1. Exercise Sheet

October 28, 2021

###

Exercise 1

a)

X_1 for number of quince rings X_2 for number of gooseberry bear

Cost function: $C(X_1, X_2) = 2 * X_1 + 2.5 * X_2$

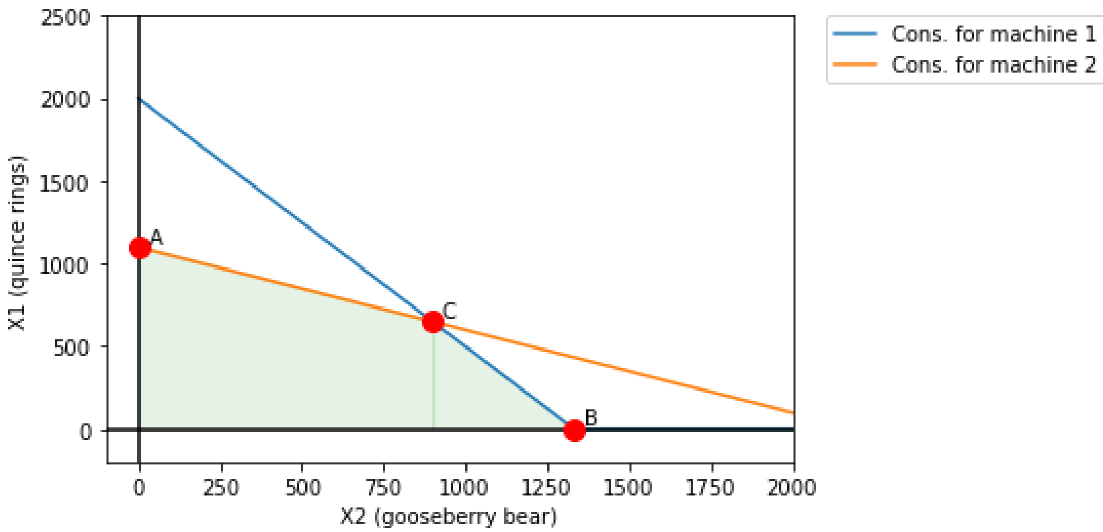
Constraints: Cons.1 $\quad 2 * X_1 + 3 * X_2 \leqslant 4000$ Cons.2 $\quad 4 * X_1 + 2 * X_2 \leqslant 4400$ Cons.3 $\quad 0 \leqslant X_1, X_2$

b)

[2]:
```
import numpy as np
import matplotlib.pyplot as plt

units=np.arange(2000)

#4000 2x1+3x2
x1=(4000-3*units)/2
x1=x1.astype(int)
x1[x1 < 0] = 0
#4400 2x1+3x2
x2=(4400-2*units)/4
x2=x2.astype(int)
x2[x2 < 0] = 0

plt.plot(units, x1, label='Cons. for machine 1')
plt.plot(units, x2, label='Cons. for machine 2')

plt.axhline(0, color='black')
plt.axvline(0, color='black')

plt.xlabel('X2 (gooseberry bear)')
plt.ylabel('X1 (quince rings)')

plt.ylim((-200, 2500))
plt.xlim((-100, 2000))

plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
```

```
plt.fill_between(units,x1, where=x2>=x1, color='green', alpha=0.1)
plt.fill_between(units,x2, where=x2<=x1, color='green', alpha=0.1)

plt.plot(0, 1100, marker=".", markersize=20, color='red')
plt.annotate("A", (30, 1130))
plt.plot(1333, 0, marker=".", markersize=20, color='red')
plt.annotate("B", (1363, 30))
plt.plot(900, 650, marker=".", markersize=20, color='red')
plt.annotate("C", (930, 680))

print("Feasabe solutions marked in green, coordinates are in (X2,X1).")
print("A=",(0, 1100))
print("B=",(1333, 0))
print("A=",(900, 650))
```

```
Feasabe solutions marked in green, coordinates are in (X2,X1).
A= (0, 1100)
B= (1333, 0)
A= (900, 650)
```



c)

```
[3]: def cost(x1,y1):
       return 2*x1+2.5*y1

     print("Profit for A=",cost(0, 1100),"€")
     print("Profit for B=",cost(0, 1333),"€")
     print("Profit for C=",cost(900, 650),"€")
```

```
Profit for A= 2750.0 €
```

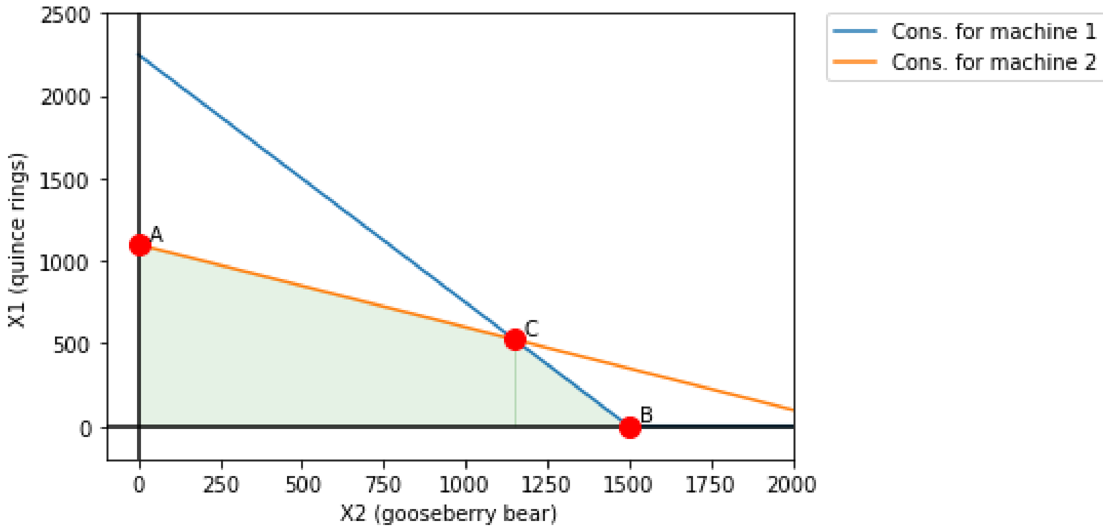
```
Profit for B= 3332.5 €
Profit for C= 3425.0 €
```

Optimal solution at $X_1 = 650$ $X_2 = 900$ with 3425 € profit

d)

[4]:
```python
import numpy as np
import matplotlib.pyplot as plt

units=np.arange(2000)

#4000 2x1+3x2
x1=(4500-3*units)/2
x1=x1.astype(int)
x1[x1 < 0] = 0
#4400 2x1+3x2
x2=(4400-2*units)/4
x2=x2.astype(int)
x2[x2 < 0] = 0

plt.plot(units, x1, label='Cons. for machine 1')
plt.plot(units, x2, label='Cons. for machine 2')

plt.axhline(0, color='black')
plt.axvline(0, color='black')

plt.xlabel('X2 (gooseberry bear)')
plt.ylabel('X1 (quince rings)')

plt.ylim((-200, 2500))
plt.xlim((-100, 2000))

plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.fill_between(units,x1, where=x2>=x1, color='green', alpha=0.1)
plt.fill_between(units,x2, where=x2<=x1, color='green', alpha=0.1)

plt.plot(0, 1100, marker=".", markersize=20, color='red')
plt.annotate("A", (30, 1130))
plt.plot(1500, 0, marker=".", markersize=20, color='red')
plt.annotate("B", (1530, 30))
plt.plot(1150, 525, marker=".", markersize=20, color='red')
plt.annotate("C", (1180, 555))

print("Feasabe solutions marked in green, coordinates are in (X2,X1).")
print("A=",(0, 1100))
print("B=",(1500, 0))
print("C=",(1150, 525))
```

```
Feasabe solutions marked in green, coordinates are in (X2,X1).
A= (0, 1100)
B= (1500, 0)
C= (1150, 525)
```



[5]:
```python
print("Profit for A=",cost(0, 1100),"€")
print("Profit for B=",cost(1500, 0),"€")
print("Profit for C=",cost(1150, 525),"€")
```

```
Profit for A= 2750.0 €
Profit for B= 3000.0 €
Profit for C= 3612.5 €
```

Optimal solution at $X_1 = 525$ $X_2 = 1150$ with 3612.5 € profit

The change is only worth if the repair cost are below 187.5€

###

Exercise 2

The cost of building j warehouses: let $x_j = 1$ if we build a warehouse at location j and otherwise $x_j = 0$. Then the total building cost is: $C_{Build.} = \sum_{j\in N} x_j c_j$ The cost of serving %p needs of costumer i from warehouses j is: $C_{Costum.} = \sum_{j\in N, i\in I} h_{i,j} \frac{p_{i,j}}{100}$ A warehouse at location j can serve a costumer only if it exists and can not serve more than 100%, so our first constraint is: $0 \leq p_{i,j} \leq 100 * x_j$ and $p_{i,j} \in \mathbb{Q}$ A costumer has to be served at 100%, so the second constraint is: $\sum_{j\in N} p_{i,j} = 100$ With these constraints we need to minimize: $C_{Total} = C_{Build.} + C_{Costum.}$

[ ]: