1 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?

(a) 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:

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

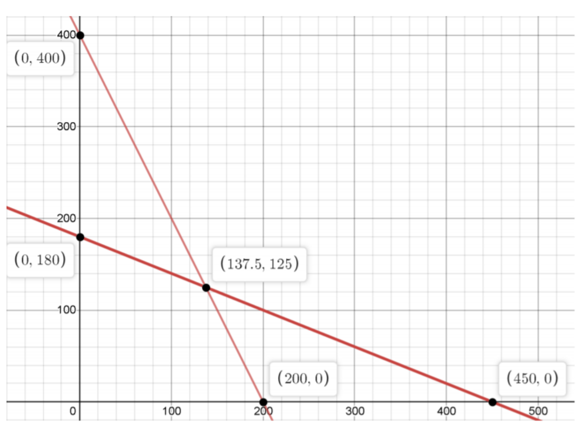
Constraints:

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

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

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

That gives us the following graph:



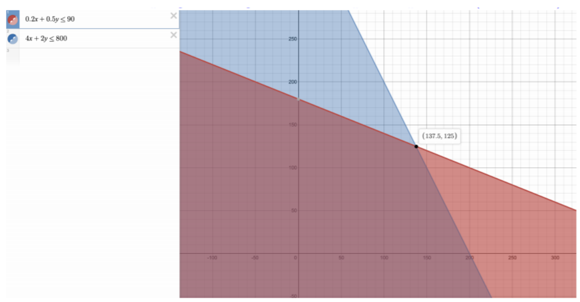
Since we want to maximize profit, we want to choose the furthest point, giving us $x = 137.5$ and $y = 125$.

Now suppose the items can only be sold in whole units (by ounce/inch).

(b) 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.

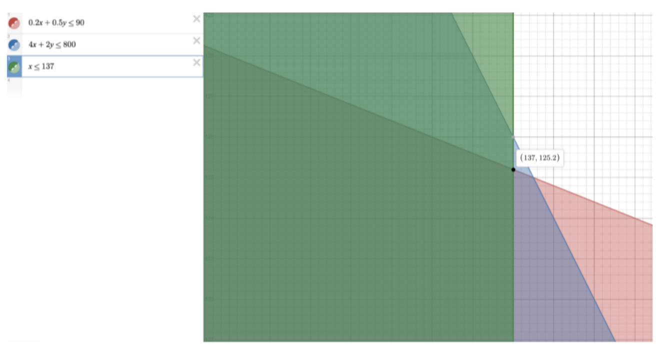
This is an integer programming problem, and the formulation is identical to part (a). However, the domains of x and y are reduced to integers. We can solve the problem with branch and bound.

We first use linear programming to find the optimal point of (137.5, 125).



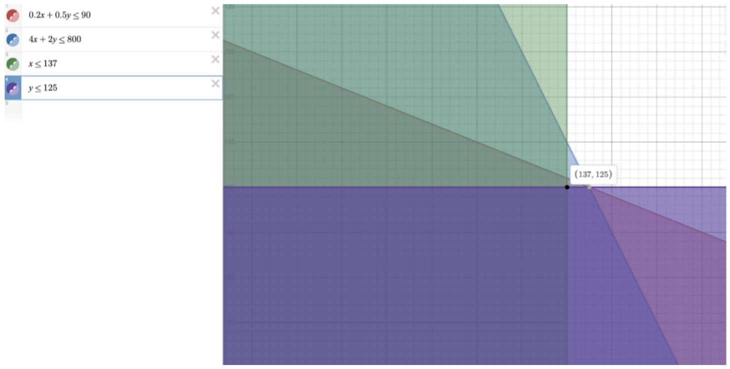
Since $x = 137.5$ is not an integer, we branch on it by adding the constraints that $x \leq 137$ or $x \geq 138$.

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 now need to branch on the y value by adding the constraints that $y \leq 125$ or $y \geq 126$.

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

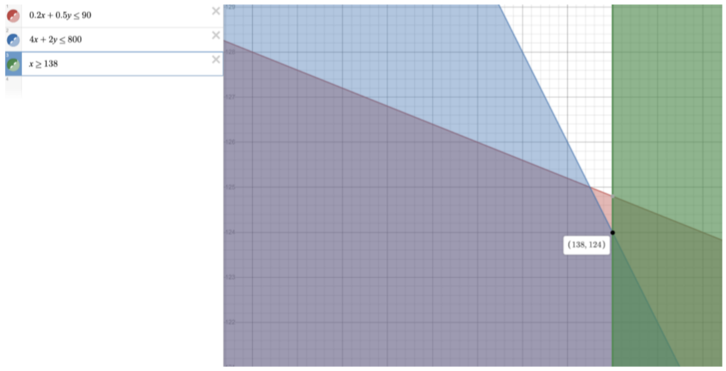


Since the linear programming solution (137, 125) is also an integer solution, we stop branching on the left-left branch and return this point along with its objective value of 7860.

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

Since the linear programming solution (135, 126) is also an integer solution, we stop branching on the left-left branch and return this point along with its objective value of 7830.

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



Since the linear programming solution (138, 124) is also an integer solution, we stop branching on the right branch and return this point, (138, 124), and objective value, 7860. The final solution returned will be (137,125) with objective value 7860.

Note that depending on implementation, (138, 124), with objective value 7860 is also an acceptable solution.

(c) 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.

(d) 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 $x = 0$).

2 4-Queens (CSP to IP)

Recall our formulation of the 4-Queens Problem as a CSP. Our goal was to place 4 chess queens on a 4x4 board such that no two queens are on the same row, column, or diagonal.

Now, let's try formulating the 4-Queens problem as an integer programming problem. We have defined the variables for you as a first step.

Variables: 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.

(a) What is the goal?
Hint: Think about how we can write the goal by summing over the variables defined above.

Goal: We want to find the $\max_x \sum_i \sum_j x_{ij}$ such that $x_{ij} \in \{0,1\}$.

(b) What are the constraints?
Hint: How many queens can we have in any given row, column, or diagonal? How can we write these as summations over the variables?

Constraints:

- Only one queen in each row
Fix i and iterate over each column, ensuring they sum up to ≤ 1.

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

- Only one queen in each column.
Fix j and iterate over each row, ensuring they sum up to ≤ 1.

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

- 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 ...)

- 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 $\leq$ -1.

3 Discussion Based Warm-ups

(a) Vocabulary check: Are you familiar with the following terms?

- Symbols:

  Variables that can be T/F (capital letter)

- Operators:

  And ($\wedge$), Or ($\vee$), Implies ($\Rightarrow$), Equivalent ($\iff$)

- Sentences:

  Symbols connected with operators, can be T/F

- Equivalence:

  True in all models that A and B imply each other (A equivalent to B)

- Literals:

  Atomic Sentence

- Knowledge Base:

  Sentences known to be true

- Entailment:

  A entails B iff for every model that satisfies A, B is also true

- Clauses (Definite vs. Horn Clauses):

  Clause: A conjunction of literals
  Definite Clause: Clause with exactly one positive literal
  Horn Clause: Clause with at most one positive literal

- Model Checking:

  Check if sentences are true in given model/check entailment

- Theorem Proving:

  Search for sequence of proof steps (e.g. Forward Chaining)

- Modus Ponens:

  From P and ($P \Rightarrow Q$), infer Q

(b) Determine which of the following are correct, and explain your reasoning.

- $(A \vee B) \models (A \Rightarrow B)$

  False (when A is true and B is False, $A \vee B$ is true but $A \Rightarrow B$ is false)

- $A \iff B \models A \vee \neg B$

  True (the RHS is $B \Rightarrow A$, which is one of the conjuncts in the definition of $A \iff B$)

- $(A \vee B) \wedge \neg(A \Rightarrow B)$ is satisfiable

True (the model has A and $\neg B$)