

CSE 101: Homework 5 Solutions

1. 7.1: Here the feasible region is defined by the points interior to the convex hull of $(0, 0)$, $(5, 0)$, $(5, 2)$, and $(2, 5)$. Of these vertices $(5, 2)$ gives the largest objective value (31), meaning $x = 5$ and $y = 2$ is the optimal solution.
2. 7.5(a): Let's assume that we can sell everything that we manufacture, so if x_1 is the number of Frisky Pup dog food packages we make and x_2 is the number of Husky Hound packages, our profit can be calculated by subtracting cost of materials out of the sale price. This gives $(7 - 1(1) - 1.5(2) - 1.40)x_1 + (6 - 1(2) - 2(1) - .60)x_2 = 1.60x_1 + 1.40x_2$. Since we have only 240,000 lbs of cereal and we need 1 lbs of cereal for each Frisky Pup and 2 for each Husky Hound, we would need $x_1 + 2x_2 \leq 240000$. Similarly for meat we need $1.5x_1 + x_2 \leq 180000$. The final constraint is that only 110,000 Frisky Pups can be made by the factors meaning $x_1 \leq 110,000$. So the linear program is,

$$\begin{aligned} \max_{x_1, x_2} \quad & 1.6x_1 + 1.4x_2 \\ & x_1 + 2x_2 \leq 240000 \\ & 1.5x_1 + x_2 \leq 180000 \\ & x_1 \leq 110000 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{aligned}$$

3. 7.5(b): The feasible region is define by the convex hull of the vertices: $(0, 0)$, $(110k, 0)$, $(0, 120k)$, $(110k, 15k)$, and $(60k, 90k)$. The combination of husky and frisky that maximize profit is $x_1 = 60,000$ and $x_2 = 90,000$ with a profit of \$222,000.
4. 7.18(a): Let s_1, \dots, s_n be the sources and t_1, \dots, t_m the sinks. Create a new master source node S and a master sink node T . Connect S to each s_i by an edge with infinite capacity. Similarly connect each t_i to T with an edge of infinite capacity. Running the traditional max-flow algorithm on this slightly augmented graph will find the desired result.
5. 7.18(b): Let c_v be the capacity on the maximum flow that can enter vertex v . We will split v into two vertices v_1 and v_2 connected by a single edge.

All of the incoming edges that were coming into v will go into v_1 and all the outgoing edges from v in the original graph will go out of v_2 . Now, the edge (v_1, v_2) will have capacity c_v . This will simulate the regulation of flow through v . Doing this split into 2 vertices for each v in the graph, and running max-flow on this graph will give the desired result.

6. 7.18(c): Let l_e be the lower bound on the flow for edge e . In the original max-flow LP there are constraints that say that the flow sent along edge e , f_e , must be between 0 and c_e , the capacity for edge e . We simply change this constraint to $l_e \leq f_e \leq c_e$. We add this constraint for every edge e .
7. 7.18(d): We know that there are conservation of flow constraints in the max-flow LP. We simply alter these constraints to reflect this loss coefficient. The conservation of flow constraint for a node u in the original max-flow LP looked like

$$\sum_{e=(x,u)} f_e = \sum_{e=(u,x)} f_e$$

We incorporate the discount factor like so

$$(1 - \epsilon_u) \sum_{e=(x,u)} f_e = \sum_{e=(u,x)} f_e$$

Adding this equality constraint for every node u in the graph will do the trick.

8. Let us be given a potential flow that we want to verify whether it is maximal. We can first make sure each flow is no larger than the capacity of each edge, and that each flow on an edge is at least 0. If any of these simple constraints are not met, then we can return that it is not the maximum flow.

We also know a maximum flow will have no $s - t$ path in the residual graph, G^f , on which flow can be increased. So, we explicitly construct G^f based on the flow in question (per the definition on pg. 214). We will then run BFS to see if we can find an $s - t$ path in G^f where all edges on the path have capacity strictly larger than 0. As we examine an edge (u, v) in *BFS* we have an auxiliary array that marks v as potentially part of such an $s - t$ path if (u, v) has capacity > 0 and u was also marked as true in this auxiliary array. We initially make the entry corresponding to s true in this array, and if t has a true entry in it after this BFS process finishes, then we know that an $s-t$ path exists where more flow could be sent from s to t . This would mean that the given flow is sub-optimal. If no such $s - t$ path can be found, then the flow is maximal. The running time of this algorithm is the same as BFS which is $O(|V| + |E|)$. The correctness of this algorithm hinges on the fact that no $s - t$ path in the residual graph with all edges having capacity greater than 0 exists for a maximum flow, which is shown in the book on pages 214 and 215.