APMA 1210 Practice Final Solutions

December 2022

1 Simplex method

We begin by rewriting the LP in a way that allows us to label the constraints:

$$
\begin{aligned}
\text{Maximize: } & z = 3x_1 + x_2 + 3x_3 \\
\text{Subject to: } & 2x_1 + x_2 + x_3 \leq 2 \quad (1) \\
& x_1 + 2x_2 + 3x_3 \leq 5 \quad (2) \\
& 2x_1 + 2x_2 + x_3 \leq 6 \quad (3) \\
& x_1 \geq 0 \quad (4) \\
& x_2 \geq 0 \quad (5) \\
& x_3 \geq 0 \quad (6)
\end{aligned}
$$

We can now quickly check the origin for feasibility. The point $(x_1, x_2, x_3) = (0, 0, 0)$ is in the feasible region, and in particular it is a corner point, because there are 3 variables and 3 tight constraints (4), (5), and (6). So this will be the starting point.

Current vertex: $\{(4), (5), (6)\}, (x_1, x_2, x_3) = (0, 0, 0)$

Objective value: $z = 3(0) + 0 + 3(0) = 0$, this is non-optimal because increasing any of the variables will increase the objective value. Increasing either x_1 or x_3 would provide the largest increase, since they have equally large coefficients in the objective function, so we can choose either variable. We will proceed with x_1 here.

Move: Increasing the selected variable from $x_1 = 0$ to $x_1 = 1$ makes constraint (1) tight, and constraint (4) has been released. We have now stepped to the point (1,0,0) corresponding to the set of tight constraints $\{(1), (5), (6)\}$.

Coordinates: We choose $y_1 = 2 - 2x_1 - x_2 - x_3, y_2 = x_2, y_3 = x_3$ based on the set of tight constraints at this point.

Rewrite LP: Solving back from the new coordinate system, we compute $x_2 = y_2, x_3 = y_3$ and this allows us to find $x_1 = \frac{1}{2}(2 - y_1 - y_2 - y_3)$. Substitute these into the LP to find the new system:

$$\begin{aligned}
\text{Maximize: } & z = 3 - \frac{3}{2}y_1 - \frac{1}{2}y_2 + \frac{3}{2}y_3 \\
\text{Subject to: } & y_1 \geq 0 \quad (1) \\
& -y_1 + 3y_2 + 5y_3 \leq 8 \quad (2) \\
& -y_1 + y_2 \leq 4 \quad (3) \\
& y_1 + y_2 + y_3 \leq 2 \quad (4) \\
& y_2 \geq 0 \quad (5) \\
& y_3 \geq 0 \quad (6)
\end{aligned}$$

Current vertex: $\{(1), (5), (6)\}, (y_1, y_2, y_3) = (0, 0, 0)$

Objective value: $z = 3 - \frac{3}{2}(0) - \frac{1}{2}(0) + \frac{3}{2}(0) = 3$, this is non-optimal since increasing y_3 would increase the objective value. We will proceed by increasing y_3.

Move: Increasing this variable from $y_3 = 0$ to $y_3 = \frac{8}{5}$ releases constraint (6) and tightens constraint (2). We have now stepped to the point $(y_1, y_2, y_3) = (0, 0, \frac{8}{5})$, corresponding to the set of tight constraints $\{(1), (2), (5)\}$.

Coordinates: We choose $w_1 = y_1$ and $w_2 = y_2$ corresponding to the tight constraints (1) and (5), and then we also choose $w_3 = 8 + y_1 - 3y_2 - 5y_3$ according to constraint (2).

Rewrite LP: Solving back gives us $y_1 = w_1$ and $y_2 = w_2$, allowing us to compute $y_3 = \frac{1}{5}(8 + w_1 - 3w_2 - w_3)$. Using these substitutions, we construct the new system:

$$\begin{aligned}
\text{Maximize: } & z = \frac{27}{5} - \frac{6}{5}w_1 - \frac{7}{5}w_2 - \frac{3}{10}w_3 \\
\text{Subject to: } & w_1 \geq 0 \quad (1) \\
& w_2 \geq 0 \quad (2) \\
& -w_1 + w_2 \leq 4 \quad (3) \\
& 6w_1 + 2w_2 - w_3 \leq 2 \quad (4) \\
& w_2 \geq 0 \quad (5) \\
& -w_1 + 3w_2 + w_3 \leq 8 \quad (6)
\end{aligned}$$

Current vertex: $\{(1), (2), (5)\}, (w_1, w_2, w_3) = (0, 0, 0)$

Objective value: $z = \frac{27}{5} - \frac{6}{5}(0) - \frac{7}{5}(0) - \frac{3}{10}(0) = \frac{27}{5}$, this is optimal since the coefficients of all variables are negative.

Optimal vertex: We can now translate our current vertex, at which the optimal value was achieved, back to the original coordinate system. We already know that $(w_1, w_2, w_3) = (0, 0, 0)$ corresponds to $(y_1, y_2, y_3) = (0, 0, \frac{8}{5})$ since this is the vertex from which we found the new coordinates. Recall that the translation from x-coordinates to y-coordinates used $x_2 = y_2$, $x_3 = y_3$; thus we have $x_2 = 0$ and $x_3 = \frac{8}{5}$. Then $x_1 = \frac{1}{2}(2 - y_1 - y_2 - y_3) = \frac{1}{2}(\frac{2}{5}) = \frac{1}{5}$, so we have that the optimal vertex is $(x_1, x_2, x_3) = (\frac{1}{5}, 0, \frac{8}{5})$.

2 Dynamic programming

Let w be the remaining weight on the plane, $f(w)$ is the max value of the goods the plane can carry with the remaining weight w. From the table we have $\forall a, b \in^{+}$, $f(a+b) \geq f(a) + f(b)$. So for the state transition, we have

$$f(w) = \max_{\substack{a,b \leq w, \\ a,b \in^{+}}} \{f(a) + f(b), V(w)\} \tag{1}$$

where $V(w) = V_i$ if $\exists i \in^{+}$ s.t. $w = W_i$, otherwise $V(w) = 0$.
Now we can compute from $f(1)$ to $f(13)$,

$$\begin{aligned}
f(1) &= 0.5 \\
f(2) &= 1 \\
f(3) &= \max\{f(1) + f(2), V_4\} = \max\{1.5, 2\} = 2 \\
f(4) &= \max\{f(1) + f(3), f(2) + f(2), V_3\} = \max\{2.5, 2, 3\} = 3 \\
f(5) &= \max\{f(1) + f(4), f(2) + f(3), V_2\} = \max\{3.5, 3, 4\} = 4 \\
f(6) &= \max\{f(1) + f(5), f(2) + f(4), f(3) + f(3)\} = \max\{4.5, 4, 4\} = 4.5 \\
f(7) &= \max\{f(1) + f(6), f(2) + f(5), f(3) + f(4), V_1\} = \max\{5, 5, 5, 9\} = 9 \\
f(8) &= \max\{f(1) + f(7), f(2) + f(6), f(3) + f(5), f(4) + f(4)\} = \max\{9.5, \cdots\} = 9.5 \\
f(9) &= \max\{f(1) + f(8), f(2) + f(7), \cdots\} = \max\{10, 10, \cdots\} = 10 \\
f(10) &= \max\{f(1) + f(9), f(2) + f(8), f(3) + f(7), \cdots\} = \max\{10.5, 10.5, 11, \cdots\} = 11 \\
f(11) &= \max\{f(1) + f(10), f(2) + f(9), f(3) + f(8), f(4) + f(7), \cdots\} = \max\{11.5, 11, 11.5, 12, \cdots\} = 12 \\
f(12) &= \max\{f(1) + f(11), f(2) + f(10), f(3) + f(9), f(4) + f(8), f(5) + f(7), \cdots\} \\
&= \max\{12.5, 12, 12, 12.5, 13, \cdots\} = 13 \\
f(13) &= \max\{f(1) + f(12), f(2) + f(10), f(3) + f(10), f(4) + f(9), f(5) + f(8), f(6) + f(7)\} \\
&= \max\{13.5, 13, 13, 13, 13.5, 13.5\} = 13.5
\end{aligned}$$

And the maximum value is 13.5, and from the process, we have

$$\begin{aligned}
13 &= 1 + 12 = 1 + 5 + 7 \\
13 &= 5 + 8 = 5 + 1 + 7 \\
13 &= 6 + 7 = 1 + 5 + 7
\end{aligned}$$

So the plane should load the product 1, 2, and 5 for 1 unit respectively.

3 Max flow/min cut

We begin by choosing the $s-t$ path $s \to A \to C \to t$ with weight 12, limited by the maximum capacity on (A, C). This path and its corresponding residual graph are shown below; on the flow graph, the numbers correspond to flow on each edge. On the residual graph, the green arrows are the reversed edges yielded by the flow, and numbers are color-coded for the edges corresponding to them.

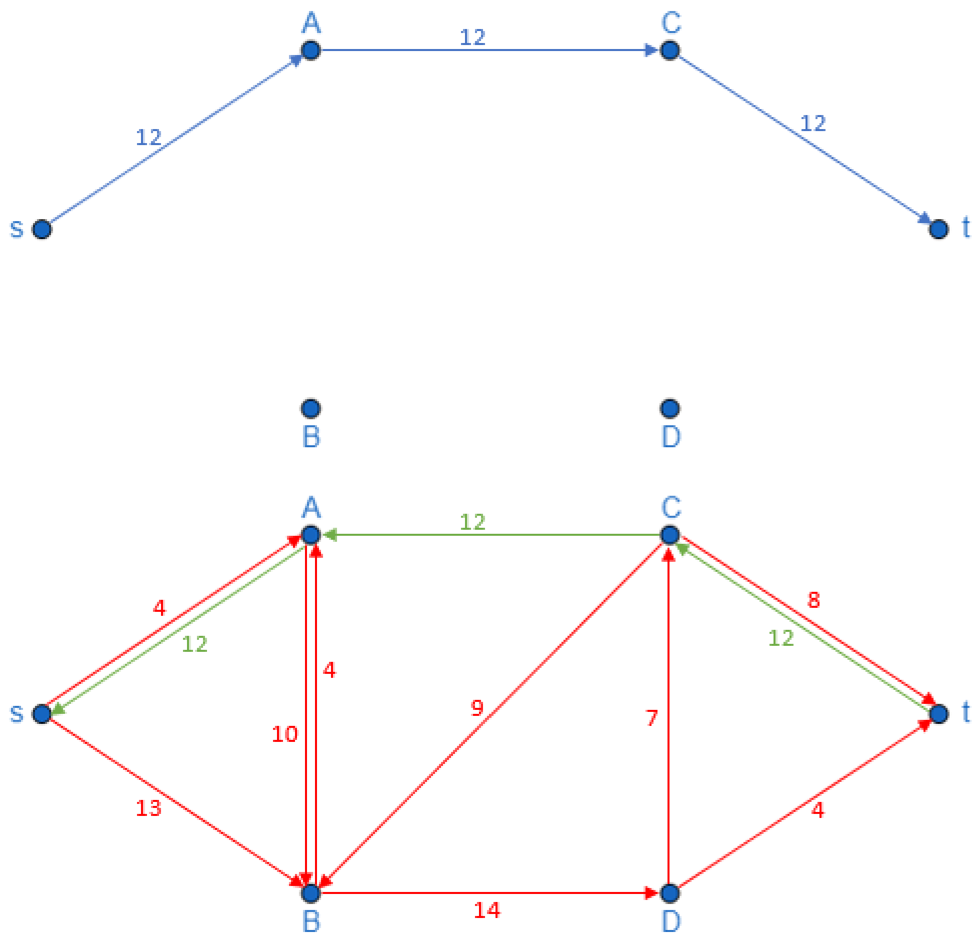

Figure 1: First $s-t$ path and residual

We next choose the path $s \to B \to D \to C \to t$ with weight 7, since this is the option that adds the most to our existing flow. Augmenting the flow with this path yields the following flow network and corresponding residual graph.

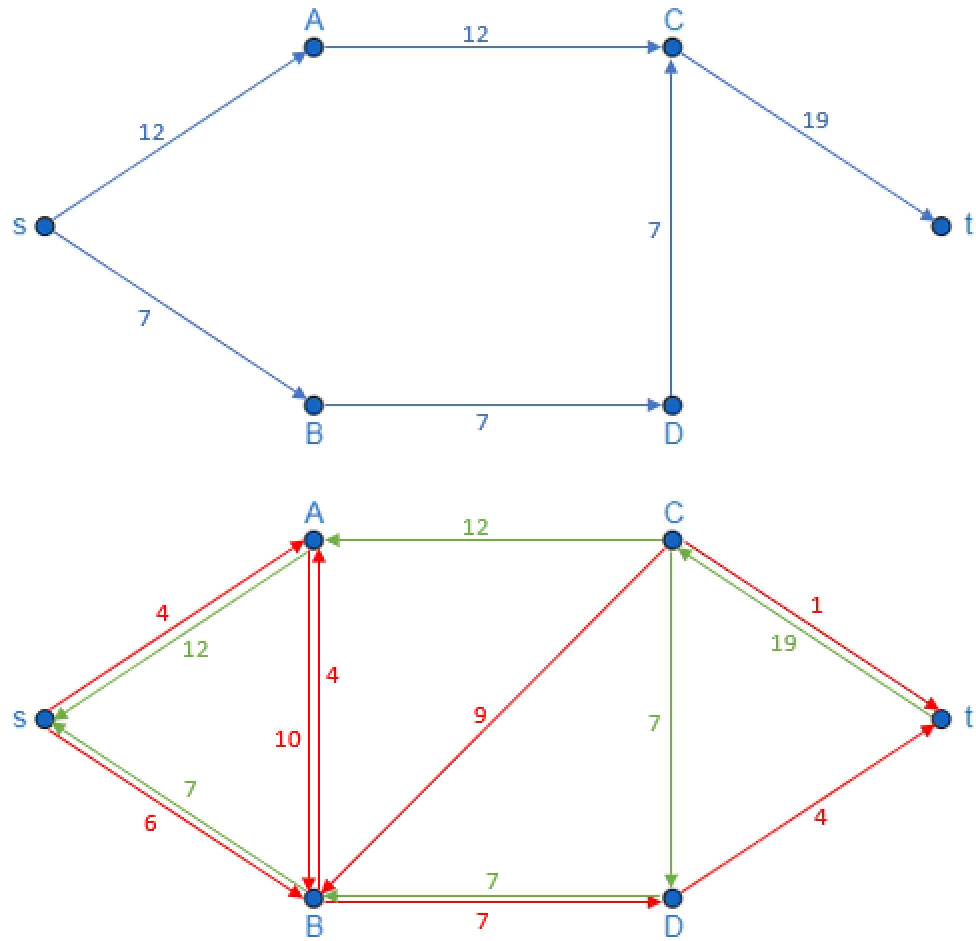

Figure 2: Augmented graph for second $s - t$ path, with residual graph

Finally we choose the path $s \to B \to D \to t$ with weight 4. Notice that after augmenting our flow network with this path, the residual graph contains no path from s to t, meaning that this is the maximum possible flow (a total of 23 units of flow). This solution is not unique; there are several ways to run 23 units of flow through this network. However, in all viable solutions, the edges $(A, C), (D, C), (D, t)$ will be at capacity.

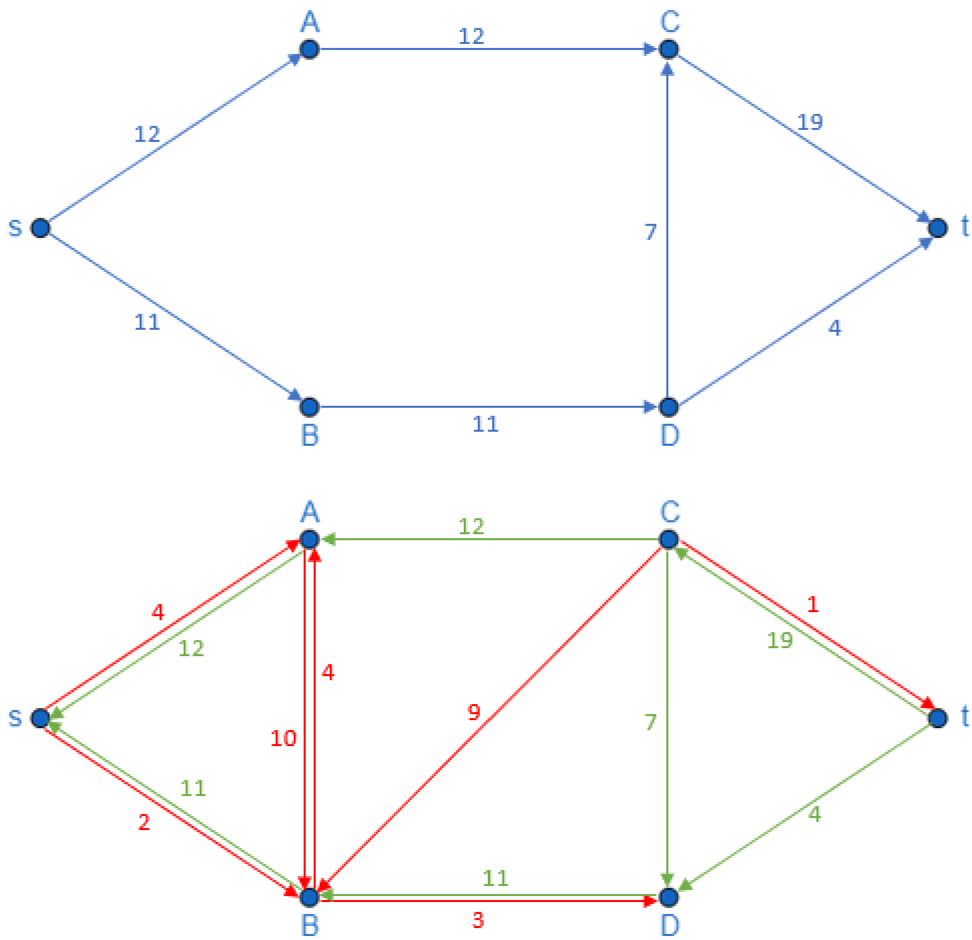

Figure 3: Augmented graph for third $s-t$ path, with final residual graph

Having established that the maximum flow is 23, it follows that the minimum-weight cut is also 23. The corresponding cut is shown below, separating the network into the vertex sets $\{s, A, B, D\}, \{C, t\}$. Note that although the edge (C, B) crosses this cut, it is passing from the set containing t to the set containing s, which is the wrong direction. The only edges whose weights are counted for the cut are those that pass from the set containing s to the set containing s; in this case, these edges are $(A, C), (D, C), (D, t)$, whose weights sum to 23 and which are precisely the edges which are at capacity in any maximum flow.

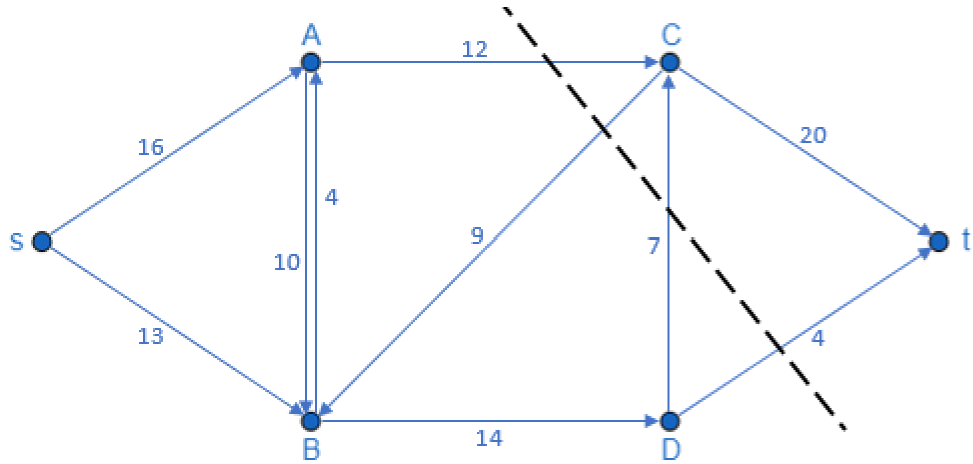

Figure 4: Minimum-weight cut indicated by dashed black line

4 Network simplex method

In the network below, the edges drawn in orange form a spanning tree of the graph. All edges are labelled with their costs, with blue corresponding to edges outside the initial spanning tree and orange corresponding to edges belonging to the initial spanning tree. Starting from this tree, use the network simplex algorithm to find the minimum cost spanning tree.

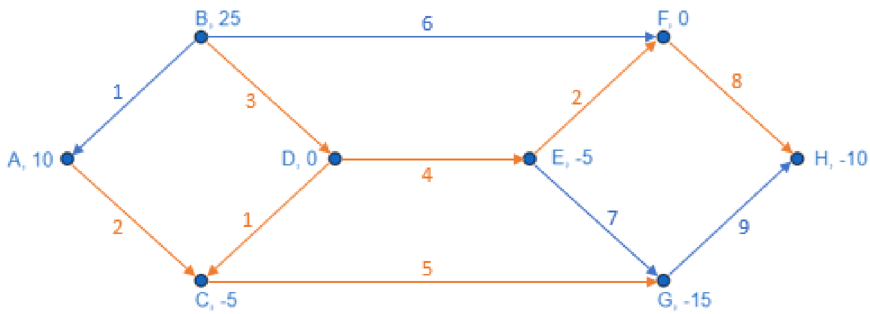

Figure 5: Network for Simplex question

We begin by determining the weights needed on each edge of the original spanning tree. In the figure below, on each of the orange edges, the first number corresponds to the cost and the second number corresponds to the weight to balance the inputs and outputs for each vertex. Note that the dashed blue edges are not in the spanning tree and automatically receive a weight of 0 (unwritten).

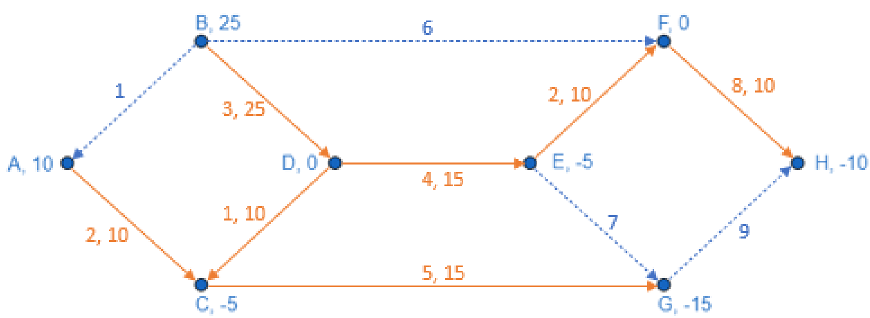

Figure 6: Initial tree with edge weights

We next compute the reduced cost for each edge that is not in the spanning tree. Note that these computations do NOT include edge weights; they are solely based on the cost per unit for each edge.

Edge	Same direction	Opposite direction	Reduced cost
(B, A)	$(B, A), (A, C)$	$(D, C), (B, D)$	1 + 2 - 1 - 3 = -1
(B, F)	(B, F)	$(E, F), (D, E), (B, D)$	6 - 2 - 4 - 3 = -3
(E, G)	$(E, G), (D, E)$	$(C, G), (D, C)$	7 + 4 - 5 - 1 = 5
(G, H)	$(G, H), (B, D), (D, C), (C, G)$	$(F, H), (B, F)$	9 + 3 + 1 + 5 - 8 - 2 = 8

We choose to add edge (B, F) because its reduced cost is the most negative. The smallest edge weight in its corresponding cycle is on (E, F) with weight 10, so we give (B, F) weight 10 and adjust accordingly to get the following, where the new spanning tree and its costs and edge weights are shown in orange.

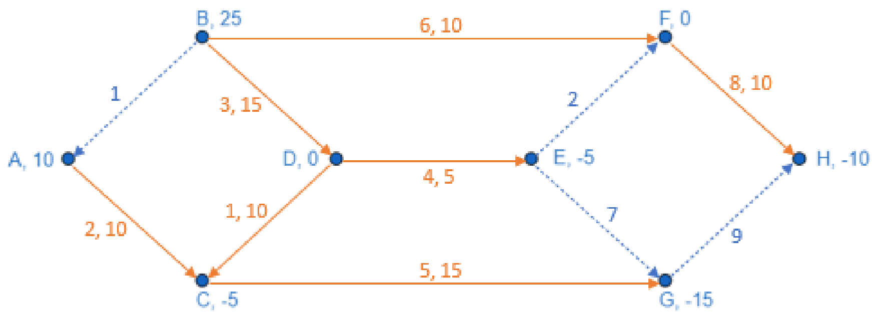

Figure 7: Second spanning tree with edge weights

We now compute the reduced costs for this network:

Edge	Same direction	Opposite direction	Reduced cost
(B, A)	$(B, A), (A, C)$	$(D, C), (B, D)$	1 + 2 - 1 - 3 = -1
(E, F)	$(E, F), (B, D), (D, E)$	(B, F)	2 + 3 + 4 - 6 = 3
(E, G)	$(E, G), (D, E)$	$(C, G), (D, C)$	7 + 4 - 5 - 1 = 5
(G, H)	$(G, H), (D, C), (C, G)$	$(F, H), (E, F), (D, E)$	9 + 1 + 5 - 8 - 2 - 4 = 1

There is only one edge, (B, A), with a negative reduced cost, so we will add it to our spanning tree. The lowest-weight edge in the opposite direction in its corresponding cycle is (D, C) with weight 10, so we add (B, A) with weight 10 and adjust the other edges in the cycle accordingly to result in the following network.

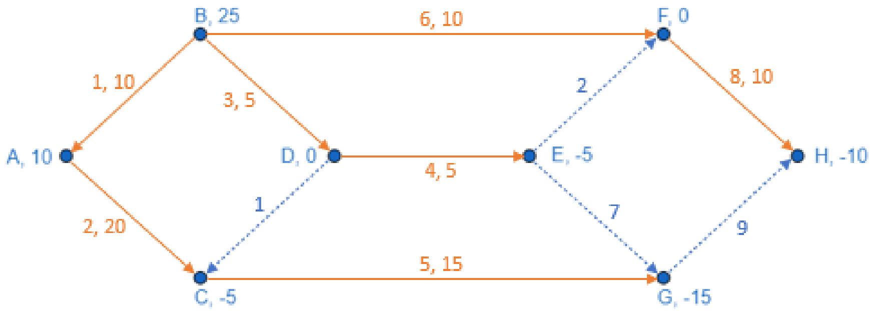

Figure 8: Third spanning tree with edge weights

Evaluating the reduced costs on this updated spanning tree yields the following results:

Edge	Same direction	Opposite direction	Reduced cost
(D, C)	$(D, C), (B, D)$	$(A, C), (B, A)$	1 + 3 - 1 - 2 = 1
(E, F)	$(E, F), (B, D), (D, E)$	(B, F)	2 + 3 + 4 - 6 = 3
(E, G)	$(E, G), (B, D), (D, E)$	$(C, G), (A, C), (B, A)$	7 + 3 + 4 - 5 - 2 - 1 = 6
(G, H)	$(G, H), (B, A), (A, C), (C, G)$	$(F, H), (B, F)$	9 + 1 + 2 + 5 - 8 - 6 = 3

All reduced costs for the edges outside this tree are positive, so this third tree is the minimum-cost spanning tree.

5 Integer programming

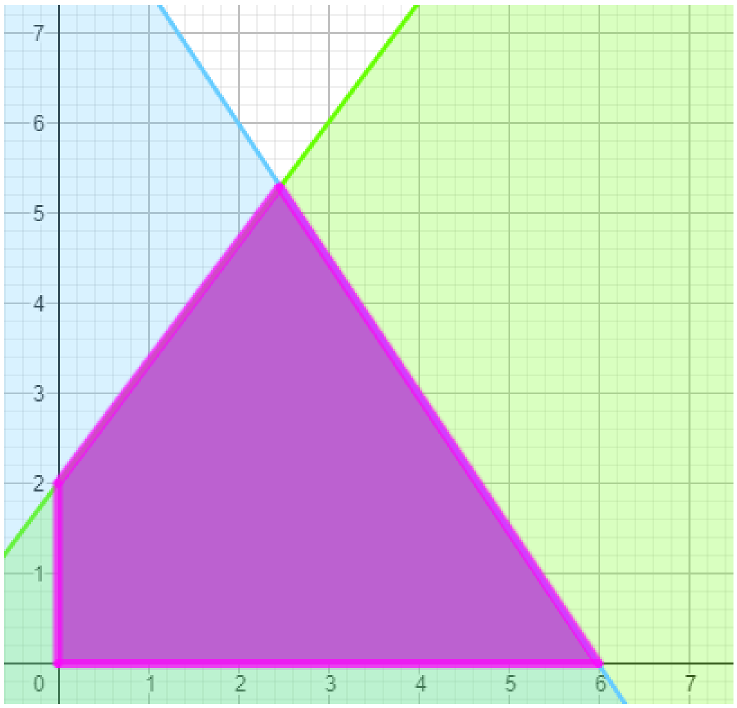

Figure 9: LP relaxation of program

Here is the graph for the LP relaxation. The optimal solution for the LP is $(\frac{42}{17}, \frac{90}{17})$. So we can branch into $x_1 \leq 2$ or $x_1 \geq 3$.
For $x_1 \leq 2$, solving the LP relaxation gives optimal solution $(2, \frac{14}{3})$. Then we further branch it into $x_2 \leq 4$ as $x_2 \geq 5$ is not in the feasible region of LP relaxation. Solving this LP gives the optimal solution $(2, 4)$.
For $x_1 \geq 3$, solving the LP relaxation gives optimal solution $(3, \frac{9}{2})$. Then we further branch it into $x_2 \leq 4$ as $x_2 \geq 5$ is not in the feasible region of LP relaxation. Solving this LP gives the optimal solution $(\frac{10}{3}, 4)$. Branch again, we have $x_1 \leq 3$ or $x_1 \geq 3$, which gives $(3, 4)$ and $(4, 3)$ respectively. Hence there are three potential optimal solutions: $(2, 4)$, $(3, 4)$, and $(4, 3)$. The corresponding values are 22, 23, and 19. So the optimal solution is $(3, 4)$ with value 23.

6 Nonlinear programming

Assume the quantity of product A and B are q_1 and q_2. The cost for producing their prototype C is $\frac{1}{2}(q_1 + q_2)^2$. Then we can formulate the NLP as

$$\begin{aligned} \max \quad & z = 10q_1 + 16q_2 - q_1 - \frac{1}{2}q_2^2 - \frac{1}{2}(q_1 + q_2)^2 \\ \text{s.t.} \quad & q_1 + q_2 \leq 20 \\ & q_1, q_2 \geq 0 \end{aligned}$$

The Jacobian of z is

$$J = \begin{pmatrix} zq_1 \\ zq_2 \end{pmatrix} = \begin{pmatrix} -q_1 - q_2 + 9 \\ -q_1 - 2q_2 + 16 \end{pmatrix}$$

And the Hessian of z is

$$H = \begin{pmatrix} \frac{\partial^2 z}{\partial q_1^2} & \frac{\partial^2 z}{\partial q_1 \partial q_2} \\ \frac{\partial^2 z}{\partial q_2 \partial q_1} & \frac{\partial^2 z}{\partial q_2^2} \end{pmatrix} = \begin{pmatrix} -1 & -1 \\ -1 & -2 \end{pmatrix}$$

And the eigenvalue of H are $\frac{-3 \pm \sqrt{5}}{2} < 0$. So the Hessian is negative definite, which means z takes maximum when $J = 0$. Solving this we have $q_1 = 2$, $q_2 = 7$. And $q_1 + q_2 = 9 < 20$, so the optimal solution is in the feasible region.

7 Convexity

Note that by the definition of $h(x)$, we have $f(x) \leq h(x)$ and $g(x) \leq h(x)$ for any x. In particular, we also have that if there is some quantity c such that $f(x) \leq c$ and $g(x) \leq c$, it must also be true that $h(x) \leq c$. Consider any two points x, y and some $\lambda \in [0,1]$. We want to show that $h(\lambda x+(1-\lambda)y) \leq \lambda h(x)+(1-\lambda)h(y)$; by the previous statements, it suffices to show that $f(\lambda x+(1-\lambda)y)$ and $g(\lambda x + (1-\lambda)y)$ are smaller than this quantity.

Since $f(x)$ and $g(x)$ are convex, we know:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$$
$$g(\lambda x + (1-\lambda)y) \leq \lambda g(x) + (1-\lambda)g(y)$$

But by the definition of $h(x)$ we know that $f(x) \leq h(x)$ and $f(y) \leq h(y)$; moreover, from this same definition we also know $g(x) \leq h(x)$ and $g(y) \leq h(y)$. Therefore we have:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y) \leq \lambda h(x) + (1-\lambda)h(y)$$
$$g(\lambda x + (1-\lambda)y) \leq \lambda g(x) + (1-\lambda)g(y) \leq \lambda h(x) + (1-\lambda)h(y)$$

Thus both $f(\lambda x + (1-\lambda)y)$ and $g(\lambda x + (1-\lambda)y)$ are lesser or equal to $\lambda h(x) + (1-\lambda)h(y)$, and since $h(x) = \max\{f(x), g(x)\}$, it follows by our earlier reasoning that $h(\lambda x+(1-\lambda)y) \leq \lambda h(x)+(1-\lambda)h(y)$. Therefore $h(x)$ is convex.