COMP 360 - Fall 2015 - Assignment 2

Due: 6:00 pm Oct 13th.

General rules: In solving these questions you may consult books but you may not consult with each other. There are in total 105 points, but your grade will be considered out of 100. You should drop your solutions in the assignment drop-off box located in the Trottier Building.

---

1. (20 Points) Consider a graph where some of the edges have "directions", but some are undirected edges. We want to assign direction to all the undirected edges such that in the resulting directed graph, the incoming degree of every vertex is equal to its outgoing degree. Note that we are not allowed to change the direction of the edges that have directions in the beginning. Show that this problem can be solved using the max-flow problem (Hint: think about the baseball elimination problem).

   Solution: The idea is that every undirected edge ab can contribute 1 to the indegree of a or b, and we want to decide these contributions so that every vertex a will finally have indegree $\deg(a)/2$, where $\deg(a)$ is the number of edges (directed and undirected) incident to a.

   Construct a network flow as in the following. For every vertex a in the graph put a vertex v_a in the network, and for every edge ab in the graph put a vertex v_{ab} in the network. If ab has no direction, then add the edges $v_{ab}v_a$ and $v_{ab}v_b$, each with capacity 1 to the network. If ab has already a direction, say from a to b, then only add the edge $v_{ab}v_b$, and if the direction is from b to a then add $v_{ab}v_a$ instead. Add a source s and connect it to each vertex v_{ab} with an edge of capacity 1. Add a sink t, and for every vertex v_a, add the edge v_at with capacity $\deg(a)/2$.

2. (10 points) Gerlinde Kaltenbrunner wants to pack food for her next expedition, which is going to be three weeks long. She wants to pack a combination of foods A, B, C, D, E so that she has at least 5000 calories per day, and 50-65 % of those calories come as carbohydrates, 20-35 % as fat, and 15-20 % as proteins. The following table shows the calories as carbohydrates, fat, and proteins per 1kg of the different foods A, B, C, D, E. Write a linear program that will help her to decide how much of each food to pack in order to minimize the weight of the food that she is going to carry (You do not need to solve the linear program).

	carbs	fats	proteins
A	2500	3500	500
B	1500	100	3500
C	2500	100	2500
D	3500	1800	200
E	1000	2900	500

   Solution: Let $I = \{A, B, C, D, E\}$. If $i \in I$, let X_i represents the amount in KG of each kind of food, K_i represents the amount of carbs for food item i, F_i the amount of fat for food item i and P_i the amount of proteins for food item i. We have the following LP:

$$
\begin{aligned}
\min \quad & \textstyle\sum_{i\in I} X_i \\
s.t. \quad & \textstyle\sum_{i\in I} X_i(K_i + F_i + P_i) \geq 105000 \\
& 2500X_A + 1500X_B + 2500X_C + 3500X_D + 1000X_E \leq 0.65 \times 105000 \\
& 2500X_A + 1500X_B + 2500X_C + 3500X_D + 1000X_E \geq 0.5 \times 105000 \\
& 3500X_A + 100X_B + 100X_C + 1800X_D + 2900X_E \leq 0.35 \times 105000 \\
& 3500X_A + 100X_B + 100X_C + 1800X_D + 2900X_E \geq 0.2 \times 105000 \\
& 500X_A + 3500X_B + 2500X_C + 200X_D + 500X_E \leq 0.2 \times 105000 \\
& 500X_A + 3500X_B + 2500X_C + 200X_D + 500X_E \geq 0.15 \times 105000 \\
& X_i \geq 0, \forall i \in I
\end{aligned}
$$

3. (a) (5 Points) Draw the feasible region of the following system of linear constraints:

$$
\begin{aligned}
x_1 + x_2 &\geq 0 \\
2x_1 + x_2 &\leq 7 \\
x_1 - x_2 &\geq -2 \\
x_1 - 3x_2 &\leq 0 \\
-x_1 + 2x_2 &\leq 4
\end{aligned}
$$

Solution: Obviously, sketch the lines provided above and choose the region covered by all of them.

(b) (10 Points) What are the vertices of the feasible region? For each vertex, list the two constraints that define that vertex.

Solution: Vertices are, $v_1 = (-1, 1)$, $v_2 = (0, 2)$, $v_3 = (2, 3)$, $v_4 = (3, 1)$, $v_5 = (0, 0)$.

(c) (10 Points) Which vertex maximizes the value of the function $x_1 + 2x_2$? Now suppose that we wanted to find this vertex using the simplex algorithm starting from the vertex $(x_1, x_2) = (0, 0)$. In other words we start at $(0, 0)$ and each time we move to a neighbouring vertex that increases the value of $x_1 + 2x_2$. List the two possible paths that the algorithm can take starting from $(x_1, x_2) = (0, 0)$ and finishing at the optimal vertex. For each path, explain how the linear constraints that define the vertices on the path change as we move from one vertex to the next vertex.

Solution: $v_3 = (2, 3)$ maximizes this objective. One possible path is: start from v_5, use the the 3rd constraint instead of the 4th one and move to v_1. Use the 5th constraint instead of the first one and move to v_2 and finally use the third constraint instead of the second one and move to v_3. Similarly, you can come up with the second path.

4. (10 points) Prove that if $(x_1, \ldots, x_n)$ and $(x'_1, \ldots, x'_n)$ are feasible solutions to a linear program, then so is every point on the line segment between these two points. (Note that the points on the line segment are $(\alpha x_1 + (1-\alpha)x'_1, \ldots, \alpha x_n + (1-\alpha)x'_n)$ where $0 \leq \alpha \leq 1$. Also for simplicity you may assume that all the constraints are of the form $a_1x_1 + \ldots + a_nx_n \leq b$ where $a_1, \ldots, a_n, b \in \mathbb{R}$).

Solution: Obviously

$$
\begin{aligned}
& a_1(\alpha x_1 + (1-\alpha)x'_1) + \ldots + a_n(\alpha x_1 + (1-\alpha)x'_1) \\
& = \alpha(a_1x_1 + \ldots + a_nx_n) + (1-\alpha)(a_1x_1 + \ldots + a_nx_n) \\
& \leq \alpha b + (1-\alpha)b \leq b.
\end{aligned}
$$

5. (25 Points) We want to assign non-negative numbers to the edges of an undirected graph such that they add up to 1, and furthermore the maximum load on a vertex is minimized. Here the load of a vertex is the sum of the numbers on the edges incident to that vertex.

   (a) Consider the (undirected) graph on four vertices a, b, c, d with edges ab, ac, bc, bd, cd. Formulate the above problem as a linear program for this particular graph.
   Solution: Let w_{ab} be the weight associated to ab, we have the following LP,

$$\begin{aligned} & \min \quad w_{\max} \\ s.t. \quad & z : w_{ab} + w_{ac} + w_{bc} + w_{bd} + w_{cd} = 1 \\ & x_a : w_{\max} - w_{ab} - w_{ac} \geq 0 \\ & x_b : w_{\max} - w_{ab} - w_{bc} - w_{bd} \geq 0 \\ & x_c : w_{\max} - w_{ac} - w_{bc} - w_{cd} \geq 0 \\ & x_d : w_{\max} - w_{bd} - w_{cd} \geq 0 \\ & w_{ab}, w_{ac}, w_{bc}, w_{bd}, w_{cd} \geq 0 \end{aligned}$$

   (b) Write the dual of your linear program in part (a).
   Solution: Given above, we have the following,

$$\begin{aligned} & \max \quad z \\ s.t. \quad & z - x_a - x_b \leq 0 \\ & z - x_a - x_c \leq 0 \\ & z - x_b - x_c \leq 0 \\ & z - x_b - x_d \leq 0 \\ & z - x_c - x_d \leq 0 \\ & x_a + x_b + x_c + x_d = 1 \\ & x_a, x_b, x_c, x_d \geq 0 \end{aligned}$$

   (c) Formulate the above problem as a linear program for a general graph $G = (V, E)$.
   Solution: The variables are z and x_{uv} for all the edges $uv \in E$, and the linear program is the following:

$$\begin{array}{lrll} \min & w_{\max} & & \\ \text{s.t.} & \sum_{uv \in E} w_{uv} & = 1 & \\ & w_{\max} - \sum_{uv \in E} w_{uv} & \geq 0 & \forall u \in V \\ & w_{uv} & \geq 0 & \forall uv \in E \end{array}$$

   (d) Write the dual of your linear program in part (c).
   Solution:

$$\begin{aligned} & \max z \\ s.t. \quad & z - x_u - x_v \leq 0 \ \ \forall uv \in E \\ & \textstyle\sum_{v \in V} x_v = 1 \\ & x_v \geq 0 \ \ \forall v \in V \end{aligned}$$

6. (15 Points) Write a linear program for solving the following problem: Given an $n \times n$ matrix A and an n-dimensional vector b, we want to find a vector

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

such that $Ax = b$ and that our solution is as "flat" as possible. More precisely we want to minimize the largest difference $x_i - x_j$ over all i and j (i.e. we want to minimize $\max_{i,j} x_i - x_j$).

Solution:
We solve this by introducing a new variable y

$$\begin{array}{ll} \min & y \\ \text{s.t.} & Ax = b \\ & x_i - x_j \leq y \quad \forall i, j \end{array}$$