**Homework 2 - BI-OBJECTIVE PERIODIC REVIEW ASSIGNMENT**

Alberto Dian\*, Davide Nascivera, Yunju Li   
*\**Group representative

# Problem description

The aim of this report is to analyse the optimal inventory control policies by accounting for various objective functions. In particular the problem has been studied initially considering only the economical and environmental cost separately, secondly an objective function taking in consideration both of those aspects has been employed. After examining how the optimal reorder quantities were influenced by the degree to which the objective function assigned importance to one aspect over the other, a Pareto efficient frontier was constructed. Furthermore, it has been studied how the non dominated solutions are sensible to the change of the cost coefficients.

# Mathematical models

To achieve the goal previously stated, we followed a precise mathematical model. Using CPLEX and Python, we were able to define a model having the following parameters:

PARAMETERS:

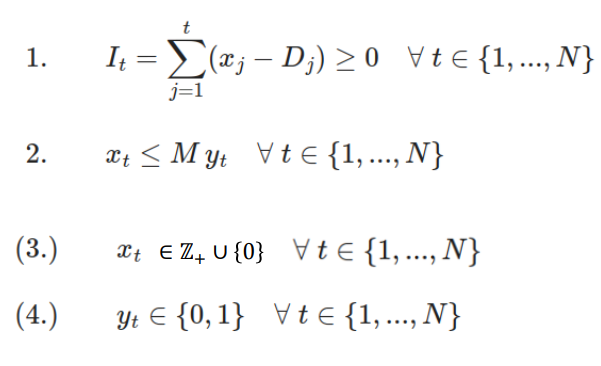
* D demand vector of items per period
* N expressed as |D|
* kC *reorder cost*
* kE *reorder emissions*
* hC *holding costs (for each item in stock at the end of a period)*
* *hE holding emissions (for each item in stock at the end of a period)*

We also defined a decision variable called xlc that is equal to 1 if the class c products are stored in the location l. The used model was subject to the following constraints:

VARIABLES:

* “Yt” variable that assume the value *1 if a reorder is issued at period t, 0 otherwise*
* “It” variable representing the leftover stock at the end of period t (decision variable, eventually).
* “Xt” decision variable representing the number of reordered units at period t (if any).

CONSTRAINTS:



The first constraint stipulates that the remaining inventory at the end of period t must be greater than 0. This is necessary since we want to avoid stock out cost due to backlog or lost sales.

The second formula is a result of the famous technique called “big M methods” used for solving problems of linear programming, where M is a big constant number. In our context it’s used such that if “Xt” is greater than zero, meaning that we have a re-order of Xt products for the period t, “Yt” must also be greater than zero, therefore leading to a cost due to the new issued order. Furthermore, since the “Yt” value can only be {0,1} it’ll be 1. Lastly, “Xt” has the constraint of being an integer number greater and-equal than zero and “Yt” of being binary.

# Main code components

The first part of the exercises focus only on finding the optimal re-order quantity, considering only one of two costs mentioned below.

In particular, as the point *1.a* ask to consider only the economical aspect of the order.

############### ECONOMIC hc,kc

from docplex.mp.model import Model

mdl = Model(name = "Periodic review ECONOMIC")

y = mdl.binary\_var\_list(N, name ="y")

X = mdl.integer\_var\_list(N, name = "X")

for t in range(N):

mdl.add\_constraint(mdl.sum((X[j]-D[j]) for j in range(t+1)) >=0)

# sinceramente non capisco proprio come faccia a funzionare sapendo che D[0] =10 ma D[N+1]= D[5]

for t in range(N):

mdl.add\_constraint(X[t]<=M\*y[t])

for t in range(N):

mdl.add\_constraint(X[t]>=0)

mdl.minimize(mdl.sum(mdl.sum((X[j]-D[j]) for j in range(t+1))\*hc + kc\*y[t] for t in range(N)))

sol = mdl.solve()

sol.display()

min\_obj\_economic = sol.objective\_value

for i in range(N):

X\_eco[i] = sol.get\_value(X[i])

Y\_eco[i] = sol.get\_value(y[i])

print("Minimum economical cost achievable:", min\_obj\_economic)

BarChartPlot(X\_eco, 'Minimization Economical Cost')

For the sake of an easy and interpretable representation, it’s useful to represent “X\_eco“ using a bar chart. The code is mentioned in the appendix due to its repetitiveness and not being fundamental for the reasoning of the exercise.

The second point *(1.b)* of the first question was really similar to the first one but taking in consideration the constant of emission and holding cost instead of the old ones. Apart from few difference the objective function is the main one:

############### ENVIROMENTAL he,ke

[similar code]

mdl2.minimize(mdl2.sum(mdl2.sum((X[j]-D[j]) for j in range(t+1))\*he + ke\*y[t] for

[rest of the code]

print("Minimum enviromental cost achivable:", min\_obj\_env)

BarChartPlot(X\_env,'Minimization Environmental Cost')

As the third point of the first question requires *(1.c)*, we have to compute the percentage of improvement in terms of total emission objective function if our company decided to switch from the economical perspective to an environmental one.

Firstly, we have to quantify environmental cost in a situation where we are optimized regarding the economical cost. For the sake of simplicity, we also evaluate the economical cost regarding the situation where the environmental cost is optimized.

EconomOptim\_EcoCost = round(min\_obj\_economic,2)

EconomOptim\_EnvCost = round(sum(sum((X\_eco[j]-D[j]) for j in range(t+1))\*he + ke\*Y\_eco[t] for t in range(N)),2)

EnvOptim\_EnvCost = round(min\_obj\_env,2)

EnvOptim\_EcoCost = round(sum(sum((X\_env[j]-D[j]) for j in range(t+1))\*hc + kc\*Y\_env[t] for t in range(N)),2)

print("EconomicOptim -> EconomicCost:", EconomOptim\_EcoCost, "EnvCost:", EconomOptim\_EnvCost )

print("EnviromentalOptim-> EconomicCost:", EnvOptim\_EcoCost, "EnvCost:", EnvOptim\_EnvCost)

To calculate the percentage of improvement in the Environmental Cost if we start from the Economic Optimum situation and migrate to the EnvironmentalOptim can be easily computed with the formula. The economical cost we’ll have to pay is readily calculated with the formula below

improvement = EnvOptim\_EnvCost/EconomOptim\_EnvCost\*100

improvement = round(improvement, 2)

print(improvement , "% improvement in environmental terms, migrating from an EconomicalOptimum to an Environmental one" )

print("in order to migrate to an environmental optimum from an economical optimum we'll have to pay", EnvOptim\_EcoCost-EconomOptim\_EcoCost, "€ more")

2 PARETO EFFICIENT SOLUTION:

As requested by the second point in the assignment, we have to consider a scenario where the decision maker wants to weigh the economical and environmental side equally. To do so, it’s sufficient to add the two objective functions together. In general, if we want to consider more the weight of one objective function over the other, we use the formula below changing alpha.



mdl3 = Model(name = "Minimization bot Economical&Enviromental a=50%")

y = mdl3.binary\_var\_list(N, name ="y")

X = mdl3.integer\_var\_list(N, name = "X")

for t in range(N):

mdl3.add\_constraint(mdl3.sum((X[j]-D[j]) for j in range(t+1)) >=0)

for t in range(N):

mdl3.add\_constraint(X[t]<=M\*y[t])

for t in range(N):

mdl3.add\_constraint(X[t]>=0)

mdl3.minimize(mdl3.sum(mdl3.sum((X[j]-D[j]) for j in range(t+1))\*he + ke\*y[t] for t in range(N))+ mdl3.sum(mdl3.sum((X[j]-D[j]) for j in range(t+1))\*hc + kc\*y[t] for t in range(N)))

sol3 = mdl3.solve()

sol3.display()

for i in range(N):

X\_mix[i] = sol3.get\_value(X[i])

Y\_mix[i] = sol3.get\_value(y[i])

MixOptim\_EnvCost = sum(sum((X\_mix[j]-D[j]) for j in range(t+1))\*he + ke\*Y\_mix[t] for t in range(N))

MixOptim\_EcoCost = sum(sum((X\_mix[j]-D[j]) for j in range(t+1))\*hc + kc\*Y\_mix[t] for t in range(N))

BarChartPlot(X\_mix, 'Minimization bot Economical&Enviromental a=50%')

After having also computed the two different cost for the mixed solution weighing 50% of the boh of the, we want to plot them in the objective space, having both of the objective function as axis:

x\_values = [EnvOptim\_EnvCost, ParetoFronteer(0.5)[0], EconomOptim\_EnvCost]

y\_values = [EnvOptim\_EcoCost, ParetoFronteer(0.5)[1], EconomOptim\_EcoCost]

point\_names = ['EnvOptim', 'Mixed 50%', 'EconomOptim']

plt.scatter(x\_values, y\_values)

for i, txt in enumerate(point\_names):

plt.annotate(txt, (x\_values[i], y\_values[i]), textcoords="offset points", xytext=(0,10), ha='center')

plt.xlabel('EnvCost [kg CO2]')

plt.ylabel('EcoCost [€]')

plt.title('Pareto Front')

plt.grid(True)

plt.show()

Following this reasoning, we want to plot the Pareto frontier, composed of all the non-dominated solutions. In order to achieve this result, we defined a function to calculate the environmental and economical cost of any configuration obtained in optimizing the problem, weighing the economical aspect over the environmental one.

def ParetoFronteer(alpha,he = 8,ke = 200,hc = 2,kc = 400):

mdl3 = Model(name = "Periodic review ENVIRONMENTAL")

y = mdl3.binary\_var\_list(N, name ="y")

X = mdl3.integer\_var\_list(N, name = "X")

for t in range(N):

mdl3.add\_constraint(mdl3.sum((X[j]-D[j]) for j in range(t+1)) >=0)

# sinceramente non capisco proprio come faccia a funzionare sapendo che D[0] =10 ma D[N+1]= D[5]

for t in range(N):

mdl3.add\_constraint(X[t]<=M\*y[t])

for t in range(N):

mdl3.add\_constraint(X[t]>=0)

mdl3.minimize(alpha\*mdl3.sum(mdl3.sum((X[j]-D[j]) for j in range(t+1))\*he + ke\*y[t] for t in range(N))+ (1-alpha)\*mdl3.sum(mdl3.sum((X[j]-D[j]) for j in range(t+1))\*hc + kc\*y[t] for t in range(N)))

sol3 = mdl3.solve()

# sol3.display()

for i in range(N):

X\_mix[i] = sol3.get\_value(X[i])

Y\_mix[i] = sol3.get\_value(y[i])

MixOptim\_EnvCost = sum(sum((X\_mix[j]-D[j]) for j in range(t+1))\*he + ke\*Y\_mix[t] for t in range(N))

MixOptim\_EcoCost = sum(sum((X\_mix[j]-D[j]) for j in range(t+1))\*hc + kc\*Y\_mix[t] for t in range(N))

return MixOptim\_EnvCost,MixOptim\_EcoCost

for i in range(len(alphas)):

x\_plot[i], y\_plot[i] = ParetoFronteer(alphas[i])

plt.scatter(x\_plot, y\_plot)

plt.xlabel('EnvCost [kg CO2]')

plt.ylabel('EcoCost [€]')

plt.title('Pareto Front')

plt.grid(True)

As requested by the last point in the assignment, it’s interesting to see how the Pareto frontier is affected by the changing in the different cost/emission factors. Since we want to study all the different variations by changing one parameter at the time, we need to plot 4 different graphs.

1. Changing factor hC ac

alphas = np.arange(0, 1.05, 0.05)

y\_plot = np.zeros\_like(alphas)

x\_plot = np.zeros\_like(alphas)

hc\_1 = [1, 2, 3]

for k in range(len(hc\_1)):

for i in range(len(alphas)):

x\_plot[i], y\_plot[i] = ParetoFronteer(alphas[i],he = 8,ke = 200,hc = hc\_1[k],kc = 400)

plt.scatter(x\_plot, y\_plot, label=f'Fitted Curve hc= {hc\_1[k]}')

plt.xlabel('EnvCost [kg CO2]')

plt.ylabel('EcoCost [€]')

plt.title('Pareto Front varying hc')

plt.legend()

plt.grid(True)

1. changing factor kC

kc\_1 = [300,400, 500]

[APPENDIX AS BEFORE]

for k in range(len(kc\_1)):

for i in range(len(alphas)):

x\_plot[i], y\_plot[i] = ParetoFronteer(alphas[i],he = he,ke = ke,hc = hc ,kc = kc\_1[k])

[PLOT CODE]

1. Changing hE

ke\_1 = [100,200, 300]

[APPENDIX AS BEFORE]

for k in range(len(ke\_1)):

for i in range(len(alphas)):

x\_plot[i], y\_plot[i] = ParetoFronteer(alphas[i],he =he,ke = ke\_1[k],hc = hc,kc = kc)

[PLOT CODE]

1. Changing kE

he\_1 = [5, 8, 11]

[APPENDIX AS BEFORE]

for k in range(len(he\_1)):

for i in range(len(alphas)):

x\_plot[i], y\_plot[i] = ParetoFronteer(alphas[i],he = he\_1[k],ke = ke,hc = hc,kc = kc)

[PLOT CODE]

CODE APPENDIX:

import math

import numpy as np

import matplotlib.pyplot as plt

D = [12, 4, 1, 5, 1, 15, 51, 2, 13, 9, 11, 12, 4, 11, 12, 42, 13, 9, 11]

N = len(D)

kc = 400

hc = 2

ke = 200

he = 8

M = 10000

X\_eco = np.zeros(N)

Y\_eco = np.zeros(N)

X\_env = np.zeros(N)

Y\_env = np.zeros(N)

X\_mix = np.zeros(N)

Y\_mix = np.zeros(N)

def BarChartPlot(X\_values, Title):

# Plotting X values

plt.bar(range(N), X\_values, color='green', label='X values')

plt.xlabel('Time Period (t)')

plt.ylabel('ReOrderQuantities')

plt.xticks(range(0, N, 1)) # Set x-axis interval to 1

plt.title(Title)

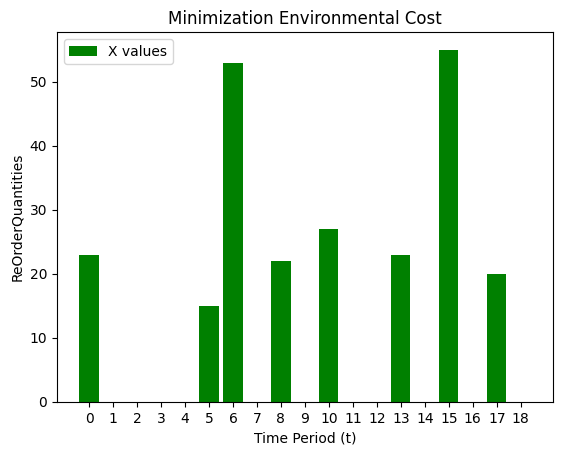
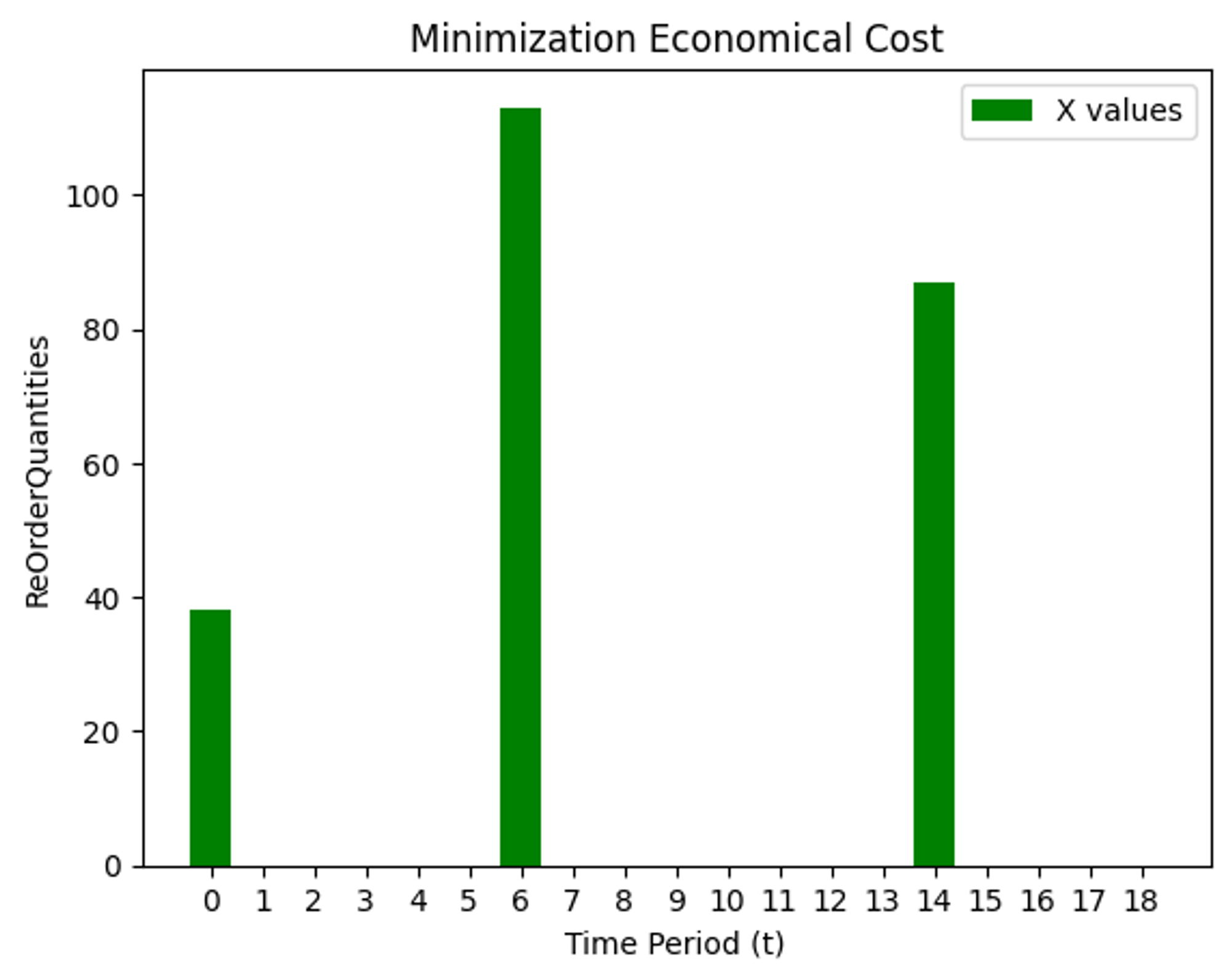
plt.legend()

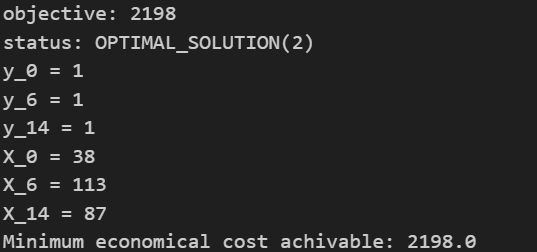
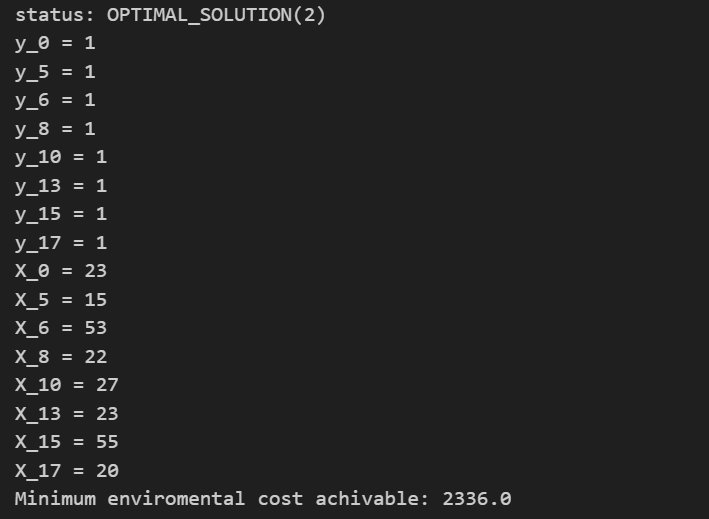
plt.show()

return

# Results and insights

The initial segment of the exercises concentrates solely on determining the optimal reorder quantity, considering only one of the two costs mentioned earlier. Specifically, under point 1.a, the task involves calculating the minimum achievable economic cost, while under 1.b, the objective is to compute the minimum achievable emissions.



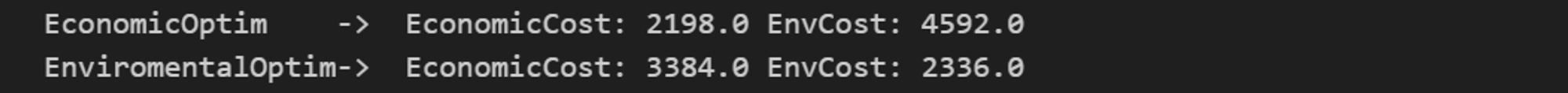
 

|  | **Economical Optim** | **Environmental Optim** |
| --- | --- | --- |
| Issuing cost | kC = 400 | kE = 200 |
| Holding cost | hC = 2 | hE = 8 |

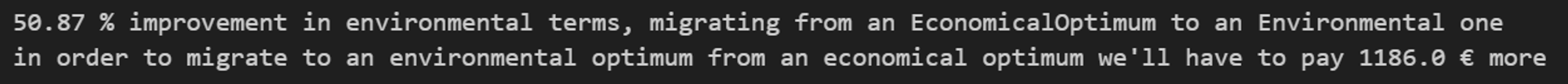
By representing the results obtained using a barchart it’s possible to make qualitative consideration about the 2 different situations we obtained. In particular by looking at the second case it’s possible to notice how we do have considerably more frequent orders but smaller quantities. This is due to the higher holding cost compared to the first one, about four times, while cutting the holding cost in half.

As the third point of the first question requires *(1.c)*, we have to compute the percentage of improvement in terms of total emission objective function if our company decided to switch from the economical perspective to an environmental one.

In order to do that, we have to quantify environmental cost in a situation where we are optimised regarding the economical cost. For the sake of simplicity, we also evaluate the economical cost regarding the situation where the environmental cost is optimised.



To calculate the percentage of improvement in the Environmental Cost if we start from the Economic Optimum situation and migrate to the EnvironmentalOptim can be easily computed with the formula mentioned in the code before. The economical cost we’ll have to pay is readily calculated running the script and it is 1186 €

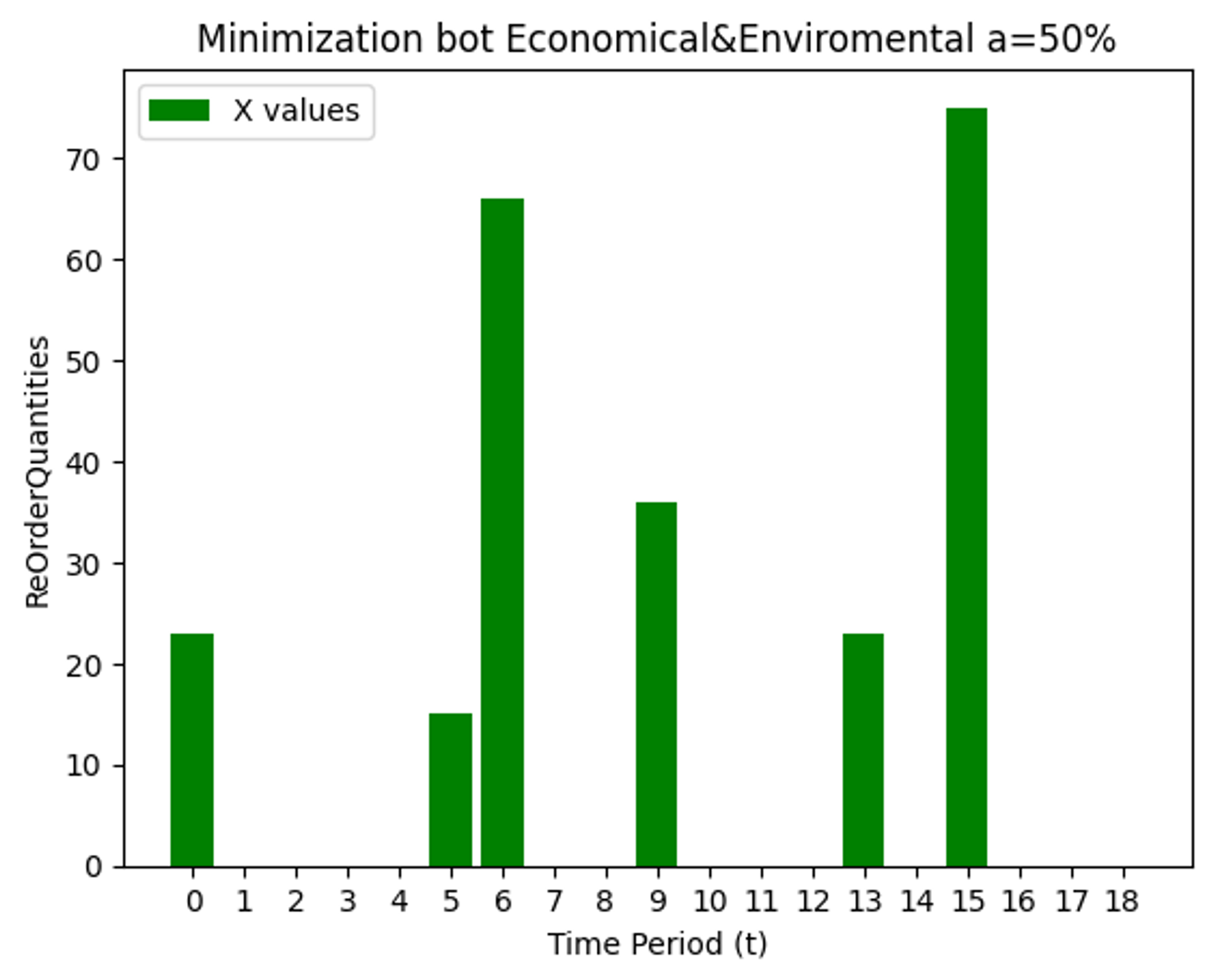


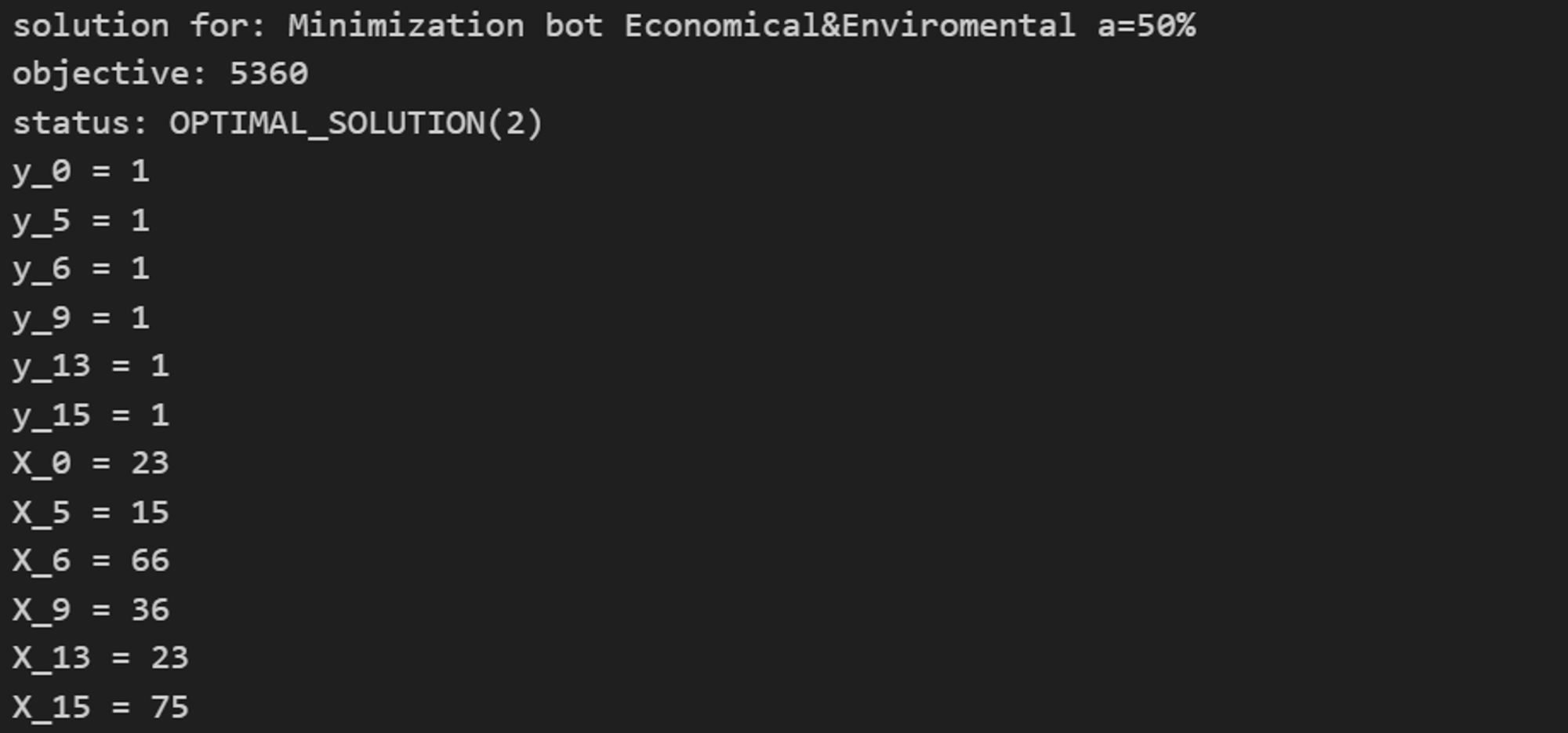
PARETO FRONTIER:

As requested by the second point in the assignment we have to consider a scenario where the decision maker wants to weigh the economical and environmental side equally. To do so it’s sufficient to add the two objective functions together. In general if we want to consider more the weight of one objective function over the other we use the formula below changing alpha.



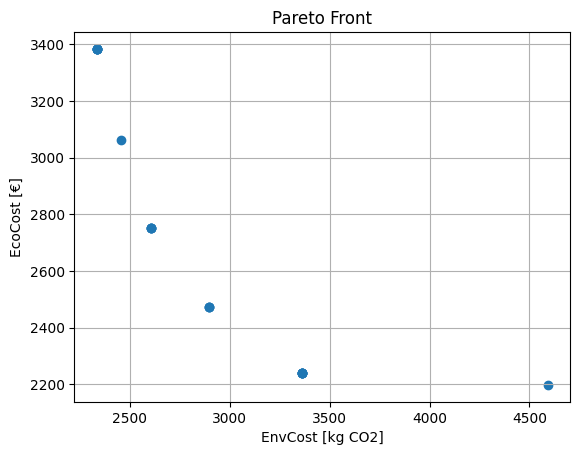
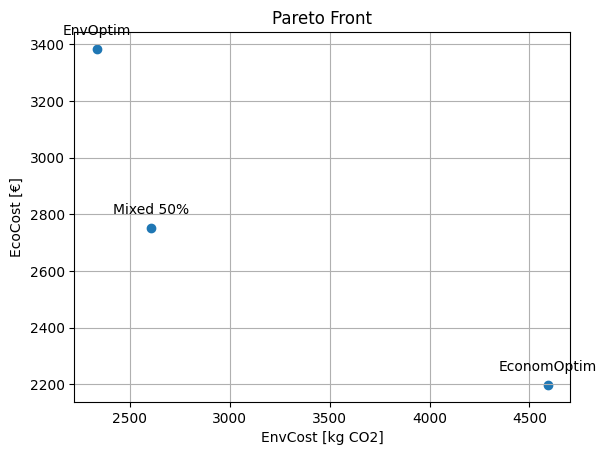
In particular by considering giving equally weight to the environmental and the economic aspect we plot the policy where:



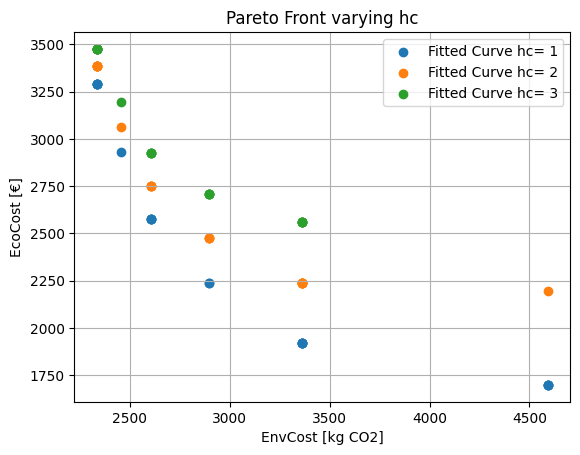
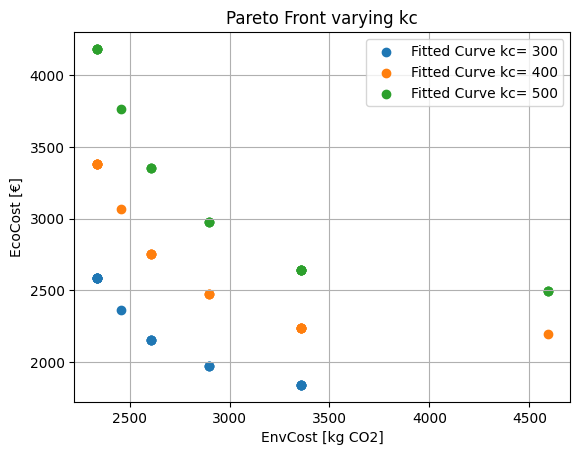


After having computed the two different costs for the mixed solution weighing 50% of the both of them, we want to plot them in the objective space, having both of the objective function as axes. (Figure below at the left).

Following this reasoning we want to plot the Pareto frontier, composed of all the non dominated solutions. In order to achieve this result we defined a function to calculate the environmental and economical cost of any configuration obtained optimizing the problem weighing the economical aspect over the environmental one. (Figure below at the right).



As requested by the last point in the assignment, it’s interesting to see how the Pareto frontier is affected by the changing in the different cost/emission factors. Since we want to study all the different variations by changing one parameter at the time, we need to plot 4 different graphs.

By doing a qualitative analysis it’s possible to see how the border does shift upwards by increasing hc and kc. Thin phenomena is due to having increased the factors of the fist objective function whilst keeping the second one constant. The environmental cost remains constant between the different curves, as expected because it hasn’t been affected by the change of the coefficient while the economical cost does grow due to the increased holding and ordering costs.Si

Similarly, by altering the emission-order/holding costs (ke, he), it is noticeable how the curves shift towards the right, in the direction of the affected objective function.

