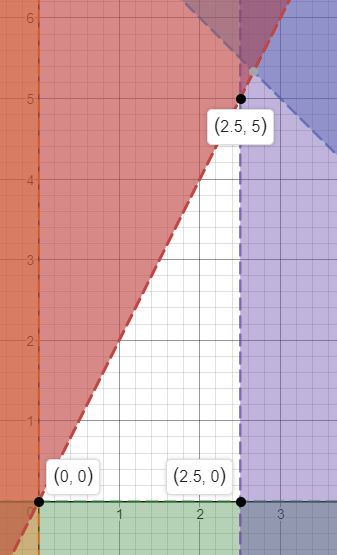
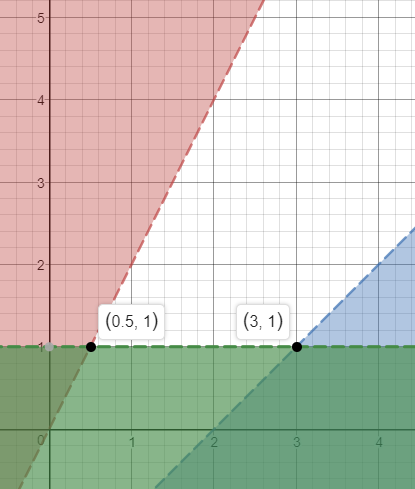
1. Maximize 3x + 2y

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Constraints | 2x – y ≥ 0 | x + y ≤ 8 | x ≤ 2.5 | x ≥ 0 | y ≥ 0 |
| Lines | y > 2x | y > 8 - x | x > 2.5 | x < 0 | y < 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| Vertex | X | Y | 3x + 2y |
| (0, 0) | 0 | 0 | 0 |
| (2.5, 0) | 2.5 | 0 | 7.5 |
| (2.5, 5) | 2.5 | 5 | 17.5 |

  
Graphing these new formulated line equations will provide the feasibility region by excluding given areas of the graph’s domain and range. Now the x and y values from these vertices will be inputs to the original maximize function. This allows us to find the optimal solution.

Therefore, the optimal solution to maximize this function given the constraints is x = 2.5 and y = 5.

1. Yes, it is possible for a linear program in two variables to have an infinite feasibility region but also an optimal solution of bounded cost. If the feasibility region went from 0 to infinity in both x and y axis, and the goal was to minimize the cost.  
     
   Example:  
     
   Find the optimal bounded cost of Cost = 4x + 2y. The feasibility region is the white part of the grid, and it extends infinitely. This happens because the constraints formed 2 lines which do not cross in the feasibility region after the constraint y ≥ 1 is applied.

|  |  |  |  |
| --- | --- | --- | --- |
| Vertex | X | Y | Cost: 4x + 2y |
| (0.5, 1) | 0.5 | 1 | 4 |
| (3, 1) | 3 | 1 | 14 |

Therefore, the optimal solution would be x = 0.5 and y = 1.

1. Setting the variable x to the number of rings, and y to the number of belts Eva makes in a week. To maximize profits, the optimal solution will be to maximize the expression 15x + 20y with the given constraints.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Constraints | 0.5x + y ≤ 10 | x + y ≤ 15 | x ≥ 0 | y ≥ 0 |
| Lines | y > 10 – 0.5x | y > 15 - x | x < 0 | y < 0 |

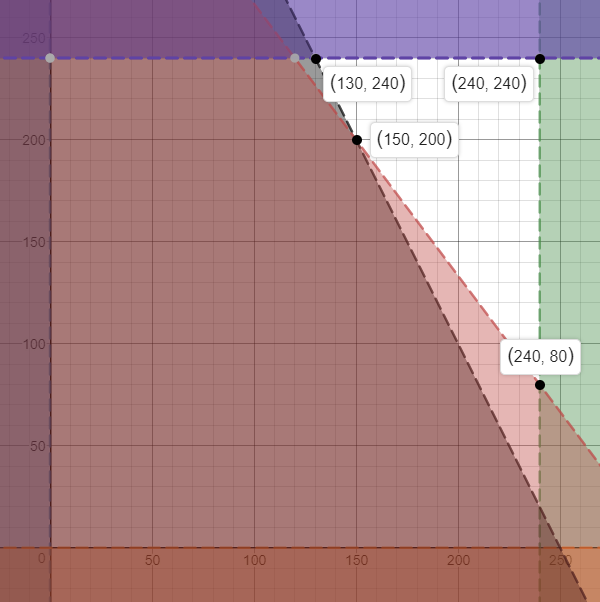


|  |  |  |  |
| --- | --- | --- | --- |
| Vertex | X: Rings | Y: Belts | Profit: 15x + 20y |
| (0, 0) | 0 | 0 | $0 |
| (0, 10) | 0 | 10 | $200 |
| (10, 5) | 10 | 5 | $250 |
| (15, 0) | 15 | 0 | $225 |

Therefore the best combination of rings and belts for Eva to make would be 10 rings and 5 belts per week. Making her a profit of $250/week.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Constraints | 0 ≤ x x ≤ 240 | 0 ≤ y  y ≤ 240 | 20x + 10y ≥ 5000 | 40x + 30y ≥ 12,000 | 60x + 50y ≤ 30,000 |
| Lines | x < 0 x > 240 | y < 0 y > 240 | y < 500 – 2x | y < 400 - x | y > 600 - x |

1. x and y being the number of days the Ontario and Quebec factory operates per year respectively. The function to minimize is 960,000 x + 750,000 y.



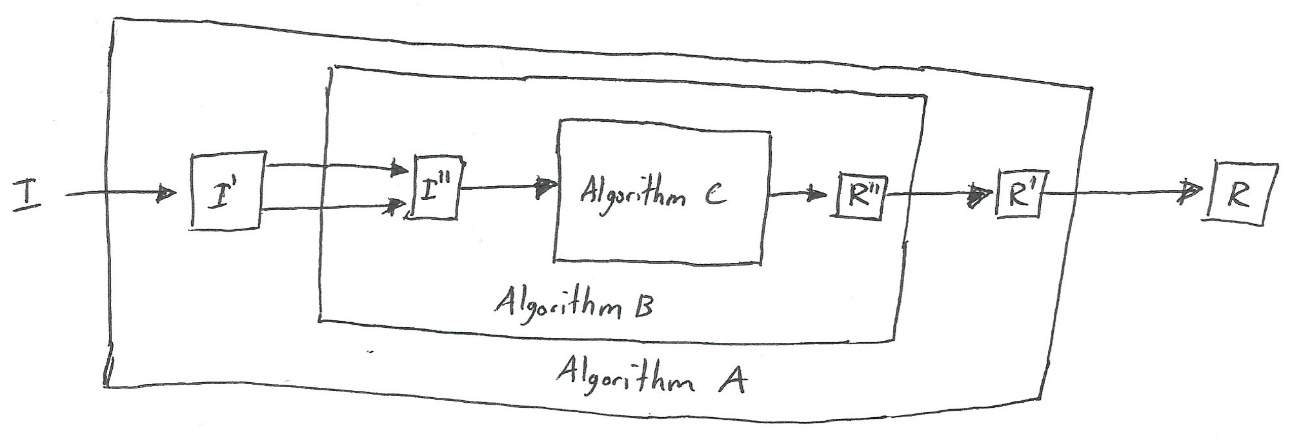
|  |  |  |  |
| --- | --- | --- | --- |
| Vertex | X: Ontario | Y: Quebec | Cost: 960,000 x + 750,000 y |
| (130, 240) | 130 | 240 | $304,800,000 |
| (150, 200) | 150 | 200 | $294,000,000 |
| (240, 80) | 240 | 80 | $290,400,000 |
| (240, 240) | 240 | 240 | $410,400,000 |

Therefore, the Ontario factory should run all possible 240 days and the Quebec factory for only 80 each year to meet the production requirments at a minimum cost.

1. If 4SAT is proven to be both NP and NP-hard, then it is also NP-complete.  
     
   NP Proof:  
   We know 4SAT is in NP, because there is an NP (nondeterministic polynomial time) algorithm which takes 4SAT and a truth assignment (x1 = True, x2 = False, etc.). This makes it very easy to verify the correctness of a certificate, done in linear time. Resulting in either the cetificate being verified or failing the verification, which will also take linear time, ensuring it stays in nondeterministic polynomial time.

NP-Hard Proof:  
In class we reduced 3SAT to 4SAT by reducing each of its clauses to 4SAT. For example: (a1 ∪ a2 ∪ a3) 🡪 (a1 ∪ a2 ∪ a3 ∪ a4) ∩ (a1 ∪ a2 ∪ a3 ∪ ¬a4)  
Where a4 is the new literal introduced. If the clause (a1 ∪ a2 ∪ a3) is satisfied, then (a1 ∪ a2 ∪ a3 ∪ a4) ∩ (a1 ∪ a2 ∪ a3 ∪ ¬a4) will also be satisfied, regardless of a4’s value.

Therefore, I have proven 4SAT to be NP-complete as a result of it being proved to be NP and NP-hard.

1. Problem X reduces to problem Y if you can use an algorithm that solves Y to help solve X. This essentially is transforming the first problem into another which will give the correct answer. If we can solve problem Y and massage the inputs to match, we can solve problem X. This conversion must be done in polynomial time to satisfy the definition of a reduction.  
     
   Proof: If A reduces to B and B reduces to C, then A reduces to C.  
     
     
    This diagram shows how A can reduce to C, since A reduces to B and B reduces to C. The reduction is satisfied by definition. Algorithm C can be used to help solve A because Algorithm C helps, Algorithm B which is used by Algorithm A.