u \in V - \{s,t\}, \\ f_{uv} & \geq & 0 & \text{for each } u,v \in V. \end{array}$$

Every valid flow in the network yields a feasible solution to the linear program because flow values satisfy each constraint in the linear program. Hence, the maximum value of the objective function is at least as large as the maximum flow.
Conversely, every feasible solution to the linear program yields a valid flow in the network because every constraint on the flow is represented by some linear inequality. Hence, the maximum value of the objective function is no larger than the maximum flow.

(b) Assume that $V = \{0,1,\ldots,|V|-1\}$. We code a pair (u,v) of vertices using the pairing function $\langle u,v\rangle = u|V| + v + 1$. Thus for $1 \leq i \leq n = |V|^2$ the variable x_i plays the role of the variable f_{uv} in part (a), where $i = \langle u,v\rangle$.
To recover the pair (u,v) from i we use the inverse pairing functions

$$\mathrm{left}(i) = \lfloor (i-1)/|V| \rfloor \qquad\qquad \mathrm{right}(i) = i - 1 - \mathrm{left}(i)\cdot|V|$$

So if $i = \langle u,v\rangle$, then $u = \mathrm{left}(i)$ and $v = \mathrm{right}(i)$.
The objective in part (a) is already a maximization problem - therefore, we do not have to change it to convert to standard form.
Thus we can define the vector c as follows: for $1 \leq i \leq n = |V|^2$,

$$c_i = \begin{cases} -1 & \text{if } \mathrm{right}(i) = s, \\ 1 & \text{if } \mathrm{left}(i) = s \text{ and } \mathrm{right}(i) \neq s, \\ 0 & \text{otherwise.} \end{cases}$$

Next we define an $[m \times n]$ matrix A and an $[m \times 1]$ vector b such that the constraints $Ax \leq b$ and $x \geq 0$ are equivalent to the constraints given in the solution to part (a). For this we keep in mind that in general $y \leq z$ is equivalent to $-y \geq -z$ and $y = z$ is equivalent to the conjunction of $y \geq z$ and $-y \geq -z$.
We partition the rows of A into blocks, and the rows of b into similar blocks, where each pair of blocks is responsible for a different set of constrains, as follows.

- Rows $1 \leq i \leq n := |V|^2$ ensure $x_i \leq c(\mathrm{left}(i), \mathrm{right}(i))$ for each i, and hence $f_{uv} \leq c(u,v)$ for all $u,v \in V$. Thus for $1 \leq i \leq n$ and $1 \leq j \leq n$, let $b_i = c(\mathrm{left}(i), \mathrm{right}(i))$ and

$$A_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

- Rows $n+1 \le i \le n+|V|$ ensure for each $v \in V$ that $(1-\epsilon_v)\sum_{u\in V} f_{uv} \ge \sum_{u \in V} f_{vu}$. (Here we have interchanged the roles of v and u in this set of constraints because of the way we defined the pairing function: note that $v = \text{right}(i-n)$ runs through all of V as i runs from $n+1$ to $n+|V|$.) For $n+1 \le i \le n+|V|$ we define $v = v_i = \text{right}(i-n)$. Thus for these values of i and for $1 \le j \le n$ we set $b_i = 0$ and define

$$A_{ij} = \begin{cases} -(1-\epsilon_{v_i}) & \text{if right}(j) = v_i, \\ 1 & \text{if left}(j) = v_i \text{ and right}(j) \ne v_i, \\ 0 & \text{otherwise.} \end{cases}$$

- Rows $n+|V|+1 \le i \le n+2|V|$ ensure for each $v \in V$ that $-(1-\epsilon_v)\sum_{u\in V} f_{uv} \ge -\sum_{u\in V} f_{vu}$. The definition of A_{ij} for these rows is the same as for the previous case, except the top row has $1-\epsilon_{v_i}$ and the middle row has -1.

Thus the resulting matrix A has $m = |V|^2 + 2|V|$ rows and $n = |V|^2$ columns, and the constraints $Ax \le b$ and $x \ge 0$ are equivalent to the constraints given in part (a).

$t_1, \dots, t_n$ indicates tool uses or not

Q2 [20 Points] Tasks and Tools $t_i = \begin{cases} 1 \\ 0 \end{cases}$ $t_i + t_j \le 1$ if incompatible

You have m different tasks to complete and to help you, n different software tools you could purchase, each with a positive integer cost c_i. You have no choice about which tasks to complete (you must complete them all), but you get to choose which tools you will purchase.
Tools are not necessary to complete tasks; however, certain tasks have additional costs if they are completed without the use of specific tools. Information about these additional costs is provided through non-negative integer *dependencies*: for all pairs i, j, $d_{i,j}$ is the additional cost of completing task i *without* tool j – a dependency can be equal to zero to indicate that there is no additional cost.
Finally, there are known *incompatibilities* between certain tools: for each tool i, you have a list of all the other tools with which tool i is incompatible. Obviously, it is not possible to install incompatible tools at the same time.
Formulate a linear program (or a binary integer program – i.e., an optimization problem with a linear objective, linear constraints, but with each variable restricted to taking a value in $\{0, 1\}$) to determine which tools to purchase in order to minimize your total cost (from the purchase of tools and the dependencies). Also, remember to justify the correctness of your solution – this is important!

$\sum_{i=1}^{v} (c_i - \sum_{j=1}^{n} d_{ji}) t_i$

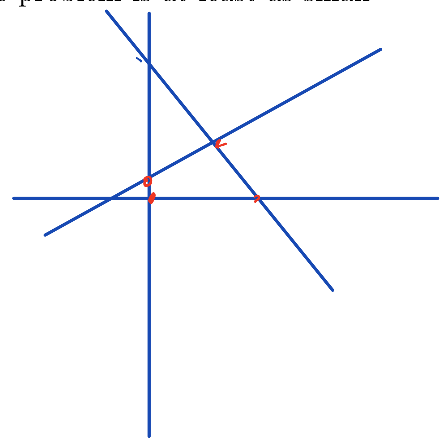
Solution to Q2

Create an Integer Program as follows:

- **Integer Variables:** $t_1, t_2, \dots, t_n$ (intention: $t_i = 1$ if we purchase tool i; $t_i = 0$ otherwise)
- **Constraints:** $0 \le t_i \le 1$ for $i = 1, 2, \dots, n$
  $t_i + t_j \le 1$ for all $i, j \in \{1, 2, \dots, n\}$ such that tools i and j are incompatible
- **Objective Function:** minimize $\sum_{i=0}^{n} \left(c_i - \sum_{j=1}^{m} d_{j,i} \right) \cdot t_i$

Note: $c_i t_i$ is the cost of purchasing tool i (zero if $t_i = 0$); $(1 - t_i) \sum_{j=1}^{m} d_{j,i}$ is the cost of the dependencies for tool i (zero if $t_i = 1$).

Find an optimum solution $t_1^*, t_2^*, \ldots, t_n^*$ for the integer program, and purchase every tool i for which $t_i^* = 1$.
(Note on running time: The time to create the integer program and to output a solution is polynomial. The time to solve the integer program may not be.)
Correctness: Every valid solution to the problem yields a feasible solution to the integer program (by setting $t_i = 1$ if tool i is purchased; $t_i = 0$ otherwise). This implies that the optimum value of the objective function is at least as small as the minimum total cost for the problem.
Conversely, every feasible solution to the integer program yields a valid solution to the problem (by purchasing tool i iff $t_i = 1$). Because of the constraints, we are guaranteed *not* to purchase incompatible tools. This implies that the minimum total cost for the problem is at least as small as the optimum value of the objective function.
Hence, the algorithm correctly solves the problem.



Q3 [10 Points] LP and IP Exercise

Consider the following linear program L in the standard form:

$$\begin{aligned} \text{Maximize} \quad & x_2 \\ \text{Subject to} \quad & -3x_1 + 5x_2 \leq 8 \\ & 7x_1 + 3x_2 \leq 12 \\ & x_1, x_2 \geq 0 \end{aligned}$$

$-3x+5y=8$ $\quad y=\frac{3}{5}x+\frac{8}{5}$

We define the corresponding integer program I as follows:

$$\begin{aligned} \text{Maximize} \quad & x_2 \\ \text{Subject to} \quad & -3x_1 + 5x_2 \leq 8 \\ & 7x_1 + 3x_2 \leq 12 \\ & x_1, x_2 \in \{0, 1\} \end{aligned}$$

$7x+3y\leq 12$

$y=4-\frac{7}{3}x$

Plot the feasible region of L. *Note: You do not need to submit this with the assignment, but it will be helpful to plot the feasible region.* You can use any online graphing programs such as desmos, fooplot, etc.

(a) [2.5 Points] What are the vertices of the feasible region of L? (No explanation is needed.)

(b) [2.5 Points] What are the optimal solutions of L and I? What are the corresponding optimal objective values? (No explanation is needed.)

(c) [2.5 Points] Provide the dual linear program (which we will call L') of L. Clearly indicate which dual variable in your formulation corresponds to which primal constraint.

(d) [2.5 Points] What are the optimal solutions of L' and its corresponding integer program I' (i.e., the program you obtain upon restricting all of the variables of L' to $\{0, 1\}$)? What are the corresponding optimal objective values? Does strong duality hold for this particular pair of primal and dual integer programs i.e., for I and I'?

Solution to Q3

(a) $(0,0)$, $(0,8/5)$, $(12/7,0)$, and $(9/11,23/11)$ (the last vertex is the point where both constraints are met with equality).

(b) The optimal solution of the primal LP L is the $(9/11,23/11)$ vertex, and the optimal objective value is $23/11$.
For the IP I, we note that the only feasible points are $(0,0)$, $(1,0)$, $(0,1)$, and $(1,1)$. So the optimal solutions are $(0,1)$ and $(1,1)$, where the optimal objective value is 1.

(c) Let y_1 and y_2 be the dual variables corresponding to the first and second constraint, respectively. The dual is then:

$$\begin{array}{ll} \text{Minimize} & 8y_1 + 12y_2 \\ \text{Subject to} & -3y_1 + 7y_2 \geq 0 \\ & 5y_1 + 3y_2 \geq 1 \\ & y_1, y_2 \geq 0 \end{array}$$

(d) The optimal dual LP solution is at $(y_1,y_2) = (7/44,3/44)$, where the optimal objective value is $23/11$. This matches the optimal primal LP objective value as expected by strong duality.
For the dual IP, we can check that $(0,0)$ and $(1,0)$ are not feasible solutions, but $(0,1)$ is. The optimal dual IP solution is $(y_1,y_2) = (0,1)$, where the optimal objective value is 12, different from the optimal primal IP objective. This is an example showing that strong duality does not hold for IPs.