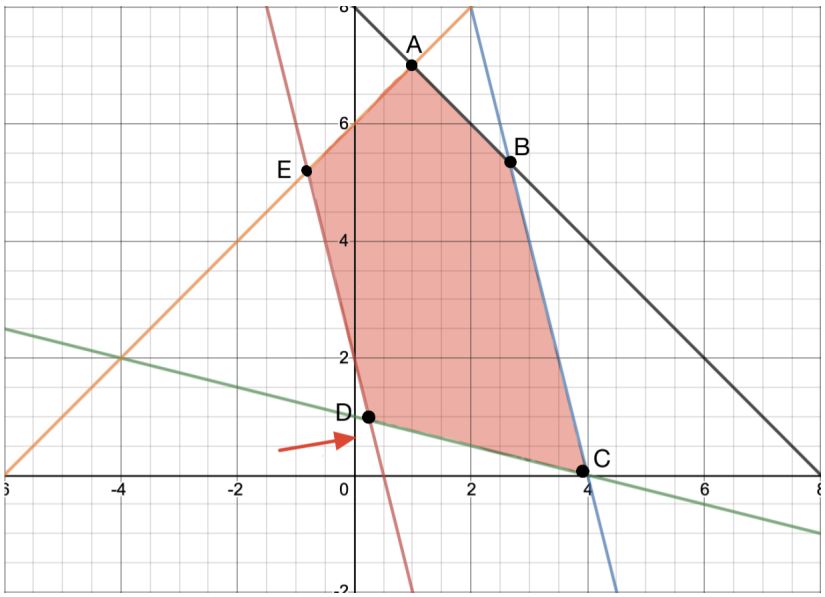
1 Algorithms for Solving Linear Programming

In lecture, we went through two algorithms for solving linear programming programs - vertex enumeration and the hill climbing algorithm.

Consider this linear programming problem. The goal is to minimize the cost, and the cost vector (red) is perpendicular to the blue and red lines.



1. Briefly describe both algorithms and explain how they differ. (hint: use terms such as vertices, intersections and neighbors).

   Vertex enumeration: Find all vertices of feasible region (feasible intersections), check objective value.

   Hill climbing: Start with an arbitrary vertex. Iteratively move to a best neighboring vertex until no better neighbor is found.

   Intersection is found by solving a pair of constraints (although some constraint pairs might not intersect). A neighboring intersection differs by only one constraint.

2. Run the hill climbing algorithm starting from point B. Now try running the algorithm starting from point C. How do their solutions differ?

   Starting from point B returns a solution of point E, while starting from point C returns a solution of point D. Notice that points E and D are equally optimal.

   Starting from point B, we have neighboring vertices point A and C. Point A is chosen as it is better. From point A, we have neighboring vertices point E and B. Point B is worse than point A, and point E is better point A. From point E, we have neighboring vertices point A and D. Point A is worse than point E, and point D is equally optimal. Thus, no better neighbor is found and the algorithm returns point E.

Starting from point C, we have neighboring vertices point B and D. Point D is chosen as it is better. From point D, the neighboring vertices A and C are either worse or equal to point D. Thus, no better neighbor is found and the algorithm returns point D.

2 Cargo Plane: Linear Programming Formulation

A cargo plane has three compartments for storing cargo: front, center and rear. These compartments have the following limits on both weight and space:

Compartment	**Weight capacity (tons)**	**Space capacity (cubic metres)**
Front	10	6800
Centre	16	8700
Rear	8	5300

The following four cargoes are available for shipment on the next flight:

Cargo	**Weight (tons)**	**Volume (cubic metres/ton)**	**Profit ($/ton)**
C1	18	480	310
C2	15	650	380
C3	23	580	350
C4	12	390	285

Any proportion of these cargos can be accepted. The objective is to determine how much of each cargo C1, C2, C3 and C4 should be accepted and how to distribute each among the compartments so that the total profit for the flight is maximised. **Formulate** the above problem as a linear program (what is the objective and the constraints?). Think about the assumptions you are making when formulating this problem as a linear program.

Variables:

We need to decide how much of each of the four cargos to put in each of the three compartments. Hence let $x_{i,j}$ be the number of tonnes of cargo i (i=1,2,3,4 for C1, C2, C3 and C4 respectively) that is put into compartment j (j=1 for Front, j=2 for Center and j=3 for Rear) where $x_{i,j} \geq 0; i = 1, 2, 3, 4; j = 1, 2, 3$.

(Note here that we are explicitly told we can split the cargos into any proportions (fractions) that we like.)

Constraints:

1. We cannot pack more of each of the four cargos than we have available.

$$x_{1,1} + x_{1,2} + x_{1,3} \leq 18$$

$$x_{2,1} + x_{2,2} + x_{2,3} \leq 15$$

$$x_{3,1} + x_{3,2} + x_{3,3} \leq 23$$

$$x_{4,1} + x_{4,2} + x_{4,3} \leq 12$$

2. The weight capacity of each compartment must be respected.

$$x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} \leq 10$$

$$x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} \leq 16$$

$$x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} \leq 8$$

3. The volume (space) capacity of each compartment must be respected.

$$480x_{1,1} + 650x_{2,1} + 580x_{3,1} + 390x_{4,1} \leq 6800$$

$$480x_{1,2} + 650x_{2,2} + 580x_{3,2} + 390x_{4,2} \leq 8700$$

$$480x_{1,3} + 650x_{2,3} + 580x_{3,3} + 390x_{4,3} \leq 5300$$

Objective: The objective is to maximise total profit, i.e.

maximise $310(x_{1,1} + x_{1,2} + x_{1,3}) + 380(x_{2,1} + x_{2,2} + x_{2,3}) + 350(x_{3,1} + x_{3,2} + x_{3,3}) + 285(x_{4,1} + x_{4,2} + x_{4,3})$

The basic assumptions are:

1. that each cargo can be split into whatever proportions/fractions we desire
2. that each cargo can be split between two or more compartments if we so desire
3. that the cargo can be packed into each compartment (for example if the cargo was spherical it would not be possible to pack a compartment to volume capacity, some free space is inevitable in sphere packing)
4. all the data/numbers given are accurate

Something also to note is that we can also solve this linear programming problem using one of the various tools available online. One in particular you may find interesting is Google's OR-tools (Click here!). If you want to see an example of this being used, we have written up a solution to this recitation problem, which can be found at this link: Linear Programming code. We do not require that you learn how to use these tools, but it is a cool resource if you want to check it out! On all homeworks and exams, you will be expected to solve it by hand if we ask you to.

If you were to put this linear program into standard form, what would be the dimensions of $A, \mathbf{b}, \mathbf{c}, \mathbf{x}$?

There are 12 variables, so $\dim \mathbf{x} = 12 \times 1$ and $\dim \mathbf{c} = 12 \times 1$. There are 10 constraints, so $\dim \mathbf{b} = 10 \times 1$ and since each constraint involves 12 variables, $\dim A = 10 \times 12$. Note that while A has 120 elements, most of them will be 0 as each constraint only involves a few of the variables, making it a sparse matrix.

Now consider a simpler problem. There is a cargo plane with a single compartment with limit on 20 tons weight and 2400 cubic meters limit on space. You want to use this cargo plane to transport boxes of oranges and pineapples to sell in a market overseas.

Your goal is to maximize the number of gold pieces under following constraints:

- The market only allows each person to sell 14 boxes.
- 1 box of oranges has weight 1 ton and volume of 100 cubic meters.
- 1 box of pineapples has weight 2 tons and volume of 300 cubic meters.
- You earn 5 gold pieces for 1 box of oranges.
- You earn 12 gold pieces for 1 box of pineapples.

We will now formulate and solve the LP.

1. Write the LP in inequality form.
2. Graph the constraints, cost vector, and at least 3 cost contours. Indicate the feasible region.
3. What is the **optimal number** of boxes of oranges and pineapples? How much gold does this earn?

Formulating problem as linear programming formulation, we have have following cost function and constraints where box of orange is x_1 and box of pineapples is x_2.

$$\begin{gathered}\text{Minimize } -5x_1 - 12x_2 \text{ where} \\ -x_1 \leq 0 \\ -x_2 \leq 0 \\ x_1 + x_2 \leq 14 \\ x_1 + 2x_2 \leq 20 \\ 100x_1 + 300x_2 \leq 2400\end{gathered}$$

Putting this problem in inequality form we have:

$$\min_{\mathbf{x}} \mathbf{c}^\top \mathbf{x} \; s.t. \; A\mathbf{x} \leq \mathbf{b}$$

where

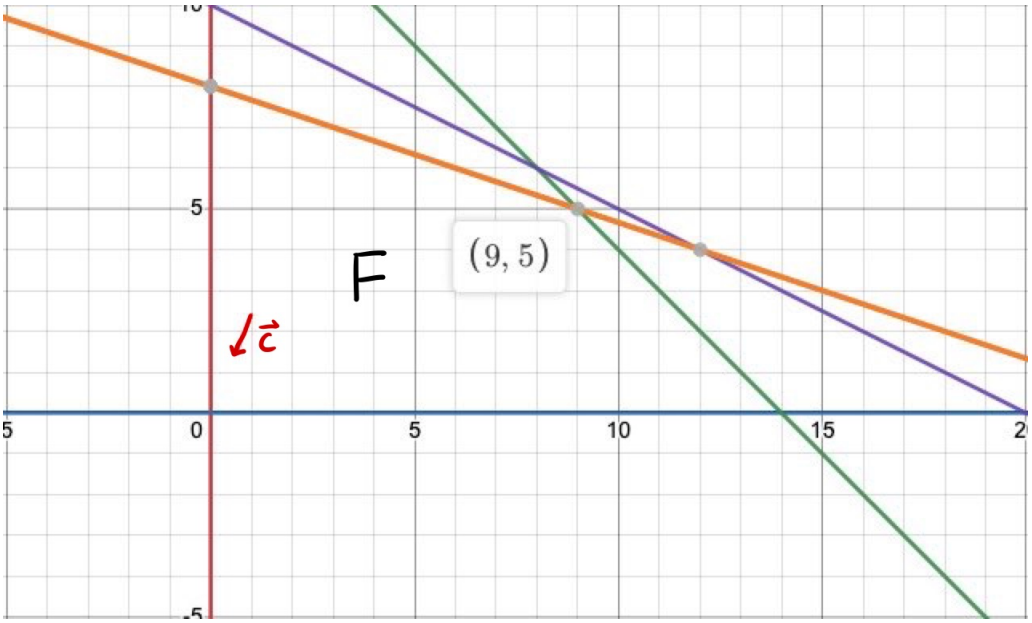
$$\mathbf{c} = [-5 \;\; -12]^\top$$

$$\mathbf{x} = [x_1 \; x_2]^\top$$

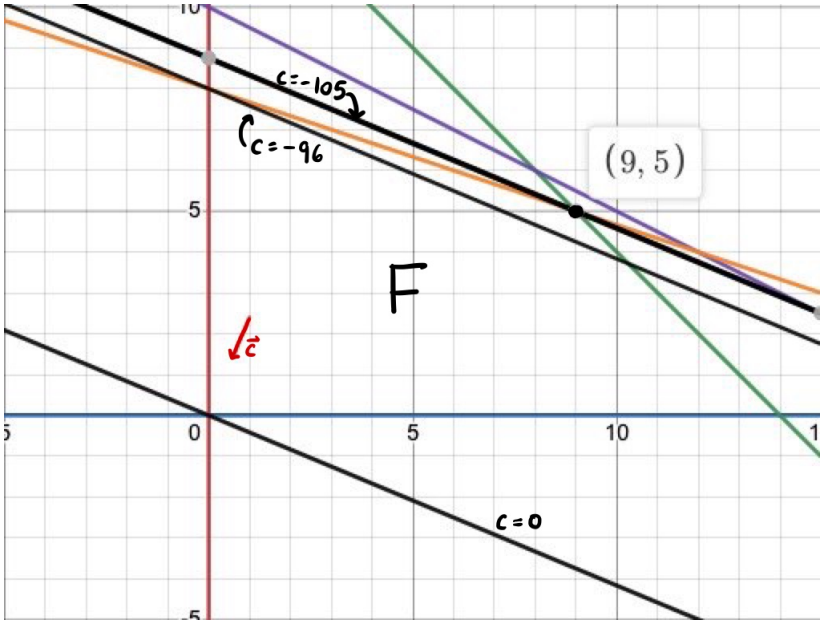
$$\mathbf{b} = [0 \; 0 \; 14 \; 20 \; 2400]$$

$$A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 1 \\ 1 & 2 \\ 100 & 300 \end{bmatrix}$$

We can solve these constraints by either drawing a graph and find intersection of constraints, or through systems of equations. Below is the graph of constraints with the cost vector labelled. Remember the cost vector points in the direction of increasing cost. (Solution: 9 boxes of oranges, 5 boxes of pineapples, 105 gold pieces)



Below is the graph with the cost contours (in black). Each of the contours represent a line of equal cost (labelled next to it) and are perpendicular to the cost vector. Subsequently, we can also use the contour lines to see that the one with cost = -105 is the last one to intersect the feasible region in the direction away from where the cost vector is pointing (since we want to minimize cost). The intersection point is (9,5) as we saw from the solution in the above image. Remember this cost is in terms of the standard form LP where we converted the maximization to a minimization. So, we must flip the resulting cost to be positive and get a result of 105 gold pieces.


c=-105
c=-96
(9, 5)
F
$\vec{c}$
5
0
5
10
c=0

3 CSP as IP

Alice, Bob, and Charles want to study in Gates, which has 9 floors. To maximize productivity, we need to assign each of them to a separate floor without violating the following constraints:

- Alice only has access to floors 4 through 8
- Bob only has access to floors 3 through 7
- Charles must be at least 2 floors higher than Bob
- Alice must be at least 1 floor higher than Bob
- Alice must be at least 1 floor lower than Charles

Our goal is to assign Alice, Bob, and Charles to the highest possible floors.

1. Formulate the problem as a CSP.

   Variables: X_A, X_B, X_C (where X_A is Alice's floor number, X_B is Bob's floor number, and X_C is Charles's floor number)

   Domains:

   - $D_A \in \{4, 5, 6, 7, 8\}$
   - $D_B \in \{3, 4, 5, 6, 7\}$
   - $D_C \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

   Constraints:

   - $X_C \geq 2 + X_B$ (Charles must be at least 2 floors higher than Bob)
   - $X_A \geq 1 + X_B$ (Alice must be at least 1 floor higher than Bob)
   - $X_A \leq X_C - 1$ (Alice must be at least 1 floor lower than Charles)

2. Formulate the problem as an IP problem.

   Variables: X_A, X_B, X_C (where X_A is Alice's floor number, X_B is Bob's floor number, and X_C is Charles's floor number).

   Goal: We want to find the $max_{X_A,X_B,X_C} X_A + X_B + X_C$. This is equivalent to saying we want to find $min_{X_A,X_B,X_C}(-X_A - X_B - X_C)$

   Constraints:

   - Alice only has access to floors 4 through 8

     $$X_A \geq 4, \text{ which is equivalent to } -X_A \leq -4$$
     $$X_A \leq 8$$

   - Bob only has access to floors 3 through 7

     $$X_B \geq 3, \text{ which is equivalent to } -X_B \leq -3$$
     $$X_B \leq 7$$

- Charles only has access to floors 1 through 9 (Note: this constraint was not explicitly given, but Gates has 9 floors, and we need to make sure that Charles is assigned to a floor between 1 and 9, inclusive)

$$X_C \geq 1, \text{ which is equivalent to } -X_C \leq -1$$
$$X_C \leq 9$$

- Charles must be at least 2 floors higher than Bob

$$X_C \geq 2 + X_B, \text{ which is equivalent to } X_B - X_C \leq -2$$

- Alice must be at least 1 floor higher than Bob

$$X_A \geq 1 + X_B, \text{ which is equivalent to } X_B - X_A \leq -1$$

- Alice must be at least 1 floor lower than Charles

$$X_A \leq X_C - 1, \text{ which is equivalent to } X_A - X_C \leq -1$$

Therefore, we want to find $min_{X_A,X_B,X_C}(-X_A - X_B - X_C)$ such that

$$-X_A \leq -4$$
$$X_A \leq 8$$

$$-X_B \leq -3$$
$$X_B \leq 7$$

$$-X_C \leq -1$$
$$X_C \leq 9$$

$$X_B - X_C \leq -2$$
$$X_B - X_A \leq -1$$
$$X_A - X_C \leq -1$$

4 Baymax's Factory

Baymax and the 281 TAs have opened a factory to produce special medicine and bandages. These are really difficult to produce and require the collaboration of robots and humans.

To produce an ounce of medicine, it takes 0.2 hours of human labor and 4 hours of robot labor. To produce an inch of bandage, it takes 0.5 hours of human labor and 2 hours of robot labor. An ounce of medicine sells for \$30 and an inch of bandages sells for \$30. Medicine and bandages can be sold in fractions of an ounce or inch.

We want to maximize our profit so we can buy gifts for all the students. However, the TAs are really busy so they can only devote 90 human hours. In addition, Baymax can only devote 800 robot hours because he has other obligations to tend to. How can we maximize our profit?

1. Is this a linear, mixed or integer programming problem? Formulate and solve it.

   It is a linear programming problem, as the medicine and bandages can be sold a fraction of a unit. Let x be the ounces of medicine and y be the inches of bandages produced.

   Objective: Maximize total profit:

   $$\min_{x,y} -30x - 30y$$

   Constraints:

   $$0.2x + 0.5y \leq 90$$
   $$4x + 2y \leq 800$$
   $$x \geq 0, y \geq 0$$

   Putting this problem in inequality form we have:

   $$\min_{\mathbf{x}} \mathbf{c}^\top \mathbf{x} \; s.t. \; A\mathbf{x} \leq \mathbf{b}$$

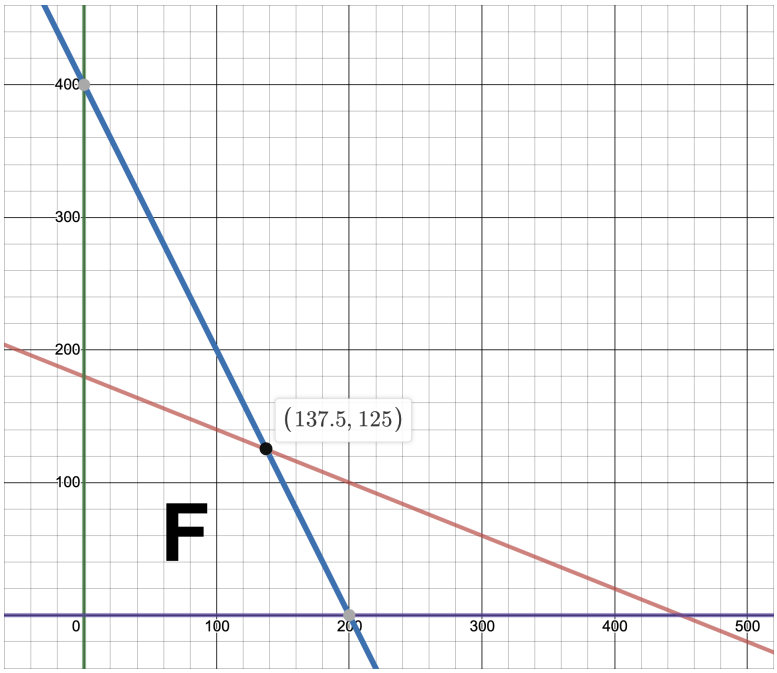
   where

   $$\mathbf{c} = [-30 \;\; -30]^\top$$
   $$\mathbf{x} = [x \; y]^\top$$
   $$\mathbf{b} = [0 \; 0 \; 90 \; 800]$$
   $$A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0.2 & 0.5 \\ 4 & 2 \end{bmatrix}$$

   Given the constraints, we can solve for x and y:

   $$y = 180 - 0.4x$$
   $$y = 400 - 2x$$

   That gives us the following graph:



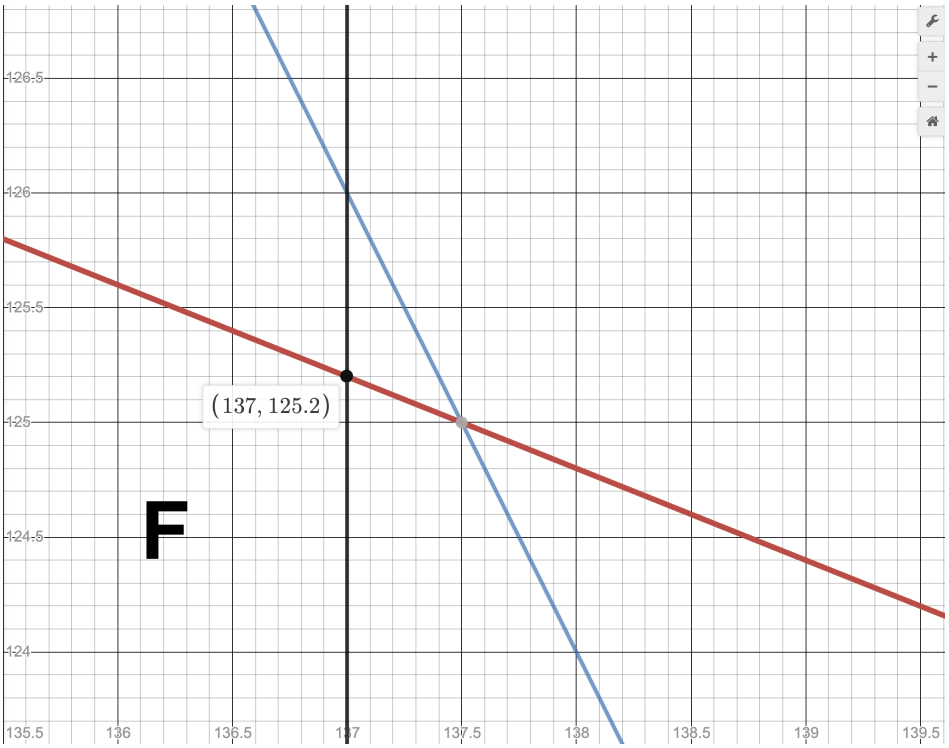
Since we want to maximize profit, we want to choose the furthest point, giving us $x = 137.5$ and $y = 125$.

2. Now suppose the items can only be sold in whole units (by ounce/inch). Is this a linear, mixed, or integer programming problem? Perform branch and bound for one branch level. You do not have to evaluate; writing out the constraints will suffice.

   This is an integer programming problem, and the formulation is identical to part (a). However, the domains of x and y are reduced to integers. We can solve the problem with branch and bound.

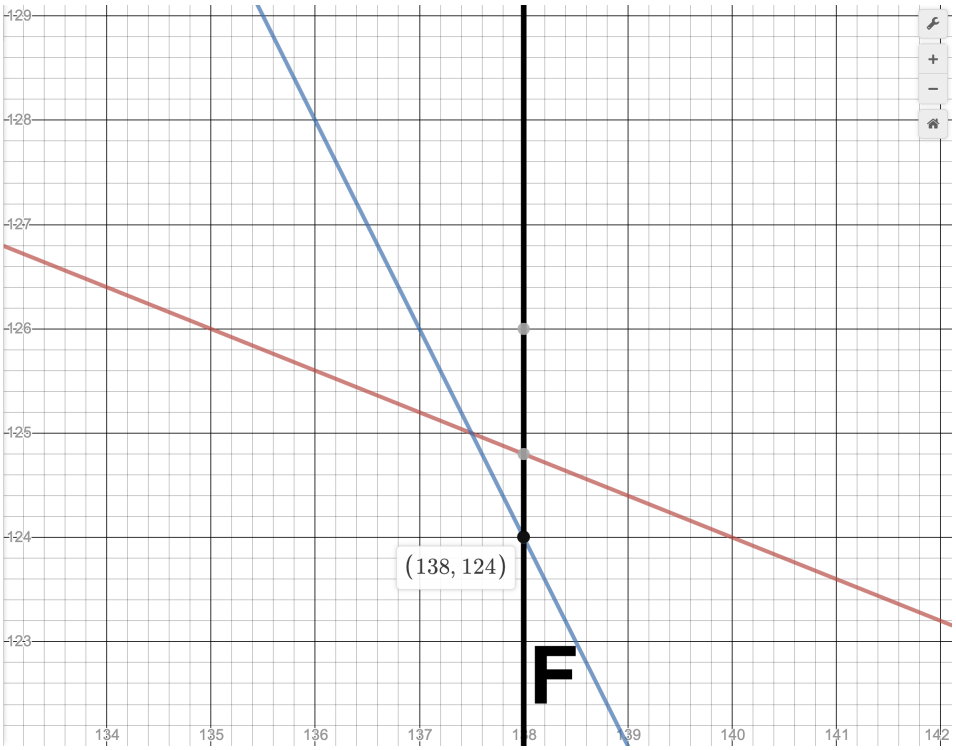
   We first use linear programming to find the optimal point of (137.5, 125), as we did in (1a). Since $x = 137.5$ is not an integer, we branch on it by adding the constraints that $x \leq 137$ or $x \geq 138$.

   Left branch:
   $x \leq 137$
   $0.2x + 0.5y \leq 90$
   $4x + 2y \leq 800$
   $x \geq 0, y \geq 0$



The linear programming solution is (137, 125.2), so we add the left branch to the pq with value -7866. Next we consider the right branch.

Right branch:
$x \geq 138$
$0.2x + 0.5y \leq 90$
$4x + 2y \leq 800$
$x \geq 0, y \geq 0$



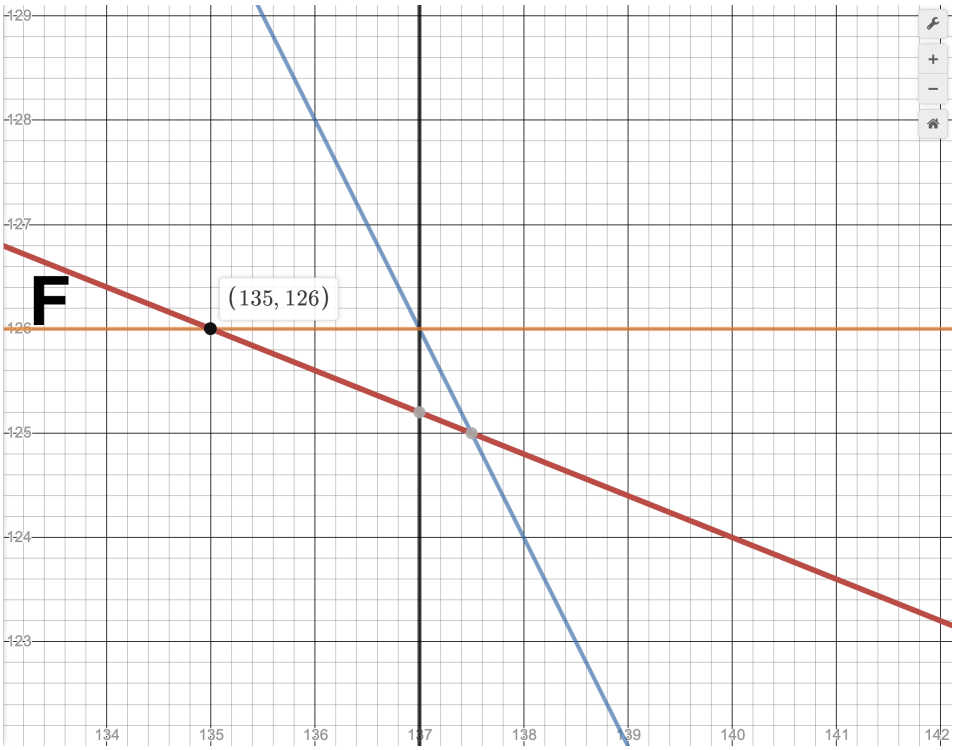
the linear programming solution (138, 124) has a value of -7860 and we similarly add it to the pq.

The left branch has a better value so it is popped of the priority queue first and then since the left branch solution is not an integer value the left-right and left-left branches are considered. Specifically we now need to branch on the y value by adding the constraints that $y \leq 125$ or $y \geq 126$.

Left-Left branch:
$x \leq 137$
$0.2x + 0.5y \leq 90$
$4x + 2y \leq 800$
$x \geq 0, y \geq 0$
$y \leq 125$

The linear programming solution is (137, 125) and it is added to the priority queue along with the value -7860.

Left-Right branch:
$x \leq 137$
$0.2x + 0.5y \leq 90$
$4x + 2y \leq 800$
$x \geq 0, y \geq 0$
$y \geq 126$



The linear programming solution is (135, 126) and it is added to the pq along with the value -7830.

Next on the pq there is a tie between right and left-left branches. Both solutions are integer so the one that is popped of the queue next will be returned as our solution with objective -7860 and point (138, 124) or (137, 125).

3. Now assume medicine can be sold in fractions but bandages can only be sold in whole units. What kind of a programming problem would this be, and how would our evaluation process differ from the problem type in part b?

   This will be a mixed integer linear programming problem. We will evaluate by only branching and bounding on the number of bandages.

4. How many optimal solutions can a LP have? How about IP?

   Both LP and IP can have an infinite number of optimal solutions. Imagine a cost vector that's perpendicular to a constraint boundary. Then, we could have that constraint boundary cross infinitely many integers/real numbers (i.e. the line $x = 0$).

5 4-Queens

Recall the 4-Queens problem. The goal is to place 4 chess queens on a 4x4 chess board such at no two queens are in the same row, column and diagonal.

Formulate the 4-Queens problem as an integer programming problem.

Let our variables be x_{ij} for $0 \leq i \leq 3$, $0 \leq j \leq 3$, representing whether there is a queen in row i, column j. We want to find $\max_x \sum_i \sum_j x_{ij}$ such that $x_{ij} \in \{0, 1\}$.

Check: only one queen in each row - fix i and iterate over each column, ensuring they sum up to ≤ 1.

$$\sum_j x_{ij} = 1 \forall i \in \{0, 3\}$$

Check: only one queen in each column: fix j and iterate over each row, ensuring they sum up to ≤ 1.

$$\sum_i x_{ij} = 1 \forall j \in \{0, 3\}$$

Check: at most one queen in positive-slope diagonals (stretching from top left to bottom right):

$$\sum_{i,j:i+j=k} x_{ij} \leq 1, \forall k \in \{0, 1, 2, ..., 6\}$$

(k=0: (0, 0) | k=1: (0,1), (1,0) | k=2: (0,2), (1,1), (2,0) | k=3 ...)

Check: at most one queen in negative-slope diagonals (stretching from bottom left to top right):

$$\sum_{i,j:i-j=k} x_{ij} \leq 1, \forall k \in \{-3, -2, -1, ..., 3\}$$

Note that the equalities should all be represented as inequalities ≤ 1 and the negation of it ≤ -1. Putting this problem in standard form we have:

$$\min_{\mathbf{x}} 0 \ s.t. \ A\mathbf{x} = \mathbf{b}$$

where

$$\mathbf{x} = [x_{00} \ x_{01} \ x_{02} \ x_{03} \ x_{10} \ x_{11} \ x_{12} \ x_{13} \ x_{20} \ x_{21} \ x_{22} \ x_{23} \ x_{30} \ x_{31} \ x_{32} \ x_{33}]^\top$$

$$\mathbf{b} = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$$

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Where the first 4 rows of A are row restrictions, the next 4 are column restrictions, and the last 7 are diagonal restrictions.