Modified genetic algorithm

```
#-------------------------------------------------------------------------------
# Name:        Import Input matrixes and transform for the rU-sbs
#-------------------------------------------------------------------------------

#-------------------------------------------------------------------------------
# Name:        Define initial variables
#-------------------------------------------------------------------------------
from random import choice
from operator import itemgetter, attrgetter
import time
import heapq
import math
import random
from heapq import nlargest
import csv
import cPickle as pickle
from random import sample
from random import uniform
from collections import defaultdict
import multiprocessing
import pdb
import collections
from collections import OrderedDict

usingrobo = 40
usingtask = 72
robonstation = 4
Backrobos=True
kahan=False

OpA=True
OpB=False
OpC=True
OpD=True


for kk in range(121):
    if kk <= 0:
        Robotproblem=False
        failures=random.sample(xrange(1,40),2)
        failures.sort()
        Robotout=failures
    elif kk <= 40:
        Robotproblem=True
        failures=random.sample(xrange(1,40),2)
        failures.sort()
        Robotout=failures
    elif kk <= 80:
        Robotproblem=True
        failures=random.sample(xrange(1,40),4)
        failures.sort()
        Robotout=failures
    elif kk <= 120:
        Robotproblem=True
        failures=random.sample(xrange(1,40),6)
        failures.sort()
        Robotout=failures
```

```
    if Robotproblem:
        mrchecker=[]
        for l in Robotout:
            mrchecker.append(l)

    exForb1 = [1,3,5,7]
    exForb11 = [33,35,37,39]
    exForb2 = [2,4,6,8]
    exForb22 = [40,38,36,34]

    exForb3 = [5,7,9,11]
    exForb33 = [29,31,33,35]
    exForb4 = [6,8,10,12]
    exForb44 = [30,32,36,34]

    exForb5 = [9,11,13,15]
    exForb55 = [25,27,29,31]
    exForb6 = [10,12,14,16]
    exForb66 = [26,28,30,32]

    exForb7 = [13,15,17,19]
    exForb77 = [21,23,25,27]
    exForb8 = [14,16,18,20]
    exForb88 = [22,24,26,28]

    opA=False
    opB=False
    opC=False
    opD=False
    opAB=False
    opAC=False
    opAD=False
    opBC=False
    opBD=False
    opCD=False
    opABC=False
    opABD=False
    opACD=False
    opBCD=False
    opABCD=False

    OpAtest=False
    OpBtest=False
    OpCtest=False
    OpDtest=False

    Mrep=False
    Mrep1=False
    Mrep2=False
    Mrep3=False
    Mrep4=False
    Mrep5=False
    Mrep6=False
    Mrep7=False
    Mrep8=False
    Mrep9=False
    Mrep10=False

    Forb1 = []
    Forb11 = []
```

```
    Forb2 = []
    Forb22= []

    Forb3 = []
    Forb33 = []
    Forb4 = []
    Forb44= []

    Forb5 = []
    Forb55 = []
    Forb6 = []
    Forb66= []

    Forb7 = []
    Forb77 = []
    Forb8 = []
    Forb88= []

    a1=0
    a11=0
    a2=0
    a22=0
    entertwiceA=False
    enteronceA=False

    b3=0
    b33=0
    b4=0
    b44=0
    entertwiceB=False
    enteronceB=False

    c5=0
    c55=0
    c6=0
    c66=0
    entertwiceC=False
    enteronceC=False

    d7=0
    d77=0
    d8=0
    d88=0
    entertwiceD=False
    enteronceD=False

    if Robotproblem:
        lis1111=[]
        lis1112=[]
        lis1113=[]
        lis1114=[]
        lis2221=[]
        lis2222=[]
        lis2223=[]
        lis2224=[]
        lis3331=[]
        lis3332=[]
        lis3333=[]
        lis3334=[]
        lis4441=[]
        lis4442=[]
```

```
        lis4443=[]
        lis4444=[]
        lis5551=[]
        lis5552=[]
        lis5553=[]
        lis5554=[]
        lis6661=[]
        lis6662=[]
        lis6663=[]
        lis6664=[]
        lis7771=[]
        lis7772=[]
        lis7773=[]
        lis7774=[]
        lis8881=[]
        lis8882=[]
        lis8883=[]
        lis8884=[]
        lis9991=[]
        lis9992=[]
        lis9993=[]
        lis9994=[]
        lis10111=[]
        lis10112=[]
        lis10113=[]
        lis10114=[]

    if Robotproblem:
        mrca=[]
        mrcb=[]
        mrcc=[]
        mrcd=[]
        if Backrobos:
            for h in Robotout:
                if OpA:
                    if h in exForb1:
                        Forb1.append(h)
                        mrca.append(h)
                        a1=1
                        OpAtest=True
                    if h in exForb11:
                        Forb11.append(h)
                        mrca.append(h)
                        a11=1
                        OpAtest=True
                    if h in exForb2:
                        Forb2.append(h)
                        mrca.append(h)
                        a2=1
                        OpAtest=True
                    if h in exForb22:
                        Forb22.append(h)
                        mrca.append(h)
                        a22=1
                        OpAtest=True
                if OpB:
                    if h in exForb3:
                        Forb3.append(h)
                        mrcb.append(h)
                        b3=1
                        OpBtest=True
```

```
                    if h in exForb33:
                        Forb33.append(h)
                        mrcb.append(h)
                        b33=1
                        OpBtest=True
                    if h in exForb4:
                        Forb4.append(h)
                        mrcb.append(h)
                        b4=1
                        OpBtest=True
                    if h in exForb44:
                        Forb44.append(h)
                        mrcb.append(h)
                        b44=1
                        OpBtest=True
                if OpC:
                    if h in exForb5:
                        Forb5.append(h)
                        mrcc.append(h)
                        c5=1
                        OpCtest=True
                    if h in exForb55:
                        Forb55.append(h)
                        mrcc.append(h)
                        c55=1
                        OpCtest=True
                    if h in exForb6:
                        Forb6.append(h)
                        mrcc.append(h)
                        c6=1
                        OpCtest=True
                    if h in exForb66:
                        Forb66.append(h)
                        mrcc.append(h)
                        c66=1
                        OpCtest=True
                if OpD:
                    if h in exForb7:
                        Forb7.append(h)
                        mrcd.append(h)
                        d7=1
                        OpDtest=True
                    if h in exForb77:
                        Forb77.append(h)
                        mrcd.append(h)
                        d77=1
                        OpDtest=True
                    if h in exForb8:
                        Forb8.append(h)
                        mrcd.append(h)
                        d8=1
                        OpDtest=True
                    if h in exForb88:
                        Forb88.append(h)
                        mrcd.append(h)
                        d88=1
                        OpDtest=True

    if Robotproblem:
        if Backrobos:
            if a1+a11 == 2:
```

```
                entertwiceA=True
            if a1+a22 == 2:
                entertwiceA=True
            if a2+a11 == 2:
                entertwiceA=True
            if not entertwiceA:
                if a1+a2 > 0:
                    enteronceA=True
                if a11+a22 > 0:
                    enteronceA=True
            if b3+b33 == 2:
                entertwiceB=True
            if b4+b33 == 2:
                entertwiceB=True
            if b4+b44 == 2:
                entertwiceB=True
            if not entertwiceB:
                if b3+b4 > 0:
                    enteronceB=True
                if b33+b44 > 0:
                    enteronceB=True
            if c5+c55 == 2:
                entertwiceC=True
            if c6+c55== 2:
                entertwiceC=True
            if c6+c66 == 2:
                entertwiceC=True
            if not entertwiceC:
                if c5+c6 > 0:
                    enteronceC=True
                if c55+c66 > 0:
                    enteronceC=True
            if d7+d77 == 2:
                entertwiceD=True
            if d7+d88== 2:
                entertwiceD=True
            if d8+d88 == 2:
                entertwiceD=True
            if not entertwiceD:
                if d7+d8 > 0:
                    enteronceD=True
                if d77+d88 > 0:
                    enteronceD=True

    if Robotproblem:
        if Backrobos:
            if OpAtest:
                if OpBtest:
                    if OpCtest:
                        if OpDtest:
                            opABCD=True
                        else:
                            opABC=True
                    elif OpDtest:
                        opABD=True
                    else:
                        opAB=True
                elif OpCtest:
                    if OpDtest:
                        opACD=True
                    else:
```

```
                    opAC=True
                elif OpDtest:
                    opAD=True
                else:
                    opA=True
            elif OpBtest:
                if OpCtest:
                    if OpDtest:
                        opBCD=True
                    else:
                        opBC=True
                elif OpDtest:
                    opBD=True
                else:
                    opB=True
                    ero=ero+1
            elif OpCtest:
                if OpDtest:
                    opCD=True
                else:
                    opC=True
            elif OpDtest:
                opD=True


    Pr={}
    with open('PMatrix2.csv', 'r') as f:
         N= csv.reader(f)
         for item in N:
            ID=int(item[0])
            Other=[]
            for i in range(0,72):
                if int(item[i+1]) == 1:
                    Other.append(i+1)
            Pr[ID]=Other
    f.close()

    Pr1={}
    with open('TaskTimeMatrix.csv', 'r') as f:
         N= csv.reader(f)
         for item in N:
            ID=int(item[0])
            TaskTime=int(item[1])
            Pr1[ID]=TaskTime
    f.close()

    Pr2={}
    with open('mc.csv', 'r') as f:
         N= csv.reader(f)
         for item in N:
            ID=int(item[0])
            Tasks=[]
            for i in range(1,usingtask+1):
                if int(item[i]) == 1:
                    Tasks.append(i)
            Pr2[ID]=Tasks
    f.close()

    if Robotproblem:
        if Backrobos:
            addtask1= []
```

```
            addtask11= []
            addtask2= []
            addtask22= []
            addtask3= []
            addtask33= []
            addtask4= []
            addtask44= []
            addtask5= []
            addtask55= []
            addtask6= []
            addtask66= []
            addtask7= []
            addtask77= []
            addtask8= []
            addtask88= []
            for i in Pr2:
                for Y in Forb1:
                    if i == Y:
                        for w1 in Pr2[i]:
                            addtask1.append(w1)
                for B in Forb2:
                    if i == B:
                        for w2 in Pr2[i]:
                            addtask2.append(w2)
                for Y1 in Forb11:
                    if i == Y1:
                        for w11 in Pr2[i]:
                            addtask11.append(w11)
                for B1 in Forb22:
                    if i == B1:
                        for w22 in Pr2[i]:
                            addtask22.append(w22)
                for Z in Forb3:
                    if i == Z:
                        for w3 in Pr2[i]:
                            addtask3.append(w3)
                for P in Forb4:
                    if i == P:
                        for w4 in Pr2[i]:
                            addtask4.append(w4)
                for Z1 in Forb33:
                    if i == Z1:
                        for w33 in Pr2[i]:
                            addtask33.append(w33)
                for P1 in Forb44:
                    if i == P1:
                        for w44 in Pr2[i]:
                            addtask44.append(w44)
                for Q in Forb5:
                    if i == Q:
                        for w5 in Pr2[i]:
                            addtask5.append(w5)
                for XX in Forb6:
                    if i == XX:
                        for w6 in Pr2[i]:
                            addtask6.append(w6)
                for Q1 in Forb55:
                    if i == Q1:
                        for w55 in Pr2[i]:
                            addtask55.append(w55)
                for XX1 in Forb66:
```

```
                    if i == XX1:
                        for w66 in Pr2[i]:
                            addtask66.append(w66)
                for L in Forb7:
                    if i == L:
                        for w7 in Pr2[i]:
                            addtask7.append(w7)
                for C in Forb8:
                    if i == C:
                        for w8 in Pr2[i]:
                            addtask8.append(w8)
                for L1 in Forb77:
                    if i == L1:
                        for w77 in Pr2[i]:
                            addtask77.append(w77)
                for C1 in Forb88:
                    if i == C1:
                        for w88 in Pr2[i]:
                            addtask88.append(w88)


        if OpAtest:
            Pr2[usingrobo+1]=addtask1
            Pr2[usingrobo+2]=addtask2
            Pr2[usingrobo+9]=addtask11
            Pr2[usingrobo+10]=addtask22
        if OpBtest:
            Pr2[usingrobo+3]=addtask3
            Pr2[usingrobo+4]=addtask4
            Pr2[usingrobo+11]=addtask33
            Pr2[usingrobo+12]=addtask44
        if OpCtest:
            Pr2[usingrobo+5]=addtask5
            Pr2[usingrobo+6]=addtask6
            Pr2[usingrobo+13]=addtask55
            Pr2[usingrobo+14]=addtask66
        if OpDtest:
            Pr2[usingrobo+7]=addtask7
            Pr2[usingrobo+8]=addtask8
            Pr2[usingrobo+15]=addtask77
            Pr2[usingrobo+16]=addtask88

    if Robotproblem:
        mrc=[]
        for rr in mrca:
            if rr not in mrc:
                mrc.append(rr)
        for pp in mrcb:
            if pp not in mrc:
                mrc.append(pp)
        for yy in mrcc:
            if yy not in mrc:
                mrc.append(yy)
        for uu in mrcd:
            if uu not in mrc:
                mrc.append(uu)

    if Robotproblem:
        if len(mrc) > 0:
            for ff in mrc:
                mrchecker.remove(ff)
```

```
    if Robotproblem:
        if kahan:
            Mrep1111=False
            Mrep1112=False
            Mrep1113=False
            Mrep1114=False
            Mrep2221=False
            Mrep2222=False
            Mrep2223=False
            Mrep2224=False
            Mrep3331=False
            Mrep3332=False
            Mrep3333=False
            Mrep3334=False
            Mrep4441=False
            Mrep4442=False
            Mrep4443=False
            Mrep4444=False
            Mrep5551=False
            Mrep5552=False
            Mrep5553=False
            Mrep5554=False
            Mrep6661=False
            Mrep6662=False
            Mrep6663=False
            Mrep6664=False
            Mrep7771=False
            Mrep7772=False
            Mrep7773=False
            Mrep7774=False
            Mrep8881=False
            Mrep8882=False
            Mrep8883=False
            Mrep8884=False
            Mrep9991=False
            Mrep9992=False
            Mrep9993=False
            Mrep9994=False
            Mrep10111=False
            Mrep10112=False
            Mrep10113=False
            Mrep10114=False
            for kl in range(1,41,1):
                if kl == 1:
                    for dd in Pr2[kl]:
                        lis1111.append(dd)
                elif kl == 2:
                    for dd in Pr2[kl]:
                        lis1112.append(dd)
                elif kl == 3:
                    for dd in Pr2[kl]:
                        lis1113.append(dd)
                elif kl == 4:
                    for dd in Pr2[kl]:
                        lis1114.append(dd)
                elif kl == 5:
                    for dd in Pr2[kl]:
                        lis2221.append(dd)
                elif kl == 6:
                    for dd in Pr2[kl]:
                        lis2222.append(dd)
```

```
            elif kl == 7:
                for dd in Pr2[kl]:
                    lis2223.append(dd)
            elif kl == 8:
                for dd in Pr2[kl]:
                    lis2224.append(dd)
            elif kl == 9:
                for dd in Pr2[kl]:
                    lis3331.append(dd)
            elif kl == 10:
                for dd in Pr2[kl]:
                    lis3332.append(dd)
            elif kl == 11:
                for dd in Pr2[kl]:
                    lis3333.append(dd)
            elif kl == 12:
                for dd in Pr2[kl]:
                    lis3334.append(dd)
            elif kl == 13:
                for dd in Pr2[kl]:
                    lis4441.append(dd)
            elif kl == 14:
                for dd in Pr2[kl]:
                    lis4442.append(dd)
            elif kl == 15:
                for dd in Pr2[kl]:
                    lis4443.append(dd)
            elif kl == 16:
                for dd in Pr2[kl]:
                    lis4444.append(dd)
            elif kl == 17:
                for dd in Pr2[kl]:
                    lis5551.append(dd)
            elif kl == 18:
                for dd in Pr2[kl]:
                    lis5552.append(dd)
            elif kl == 19:
                for dd in Pr2[kl]:
                    lis5553.append(dd)
            elif kl == 20:
                for dd in Pr2[kl]:
                    lis5554.append(dd)
            elif kl == 21:
                for dd in Pr2[kl]:
                    lis6661.append(dd)
            elif kl == 22:
                for dd in Pr2[kl]:
                    lis6662.append(dd)
            elif kl == 23:
                for dd in Pr2[kl]:
                    lis6663.append(dd)
            elif kl == 24:
                for dd in Pr2[kl]:
                    lis6664.append(dd)
            elif kl == 25:
                for dd in Pr2[kl]:
                    lis7771.append(dd)
            elif kl == 26:
                for dd in Pr2[kl]:
                    lis7772.append(dd)
            elif kl == 27:
```

```
                    for dd in Pr2[kl]:
                        lis7773.append(dd)
                elif kl == 28:
                    for dd in Pr2[kl]:
                        lis7774.append(dd)
                elif kl == 29:
                    for dd in Pr2[kl]:
                        lis8881.append(dd)
                elif kl == 30:
                    for dd in Pr2[kl]:
                        lis8882.append(dd)
                elif kl == 31:
                    for dd in Pr2[kl]:
                        lis8883.append(dd)
                elif kl == 32:
                    for dd in Pr2[kl]:
                        lis8884.append(dd)
                elif kl == 33:
                    for dd in Pr2[kl]:
                        lis9991.append(dd)
                elif kl == 34:
                    for dd in Pr2[kl]:
                        lis9992.append(dd)
                elif kl == 35:
                    for dd in Pr2[kl]:
                        lis9993.append(dd)
                elif kl == 36:
                    for dd in Pr2[kl]:
                        lis9994.append(dd)
                elif kl == 37:
                    for dd in Pr2[kl]:
                        lis10111.append(dd)
                elif kl == 38:
                    for dd in Pr2[kl]:
                        lis10112.append(dd)
                elif kl == 39:
                    for dd in Pr2[kl]:
                        lis10113.append(dd)
                elif kl == 40:
                    for dd in Pr2[kl]:
                        lis10114.append(dd)
    if Robotproblem:
        if kahan:
            if len(mrchecker) > 0:
                for oo in mrchecker:
                    Mrep=True
                    if oo == 1:
                        adder1=[]
                        for dd in lis1111:
                            adder1.append(dd)
                        for gg in lis2221:
                            adder1.append(gg)
                        Pr2[oo+4]=adder1
                        Mrep1111=True
                    elif oo == 2:
                        adder2=[]
                        for dd in lis1112:
                            adder2.append(dd)
                        for gg in lis2222:
                            adder2.append(gg)
                        Pr2[oo+4]=adder2
```

```
                    Mrep1112=True
                elif oo == 3:
                    adder3=[]
                    for dd in lis1113:
                        adder3.append(dd)
                    for gg in lis2223:
                        adder3.append(gg)
                    Pr2[oo+4]=adder3
                    Mrep1113=True
                elif oo == 4:
                    adder4=[]
                    for dd in lis1114:
                        adder4.append(dd)
                    for gg in lis2224:
                        adder4.append(gg)
                    Pr2[oo+4]=adder4
                    Mrep1114=True
                elif oo == 5:
                    adder5=[]
                    for dd in lis2221:
                        adder5.append(dd)
                    for gg in lis3331:
                        adder5.append(gg)
                    Pr2[oo+4]=adder5
                    if Mrep1111:
                        Pr2[1111]=lis1111
                        Mrep1=True
                    Mrep2221=True
                elif oo == 6:
                    adder6=[]
                    for dd in lis2222:
                        adder6.append(dd)
                    for gg in lis3332:
                        adder6.append(gg)
                    Pr2[oo+4]=adder6
                    if Mrep1112:
                        Pr2[1112]=lis1112
                        Mrep1=True
                    Mrep2222=True
                elif oo == 7:
                    adder7=[]
                    for dd in lis2223:
                        adder7.append(dd)
                    for gg in lis3333:
                        adder7.append(gg)
                    Pr2[oo+4]=adder7
                    if Mrep1113:
                        Pr2[1113]=lis1113
                        Mrep1=True
                    Mrep2223=True
                elif oo == 8:
                    adder8=[]
                    for dd in lis2224:
                        adder8.append(dd)
                    for gg in lis3334:
                        adder8.append(gg)
                    Pr2[oo+4]=adder8
                    if Mrep1114:
                        Pr2[1114]=lis1114
                        Mrep1=True
                    Mrep2224=True
```

```
                    elif oo == 9:
                        adder9=[]
                        for dd in lis3331:
                            adder9.append(dd)
                        for gg in lis4441:
                            adder9.append(gg)
                        Pr2[oo+4]=adder9
                        if Mrep2221:
                            Pr2[2221]=lis2221
                            Mrep2=True
                        Mrep3331=True
                    elif oo == 10:
                        adder10=[]
                        for dd in lis3332:
                            adder10.append(dd)
                        for gg in lis4442:
                            adder10.append(gg)
                        Pr2[oo+4]=adder10
                        if Mrep2222:
                            Pr2[2222]=lis2222
                            Mrep2=True
                        Mrep3332=True
                    elif oo == 11:
                        adder11=[]
                        for dd in lis3333:
                            adder11.append(dd)
                        for gg in lis4443:
                            adder11.append(gg)
                        Pr2[oo+4]=adder11
                        if Mrep2223:
                            Pr2[2223]=lis2223
                            Mrep2=True
                        Mrep3333=True
                    elif oo == 12:
                        adder12=[]
                        for dd in lis3334:
                            adder12.append(dd)
                        for gg in lis4444:
                            adder12.append(gg)
                        Pr2[oo+4]=adder12
                        if Mrep2224:
                            Pr2[2224]=lis2224
                            Mrep2=True
                        Mrep3334=True
                    elif oo == 13:
                        adder13=[]
                        for dd in lis4441:
                            adder13.append(dd)
                        for gg in lis5551:
                            adder13.append(gg)
                        Pr2[oo+4]=adder13
                        if Mrep3331:
                            Pr2[3331]=lis3331
                            Mrep3=True
                        Mrep4441=True
                    elif oo == 14:
                        adder14=[]
                        for dd in lis4442:
                            adder14.append(dd)
                        for gg in lis5552:
                            adder14.append(gg)
```

```
                    Pr2[oo+4]=adder14
                    if Mrep3332:
                        Pr2[3332]=lis3332
                        Mrep3=True
                    Mrep4442=True
                elif oo == 15:
                    adder15=[]
                    for dd in lis4443:
                        adder15.append(dd)
                    for gg in lis5553:
                        adder15.append(gg)
                    Pr2[oo+4]=adder15
                    if Mrep3333:
                        Pr2[3333]=lis3333
                        Mrep3=True
                    Mrep4443=True
                elif oo == 16:
                    adder16=[]
                    for dd in lis4444:
                        adder16.append(dd)
                    for gg in lis5554:
                        adder16.append(gg)
                    Pr2[oo+4]=adder16
                    if Mrep3334:
                        Pr2[3334]=lis3334
                        Mrep3=True
                    Mrep4444=True
                elif oo == 17:
                    adder17=[]
                    for dd in lis5551:
                        adder17.append(dd)
                    for gg in lis6661:
                        adder17.append(gg)
                    Pr2[oo+4]=adder17
                    if Mrep4441:
                        Pr2[4441]=lis4441
                        Mrep4=True
                    Mrep5551=True
                elif oo == 18:
                    adder18=[]
                    for dd in lis5552:
                        adder18.append(dd)
                    for gg in lis6662:
                        adder18.append(gg)
                    Pr2[oo+4]=adder18
                    if Mrep4442:
                        Pr2[4442]=lis4442
                        Mrep4=True
                    Mrep5552=True
                elif oo == 19:
                    adder19=[]
                    for dd in lis5553:
                        adder19.append(dd)
                    for gg in lis6663:
                        adder19.append(gg)
                    Pr2[oo+4]=adder19
                    if Mrep4443:
                        Pr2[4443]=lis4443
                        Mrep4=True
                    Mrep5553=True
                elif oo == 20:
```

```
                    adder20=[]
                    for dd in lis5554:
                        adder20.append(dd)
                    for gg in lis6664:
                        adder20.append(gg)
                    Pr2[oo+4]=adder20
                    if Mrep4444:
                        Pr2[4444]=lis4444
                        Mrep4=True
                    Mrep5554=True
                elif oo == 21:
                    adder21=[]
                    for dd in lis6661:
                        adder21.append(dd)
                    for gg in lis7771:
                        adder21.append(gg)
                    Pr2[oo+4]=adder21
                    if Mrep5551:
                        Pr2[5551]=lis5551
                        Mrep5=True
                    Mrep6661=True
                elif oo == 22:
                    adder22=[]
                    for dd in lis6662:
                        adder22.append(dd)
                    for gg in lis7772:
                        adder22.append(gg)
                    Pr2[oo+4]=adder22
                    if Mrep5552:
                        Pr2[5552]=lis5552
                        Mrep5=True
                    Mrep6662=True
                elif oo == 23:
                    adder23=[]
                    for dd in lis6663:
                        adder23.append(dd)
                    for gg in lis7773:
                        adder23.append(gg)
                    Pr2[oo+4]=adder23
                    if Mrep5553:
                        Pr2[5553]=lis5553
                        Mrep5=True
                    Mrep6663=True
                elif oo == 24:
                    adder24=[]
                    for dd in lis6664:
                        adder24.append(dd)
                    for gg in lis7774:
                        adder24.append(gg)
                    Pr2[oo+4]=adder24
                    if Mrep5554:
                        Pr2[5554]=lis5554
                        Mrep5=True
                    Mrep6664=True
                elif oo == 25:
                    adder25=[]
                    for dd in lis7771:
                        adder25.append(dd)
                    for gg in lis8881:
                        adder25.append(gg)
                    Pr2[oo+4]=adder25
```

```
                    if Mrep6661:
                        Pr2[6661]=lis6661
                        Mrep6=True
                    Mrep7771=True
                elif oo == 26:
                    adder26=[]
                    for dd in lis7772:
                        adder26.append(dd)
                    for gg in lis8882:
                        adder26.append(gg)
                    Pr2[oo+4]=adder26
                    if Mrep6662:
                        Pr2[6662]=lis6662
                        Mrep6=True
                    Mrep7772=True
                elif oo == 27:
                    adder27=[]
                    for dd in lis7773:
                        adder27.append(dd)
                    for gg in lis8883:
                        adder27.append(gg)
                    Pr2[oo+4]=adder27
                    if Mrep6663:
                        Pr2[6663]=lis6663
                        Mrep6=True
                    Mrep7773=True
                elif oo == 28:
                    adder28=[]
                    for dd in lis7774:
                        adder28.append(dd)
                    for gg in lis8884:
                        adder28.append(gg)
                    Pr2[oo+4]=adder28
                    if Mrep6664:
                        Pr2[6664]=lis6664
                        Mrep6=True
                    Mrep7774=True
                elif oo == 29:
                    adder29=[]
                    for dd in lis8881:
                        adder29.append(dd)
                    for gg in lis9991:
                        adder29.append(gg)
                    Pr2[oo+4]=adder29
                    if Mrep7771:
                        Pr2[7771]=lis7771
                        Mrep7=True
                    Mrep8881=True
                elif oo == 30:
                    adder30=[]
                    for dd in lis8882:
                        adder30.append(dd)
                    for gg in lis9992:
                        adder30.append(gg)
                    Pr2[oo+4]=adder30
                    if Mrep7772:
                        Pr2[7772]=lis7772
                        Mrep7=True
                    Mrep8882=True
                elif oo == 31:
                    adder31=[]
```

```
                    for dd in lis8883:
                        adder31.append(dd)
                    for gg in lis9993:
                        adder31.append(gg)
                    Pr2[oo+4]=adder31
                    if Mrep7773:
                        Pr2[7773]=lis7773
                        Mrep7=True
                    Mrep8883=True
                elif oo == 32:
                    adder32=[]
                    for dd in lis8884:
                        adder32.append(dd)
                    for gg in lis9994:
                        adder32.append(gg)
                    Pr2[oo+4]=adder32
                    if Mrep7774:
                        Pr2[7774]=lis7774
                        Mrep7=True
                    Mrep8884=True
                elif oo == 33:
                    adder33=[]
                    for dd in lis9991:
                        adder33.append(dd)
                    for gg in lis10111:
                        adder33.append(gg)
                    Pr2[oo+4]=adder33
                    if Mrep8881:
                        Pr2[8881]=lis8881
                        Mrep8=True
                    Mrep9991=True
                elif oo == 34:
                    adder34=[]
                    for dd in lis9992:
                        adder34.append(dd)
                    for gg in lis10112:
                        adder34.append(gg)
                    Pr2[oo+4]=adder34
                    if Mrep8882:
                        Pr2[8882]=lis8882
                        Mrep8=True
                    Mrep9992=True
                elif oo == 35:
                    adder35=[]
                    for dd in lis9993:
                        adder35.append(dd)
                    for gg in lis10113:
                        adder35.append(gg)
                    Pr2[oo+4]=adder35
                    if Mrep8883:
                        Pr2[8883]=lis8883
                        Mrep8=True
                    Mrep9993=True
                elif oo == 36:
                    adder36=[]
                    for dd in lis9994:
                        adder36.append(dd)
                    for gg in lis10114:
                        adder36.append(gg)
                    Pr2[oo+4]=adder36
                    if Mrep8884:
```

```
                        Pr2[8884]=lis8884
                        Mrep8=True
                    Mrep9994=True
                elif oo == 37:
                    Pr2[10111]=lis10111
                    if Mrep9991:
                        Pr2[9991]=lis9991
                        Mrep9=True
                    Mrep10=True
                elif oo == 38:
                    Pr2[10112]=lis10112
                    if Mrep9992:
                        Pr2[9992]=lis9992
                        Mrep9=True
                    Mrep10=True
                elif oo == 39:
                    Pr2[10113]=lis10113
                    if Mrep9993:
                        Pr2[9993]=lis9993
                        Mrep9=True
                    Mrep10=True
                elif oo == 40:
                    Pr2[10114]=lis10114
                    if Mrep9994:
                        Pr2[9994]=lis9994
                        Mrep9=True
                    Mrep10=True
    else:
        if len(mrchecker) > 0:
            for oo in mrchecker:
                Mrep=True
                if oo == 1:
                    Pr2[1111]=Pr2[oo]
                    for dd in Pr2[1111]:
                        lis1111.append(dd)
                    Mrep1=True
                elif oo == 2:
                    Pr2[1112]=Pr2[oo]
                    for dd in Pr2[1112]:
                        lis1112.append(dd)
                    Mrep1=True
                elif oo == 3:
                    Pr2[1113]=Pr2[oo]
                    for dd in Pr2[1113]:
                        lis1113.append(dd)
                    Mrep1=True
                elif oo == 4:
                    Pr2[1114]=Pr2[oo]
                    for dd in Pr2[1114]:
                        lis1114.append(dd)
                    Mrep1=True
                elif oo == 5:
                    Mrep2=True
                    for gg in Pr2[oo]:
                        lis1111.append(gg)
                    Pr2[1111]=lis1111
                    Pr2[2221]=Pr2[oo]
                    for dd in Pr2[2221]:
                        lis2221.append(dd)
                elif oo == 6:
                    Mrep2=True
```

```
                    for gg in Pr2[oo]:
                        lis1112.append(gg)
                    Pr2[1112]=lis1112
                    Pr2[2222]=Pr2[oo]
                    for dd in Pr2[2222]:
                        lis2222.append(dd)
                elif oo == 7:
                    Mrep2=True
                    for gg in Pr2[oo]:
                        lis1113.append(gg)
                    Pr2[1113]=lis1113
                    Pr2[2223]=Pr2[oo]
                    for dd in Pr2[2223]:
                        lis2223.append(dd)
                elif oo == 8:
                    Mrep2=True
                    for gg in Pr2[oo]:
                        lis1114.append(gg)
                    Pr2[1114]=lis1114
                    Pr2[2224]=Pr2[oo]
                    for dd in Pr2[2224]:
                        lis2224.append(dd)
                elif oo == 9:
                    Mrep3=True
                    for gg in Pr2[oo]:
                        lis2221.append(gg)
                    Pr2[2221]=lis2221
                    Pr2[3331]=Pr2[oo]
                    for dd in Pr2[3331]:
                        lis3331.append(dd)
                elif oo == 10:
                    Mrep3=True
                    for gg in Pr2[oo]:
                        lis2222.append(gg)
                    Pr2[2222]=lis2222
                    Pr2[3332]=Pr2[oo]
                    for dd in Pr2[3332]:
                        lis3332.append(dd)
                elif oo == 11:
                    Mrep3=True
                    for gg in Pr2[oo]:
                        lis2223.append(gg)
                    Pr2[2223]=lis2223
                    Pr2[3333]=Pr2[oo]
                    for dd in Pr2[3333]:
                        lis3333.append(dd)
                elif oo == 12:
                    Mrep3=True
                    for gg in Pr2[oo]:
                        lis2224.append(gg)
                    Pr2[2224]=lis2224
                    Pr2[3334]=Pr2[oo]
                    for dd in Pr2[3334]:
                        lis3334.append(dd)
                elif oo == 13:
                    Mrep4=True
                    for gg in Pr2[oo]:
                        lis3331.append(gg)
                    Pr2[3331]=lis3331
                    Pr2[4441]=Pr2[oo]
                    for dd in Pr2[4441]:
```

```
                        lis4441.append(dd)
                elif oo == 14:
                    Mrep4=True
                    for gg in Pr2[oo]:
                        lis3332.append(gg)
                    Pr2[3332]=lis3332
                    Pr2[4442]=Pr2[oo]
                    for dd in Pr2[4442]:
                        lis4442.append(dd)
                elif oo == 15:
                    Mrep4=True
                    for gg in Pr2[oo]:
                        lis3333.append(gg)
                    Pr2[3333]=lis3333
                    Pr2[4443]=Pr2[oo]
                    for dd in Pr2[4443]:
                        lis4443.append(dd)
                elif oo == 16:
                    Mrep4=True
                    for gg in Pr2[oo]:
                        lis3334.append(gg)
                    Pr2[3334]=lis3334
                    Pr2[4444]=Pr2[oo]
                    for dd in Pr2[4444]:
                        lis4444.append(dd)
                elif oo == 17:
                    Mrep5=True
                    for gg in Pr2[oo]:
                        lis4441.append(gg)
                    Pr2[4441]=lis4441
                    Pr2[5551]=Pr2[oo]
                    for dd in Pr2[5551]:
                        lis5551.append(dd)
                elif oo == 18:
                    Mrep5=True
                    for gg in Pr2[oo]:
                        lis4442.append(gg)
                    Pr2[4442]=lis4442
                    Pr2[5552]=Pr2[oo]
                    for dd in Pr2[5552]:
                        lis5552.append(dd)
                elif oo == 19:
                    Mrep5=True
                    for gg in Pr2[oo]:
                        lis4443.append(gg)
                    Pr2[4443]=lis4443
                    Pr2[5553]=Pr2[oo]
                    for dd in Pr2[5553]:
                        lis5553.append(dd)
                elif oo == 20:
                    Mrep5=True
                    for gg in Pr2[oo]:
                        lis4444.append(gg)
                    Pr2[4444]=lis4444
                    Pr2[5554]=Pr2[oo]
                    for dd in Pr2[5554]:
                        lis5554.append(dd)
                elif oo == 21:
                    Mrep6=True
                    for gg in Pr2[oo]:
                        lis5551.append(gg)
```

```
                    Pr2[5551]=lis5551
                    Pr2[6661]=Pr2[oo]
                    for dd in Pr2[6661]:
                        lis6661.append(dd)
                elif oo == 22:
                    Mrep6=True
                    for gg in Pr2[oo]:
                        lis5552.append(gg)
                    Pr2[5552]=lis5552
                    Pr2[6662]=Pr2[oo]
                    for dd in Pr2[6662]:
                        lis6662.append(dd)
                elif oo == 23:
                    Mrep6=True
                    for gg in Pr2[oo]:
                        lis5553.append(gg)
                    Pr2[5553]=lis5553
                    Pr2[6663]=Pr2[oo]
                    for dd in Pr2[6663]:
                        lis6663.append(dd)
                elif oo == 24:
                    Mrep6=True
                    for gg in Pr2[oo]:
                        lis5554.append(gg)
                    Pr2[5554]=lis5554
                    Pr2[6664]=Pr2[oo]
                    for dd in Pr2[6664]:
                        lis6664.append(dd)
                elif oo == 25:
                    Mrep7=True
                    for gg in Pr2[oo]:
                        lis6661.append(gg)
                    Pr2[6661]=lis6661
                    Pr2[7771]=Pr2[oo]
                    for dd in Pr2[7771]:
                        lis7771.append(dd)
                elif oo == 26:
                    Mrep7=True
                    for gg in Pr2[oo]:
                        lis6662.append(gg)
                    Pr2[6662]=lis6662
                    Pr2[7772]=Pr2[oo]
                    for dd in Pr2[7772]:
                        lis7772.append(dd)
                elif oo == 27:
                    Mrep7=True
                    for gg in Pr2[oo]:
                        lis6663.append(gg)
                    Pr2[6663]=lis6663
                    Pr2[7773]=Pr2[oo]
                    for dd in Pr2[7773]:
                        lis7773.append(dd)
                elif oo == 28:
                    Mrep7=True
                    for gg in Pr2[oo]:
                        lis6664.append(gg)
                    Pr2[6664]=lis6664
                    Pr2[7774]=Pr2[oo]
                    for dd in Pr2[7774]:
                        lis7774.append(dd)
                elif oo == 29:
```

```
                    Mrep8=True
                    for gg in Pr2[oo]:
                        lis7771.append(gg)
                    Pr2[7771]=lis7771
                    Pr2[8881]=Pr2[oo]
                    for dd in Pr2[8881]:
                        lis8881.append(dd)
                elif oo == 30:
                    Mrep8=True
                    for gg in Pr2[oo]:
                        lis7772.append(gg)
                    Pr2[7772]=lis7772
                    Pr2[8882]=Pr2[oo]
                    for dd in Pr2[8882]:
                        lis8882.append(dd)
                elif oo == 31:
                    Mrep8=True
                    for gg in Pr2[oo]:
                        lis7773.append(gg)
                    Pr2[7773]=lis7773
                    Pr2[8883]=Pr2[oo]
                    for dd in Pr2[8883]:
                        lis8883.append(dd)
                elif oo == 32:
                    Mrep8=True
                    for gg in Pr2[oo]:
                        lis7774.append(gg)
                    Pr2[7774]=lis7774
                    Pr2[8884]=Pr2[oo]
                    for dd in Pr2[8884]:
                        lis8884.append(dd)
                elif oo == 33:
                    Mrep9=True
                    for gg in Pr2[oo]:
                        lis8881.append(gg)
                    Pr2[8881]=lis8881
                    Pr2[9991]=Pr2[oo]
                    for dd in Pr2[9991]:
                        lis9991.append(dd)
                elif oo == 34:
                    Mrep9=True
                    for gg in Pr2[oo]:
                        lis8882.append(gg)
                    Pr2[8882]=lis8882
                    Pr2[9992]=Pr2[oo]
                    for dd in Pr2[9992]:
                        lis9992.append(dd)
                elif oo == 35:
                    Mrep9=True
                    for gg in Pr2[oo]:
                        lis8883.append(gg)
                    Pr2[8883]=lis8883
                    Pr2[9993]=Pr2[oo]
                    for dd in Pr2[9993]:
                        lis9993.append(dd)
                elif oo == 36:
                    Mrep9=True
                    for gg in Pr2[oo]:
                        lis8884.append(gg)
                    Pr2[8884]=lis8884
                    Pr2[9994]=Pr2[oo]
```

```
                        for dd in Pr2[9994]:
                            lis9994.append(dd)
                    elif oo == 37:
                        Mrep10=True
                        for gg in Pr2[oo]:
                            lis9991.append(gg)
                        Pr2[9991]=lis9991
                        Pr2[10111]=Pr2[oo]
                    elif oo == 38:
                        Mrep10=True
                        for gg in Pr2[oo]:
                            lis9992.append(gg)
                        Pr2[9992]=lis9992
                        Pr2[10112]=Pr2[oo]
                    elif oo == 39:
                        Mrep10=True
                        for gg in Pr2[oo]:
                            lis9993.append(gg)
                        Pr2[9993]=lis9993
                        Pr2[10113]=Pr2[oo]
                    elif oo == 40:
                        Mrep10=True
                        for gg in Pr2[oo]:
                            lis9994.append(gg)
                        Pr2[9994]=lis9994
                        Pr2[10114]=Pr2[oo]

    print(Robotout)
    if Robotproblem:
        mrtime=[]
        for ss in Pr2:
            if ss > 1000:
                for nn in Pr2[ss]:
                    if nn not in mrtime:
                        mrtime.append(nn)
        if len(mrtime) > 0:
            for ee in mrtime:
                Pr1[ee]=Pr1[ee]*3

    if Robotproblem:
        for RU in Pr2:
            if RU in Robotout:
                Pr2[RU]=[]
    print(Pr2)

    pickle.dump(Pr,open('P1.pkl','wb'),protocol=2)
    pickle.dump(Pr1,open('P2.pkl','wb'),protocol=2)
    pickle.dump(Pr2,open('P3.pkl','wb'),protocol=2)

    #----------------------------------------------------------------------
    # End Import matrix
    #----------------------------------------------------------------------

    #----------------------------------------------------------------------
    # Start the Genetic algorithm with input tranformation into variables
    #----------------------------------------------------------------------


    # ############## Dictionaries with input data ############################

    f = open('P1.pkl','rb')
```

```
Pr = pickle.load(f)
f.close()

f = open('P2.pkl','rb')
Pr2 = pickle.load(f)
f.close()

f = open('P3.pkl','rb')
Pr3 = pickle.load(f)
f.close()

Pr3rev={}
for da in Pr3:
    for di in Pr3[da]:
        if di not in Pr3rev:
            qs=[]
            qs.append(da)
            Pr3rev[di]=qs
        else:
            qs=[]
            for l in Pr3rev[di]:
                if l not in qs:
                    qs.append(l)
            qs.append(da)
            Pr3rev[di]=qs


# ##################### Function for Task sequence ########################

def Order(s):
    for i in range(40):
        for item in s:
            St=s.index(item)
            End=Pr[item]
            if len(End)>0:
                for temp in range(len(End)):
                    End2=s.index(End[temp])
                    if St<End2:
                        it=s[St]
                        s[St]=s[End2]
                        s[End2]=it
    return s

# ####### Function to find a feasible task to station allocation ########

def fescheck(seq1,Path2):
    os=0
    us=0
    roballoc=[]
    robcollec=[]
    robadd = 1
    roc = 1
    rocacc = 1
    mrsetter=1
    stationamount = int(len(seq1))
    aa=0
    bb=0
    cc=0
    dd=0
    aa2=0
    bb2=0
```

```
        cc2=0
        dd2=0
        if Backrobos:
            while roc <= stationamount:
                if  roc == 2:
                    if OpAtest:
                        caa=[]
                        caa.append(usingrobo+1)
                        caa.append(usingrobo+2)
                        roballoc.append(caa)
                        roc=roc+1
                        aa=1
                if  roc == (3+aa):
                    if OpBtest:
                        caa=[]
                        caa.append(usingrobo+3)
                        caa.append(usingrobo+4)
                        roballoc.append(caa)
                        roc=roc+1
                        bb=1
                if  roc == (4+aa+bb):
                    if OpCtest:
                        caa=[]
                        caa.append(usingrobo+5)
                        caa.append(usingrobo+6)
                        roballoc.append(caa)
                        roc=roc+1
                        cc=1
                if  roc == (5+aa+bb+cc):
                    if OpDtest:
                        caa=[]
                        caa.append(usingrobo+7)
                        caa.append(usingrobo+8)
                        roballoc.append(caa)
                        roc=roc+1
                        dd=1
                if  roc == (8+aa+bb+cc):
                    if OpDtest:
                        caa=[]
                        caa.append(usingrobo+15)
                        caa.append(usingrobo+16)
                        roballoc.append(caa)
                        roc=roc+1
                        dd2=1
                if  roc == (9+aa+bb+dd+dd2):
                    if OpCtest:
                        caa=[]
                        caa.append(usingrobo+13)
                        caa.append(usingrobo+14)
                        roballoc.append(caa)
                        roc=roc+1
                        cc2=1
                if  roc == (10+aa+cc+cc2+dd+dd2):
                    if OpBtest:
                        caa=[]
                        caa.append(usingrobo+11)
                        caa.append(usingrobo+12)
                        roballoc.append(caa)
                        roc=roc+1
                        bb2=1
                if  roc == (11+bb+bb2+cc+cc2+dd+dd2):
```

```
                    if OpAtest:
                        caa=[]
                        caa.append(usingrobo+9)
                        caa.append(usingrobo+10)
                        roballoc.append(caa)
                        roc=roc+1
                        aa2=1
                while rocacc <= robonstation:
                    robcollec.append(robadd)
                    rocacc = rocacc +1
                    robadd = robadd +1
                if Mrep:
                    if mrsetter == 1:
                        if Mrep1:
                            robcollec.append(1111)
                            robcollec.append(1112)
                            robcollec.append(1113)
                            robcollec.append(1114)
                    elif mrsetter == 2:
                        if Mrep2:
                            robcollec.append(2221)
                            robcollec.append(2222)
                            robcollec.append(2223)
                            robcollec.append(2224)
                    elif mrsetter == 3:
                        if Mrep3:
                            robcollec.append(3331)
                            robcollec.append(3332)
                            robcollec.append(3333)
                            robcollec.append(3334)
                    elif mrsetter == 4:
                        if Mrep4:
                            robcollec.append(4441)
                            robcollec.append(4442)
                            robcollec.append(4443)
                            robcollec.append(4444)
                    elif mrsetter == 5:
                        if Mrep5:
                            robcollec.append(5551)
                            robcollec.append(5552)
                            robcollec.append(5553)
                            robcollec.append(5554)
                    elif mrsetter == 6:
                        if Mrep6:
                            robcollec.append(6661)
                            robcollec.append(6662)
                            robcollec.append(6663)
                            robcollec.append(6664)
                    elif mrsetter == 7:
                        if Mrep7:
                            robcollec.append(7771)
                            robcollec.append(7772)
                            robcollec.append(7773)
                            robcollec.append(7774)
                    elif mrsetter == 8:
                        if Mrep8:
                            robcollec.append(8881)
                            robcollec.append(8882)
                            robcollec.append(8883)
                            robcollec.append(8884)
                    elif mrsetter == 9:
```

```
                        if Mrep9:
                            robcollec.append(9991)
                            robcollec.append(9992)
                            robcollec.append(9993)
                            robcollec.append(9994)
                    elif mrsetter == 10:
                        if Mrep10:
                            robcollec.append(10111)
                            robcollec.append(10112)
                            robcollec.append(10113)
                            robcollec.append(10114)
                roballoc.append(robcollec)
                roc = roc +1
                rocacc=1
                mrsetter=mrsetter+1
                robcollec=[]
        else:
            while roc <= stationamount:
                while rocacc <= robonstation:
                    robcollec.append(robadd)
                    rocacc = rocacc +1
                    robadd = robadd +1
                if Mrep:
                    if mrsetter == 1:
                        if Mrep1:
                            robcollec.append(1111)
                            robcollec.append(1112)
                            robcollec.append(1113)
                            robcollec.append(1114)
                    elif mrsetter == 2:
                        if Mrep2:
                            robcollec.append(2221)
                            robcollec.append(2222)
                            robcollec.append(2223)
                            robcollec.append(2224)
                    elif mrsetter == 3:
                        if Mrep3:
                            robcollec.append(3331)
                            robcollec.append(3332)
                            robcollec.append(3333)
                            robcollec.append(3334)
                    elif mrsetter == 4:
                        if Mrep4:
                            robcollec.append(4441)
                            robcollec.append(4442)
                            robcollec.append(4443)
                            robcollec.append(4444)
                    elif mrsetter == 5:
                        if Mrep5:
                            robcollec.append(5551)
                            robcollec.append(5552)
                            robcollec.append(5553)
                            robcollec.append(5554)
                    elif mrsetter == 6:
                        if Mrep6:
                            robcollec.append(6661)
                            robcollec.append(6662)
                            robcollec.append(6663)
                            robcollec.append(6664)
                    elif mrsetter == 7:
                        if Mrep7:
```

```
                            robcollec.append(7771)
                            robcollec.append(7772)
                            robcollec.append(7773)
                            robcollec.append(7774)
                    elif mrsetter == 8:
                        if Mrep8:
                            robcollec.append(8881)
                            robcollec.append(8882)
                            robcollec.append(8883)
                            robcollec.append(8884)
                    elif mrsetter == 9:
                        if Mrep9:
                            robcollec.append(9991)
                            robcollec.append(9992)
                            robcollec.append(9993)
                            robcollec.append(9994)
                    elif mrsetter == 10:
                        if Mrep10:
                            robcollec.append(10111)
                            robcollec.append(10112)
                            robcollec.append(10113)
                            robcollec.append(10114)
                roballoc.append(robcollec)
                roc = roc +1
                rocacc=1
                mrsetter=mrsetter+1
                robcollec=[]

        valcolle=[]
        valcolle.append(aa)
        valcolle.append(bb)
        valcolle.append(cc)
        valcolle.append(dd)
        valcolle.append(aa2)
        valcolle.append(bb2)
        valcolle.append(cc2)
        valcolle.append(dd2)
        sp =1
        fes=[]
        adder=[]
        for at in seq1:
            os=os+at
            Taskamount=Path2[us:os]+adder
            ok=[]
            notok=[]
            for ca in Taskamount:
                for cc in roballoc[sp-1]:
                    for d in Pr3rev[ca]:
                        if d == cc:
                            if ca not in ok:
                                ok.append(ca)
            fes.append(ok)
            us=os
            sp=sp+1
            for dd in ok:
                Taskamount.remove(dd)
            adder=Taskamount
        sp=1
        while len(Taskamount) > 0:
            ok=[]
            for bo in Taskamount:
```

```
                for yy in roballoc[sp-1]:
                    for d in Pr3rev[bo]:
                        if d == yy:
                            if bo not in ok:
                                ok.append(bo)
            fes[sp-1]=fes[sp-1]+ok
            for zz in ok:
                Taskamount.remove(zz)
            sp=sp+1
        newstation = []
        jj=0
        newseq=[]
        for rr in fes:
            newstation.append(len(fes[jj]))
            for ww in fes[jj]:
                newseq.append(ww)

            jj=jj+1
        return [newstation,newseq,roballoc,valcolle]

    # ######## Function for calculating each station Cycle time ##############

    def LengthPiece(Path1,scheck,roballoc):
        TaskTime=0
        Ta=[]
        TaTime=[]
        Statadviser={}
        Statsender={}
        for t in Path1:
            posrob=[]
            qr=[]
            for v in Pr3rev[t]:
                posrob.append(v)
            takerob=random.choice(posrob)
            while takerob not in roballoc[scheck]:
                posrob.remove(takerob)
                takerob=random.choice(posrob)
            Statadviser[t]=takerob
            qr.append(takerob)
            Statsender[t]=qr
        robtaker={}
        for ol in  Statsender:
            for ul in Statsender[ol]:
                qs=[]
                if ul not in robtaker.keys():
                    qs.append(ol)
                    robtaker[ul]=qs
                else:
                    for y in robtaker[ul]:
                        qs.append(y)
                    qs.append(ol)
                    robtaker[ul]=qs
        le = len(Path1)
        dbPath1=[]
        for t in Path1:
            dbPath1.append(t)
        Robotime=0
        trueset=[]
        proofer=False
        for t in Path1:
            dbPath1.remove(t)
```

```
            dbTaskTime=0
            for ti in dbPath1:
                if Statadviser[t] ==  Statadviser[ti]:
                    dbTaskTime=Pr2[ti]+dbTaskTime
                    Ta.append(ti)
            TaskTime=Pr2[t]+dbTaskTime
            if Statadviser[t] > 1000:
                mst=TaskTime
                proofer=True
                trueset.append(mst)
                TaskTime=0
            Ta.append(t)
            TaTime.append(TaskTime)
            if proofer:
                mast=max(TaTime)
                mrtime=max(trueset)
                TaskTime=mrtime+mast
            else:
                TaskTime=max(TaTime)
        return [TaskTime,list(set(Ta)),robtaker,Robotime]

    # ######### Formula for the idle time ###################################
    def IdleTime(Ctime,A):
        w=0
        for item in Ctime:
            if Ctime != 0:
                w=w+(A-item)
        return w
    # ############ Start for the solution search ############################
    def LengthTasks(Path):
        CT=[]
        robadder=[]
        robotime=[]
        Task=[]
        Stations=makeStations()
        tester=fescheck(Stations,Path)
        Stations=tester[0]
        Path=tester[1]
        roballoc=tester[2]
        valluer=tester[3]
        aa=valluer[0]
        bb=valluer[1]
        cc=valluer[2]
        dd=valluer[3]
        aa2=valluer[4]
        bb2=valluer[5]
        cc2=valluer[6]
        dd2=valluer[7]
        k=0
        l=0
        scheck=0
        for St in Stations:
            l=l+St
            Length=LengthPiece(Path[k:l],scheck, roballoc)
            if Robotproblem:
                if Backrobos:
                    trace=[]
                    if OpDtest:
                        if scheck == (7+aa+bb+cc):
                            df=Length[0]+CT[4+aa+bb+cc]
                            CT.append(df)
```

```
                            trace.append(df)
                    if OpCtest:
                        if scheck == (8+aa+bb+dd+dd2):
                            df=Length[0]+CT[3+aa+bb]
                            CT.append(df)
                            trace.append(df)
                    if OpBtest:
                        if scheck == (9+aa+cc+cc2+dd+dd2):
                            df=Length[0]+CT[2+aa]
                            CT.append(df)
                            trace.append(df)
                    if OpAtest:
                        if scheck == (10+bb+bb2+cc+cc2+dd+dd2):
                            df=Length[0]+CT[1]
                            CT.append(df)
                            trace.append(df)
                    if len(trace)==0:
                            CT.append(Length[0])
                else:
                    CT.append(Length[0])
            else:
                CT.append(Length[0])
            Task.append(Length[1])
            k=1
            robadder.append(Length[2])
            robotime.append(Length[3])
            scheck=scheck+1
        C=max(CT)
        Obj = -1*C
        Obj1=IdleTime(CT,C)
        return [Obj,Path,Task,Stations,Obj1,robadder,CT]


    # ####### Function to find stations for the list of task sequence ######

    def makeStations(S=[]):

        if Backrobos:
            if opA:
                Number6 = random.randint(25, 35)
                Number5 = random.randint(19, Number6-1)
                Number7 = random.randint(Number6+1, 40)
                Number4 = random.randint(13, Number5-1)
                Number8 = random.randint(Number7+1, 48)
                Number3 = random.randint(8, Number4 - 1)
                Number9 = random.randint(Number8 + 1, 56)
                Number2 = random.randint(7, Number3-1)
                Number10 = random.randint(Number9 + 1, 64)
                Number1 = random.randint(1, Number2-1)
                Number11 = random.randint(Number10 + 1, 65)
                S=[Number1, Number2 - Number1, Number3 - Number2,Number4 - Number3,Number5 - N
umber4,Number6 - Number5,Number7 - Number6,Number8 - Number7,Number9 - Number8, Number10 - Num
ber9, Number11 - Number10, 72 - Number11]
            elif opB:
                Number6 = random.randint(25, 35)
                Number5 = random.randint(19, Number6-1)
                Number7 = random.randint(Number6+1, 40)
                Number4 = random.randint(13, Number5-1)
                Number8 = random.randint(Number7+1, 48)
                Number3 = random.randint(12, Number4 - 1)
```

```
                Number9 = random.randint(Number8 + 1, 56)
                Number2 = random.randint(8, Number3-1)
                Number10 = random.randint(Number9 + 1, 57)
                Number1 = random.randint(1, Number2-1)
                Number11 = random.randint(Number10 + 1, 64)
                S=[Number1, Number2 - Number1, Number3 - Number2,Number4 - Number3,Number5 - N
umber4,Number6 - Number5,Number7 - Number6,Number8 - Number7,Number9 - Number8, Number10 - Num
ber9, Number11 - Number10, 72 - Number11]
            elif opC:
                Number6 = random.randint(25, 35)
                Number5 = random.randint(19, Number6-1)
                Number7 = random.randint(Number6+1, 40)
                Number4 = random.randint(18, Number5-1)
                Number8 = random.randint(Number7+1, 48)
                Number3 = random.randint(13, Number4 - 1)
                Number9 = random.randint(Number8 + 1, 49)
                Number2 = random.randint(8, Number3-1)
                Number10 = random.randint(Number9 + 1, 56)
                Number1 = random.randint(1, Number2-1)
                Number11 = random.randint(Number10 + 1, 64)
                S=[Number1, Number2 - Number1, Number3 - Number2,Number4 - Number3,Number5 - N
umber4,Number6 - Number5,Number7 - Number6,Number8 - Number7,Number9 - Number8, Number10 - Num
ber9, Number11 - Number10, 72 - Number11]
            elif opD:
                Number6 = random.randint(25, 35)
                Number5 = random.randint(24, Number6-1)
                Number7 = random.randint(Number6+1, 40)
                Number4 = random.randint(19, Number5-1)
                Number8 = random.randint(Number7+1, 41)
                Number3 = random.randint(13, Number4 - 1)
                Number9 = random.randint(Number8 + 1, 48)
                Number2 = random.randint(8, Number3-1)
                Number10 = random.randint(Number9 + 1, 56)
                Number1 = random.randint(1, Number2-1)
                Number11 = random.randint(Number10 + 1, 64)
                S=[Number1, Number2 - Number1, Number3 - Number2,Number4 - Number3,Number5 - N
umber4,Number6 - Number5,Number7 - Number6,Number8 - Number7,Number9 - Number8, Number10 - Num
ber9, Number11 - Number10, 72 - Number11]
            elif opAB:
                Number7 = random.randint(25, 35)
                Number6 = random.randint(19, Number7-1)
                Number8 = random.randint(Number7+1, 40)
                Number5 = random.randint(13, Number6-1)
                Number9 = random.randint(Number8+1, 48)
                Number4 = random.randint(12, Number5 - 1)
                Number10 = random.randint(Number9 + 1, 56)
                Number3 = random.randint(8, Number4-1)
                Number11 = random.randint(Number10 + 1, 57)
                Number2 = random.randint(7, Number3-1)
                Number12 = random.randint(Number11 + 1, 64)
                Number1 = random.randint(1, Number2-1)
                Number13 = random.randint(Number12 + 1, 65)
                S=[Number1, Number2 - Number1, Number3 - Number2,Number4 - Number3,Number5 - N
umber4,Number6 - Number5,Number7 - Number6,Number8 - Number7,Number9 - Number8,Number10 - Numb
er9,Number11 - Number10,Number12 - Number11,Number13 - Number12,72 - Number13]
            elif opAC:
                Number7 = random.randint(25, 35)
                Number6 = random.randint(19, Number7-1)
                Number8 = random.randint(Number7+1, 40)
                Number5 = random.randint(18, Number6-1)
                Number9 = random.randint(Number8+1, 48)
```

```
                Number4 = random.randint(13, Number5 - 1)
                Number10 = random.randint(Number9 + 1, 49)
                Number3 = random.randint(8, Number4-1)
                Number11 = random.randint(Number10 + 1, 56)
                Number2 = random.randint(7, Number3-1)
                Number12 = random.randint(Number11 + 1, 64)
                Number1 = random.randint(1, Number2-1)
                Number13 = random.randint(Number12 + 1, 65)
                S=[Number1, Number2 - Number1, Number3 - Number2,Number4 - Number3,Number5 - N
umber4,Number6 - Number5,Number7 - Number6,Number8 - Number7,Number9 - Number8,Number10 - Numb
er9,Number11 - Number10,Number12 - Number11,Number13 - Number12,72 - Number13]
            elif opAD:
                Number7 = random.randint(25, 35)
                Number6 = random.randint(24, Number7-1)
                Number8 = random.randint(Number7+1, 40)
                Number5 = random.randint(19, Number6-1)
                Number9 = random.randint(Number8+1, 41)
                Number4 = random.randint(13, Number5 - 1)
                Number10 = random.randint(Number9 + 1, 48)
                Number3 = random.randint(8, Number4-1)
                Number11 = random.randint(Number10 + 1, 56)
                Number2 = random.randint(7, Number3-1)
                Number12 = random.randint(Number11 + 1, 64)
                Number1 = random.randint(1, Number2-1)
                Number13 = random.randint(Number12 + 1, 65)
                S=[Number1, Number2 - Number1, Number3 - Number2,Number4 - Number3,Number5 - N
umber4,Number6 - Number5,Number7 - Number6,Number8 - Number7,Number9 - Number8,Number10 - Numb
er9,Number11 - Number10,Number12 - Number11,Number13 - Number12,72 - Number13]
            elif opBC:
                Number7 = random.randint(25, 35)
                Number6 = random.randint(19, Number7-1)
                Number8 = random.randint(Number7+1, 40)
                Number5 = random.randint(18, Number6-1)
                Number9 = random.randint(Number8+1, 48)
                Number4 = random.randint(13, Number5 - 1)
                Number10 = random.randint(Number9 + 1, 49)
                Number3 = random.randint(12, Number4-1)
                Number11 = random.randint(Number10 + 1, 56)
                Number2 = random.randint(8, Number3-1)
                Number12 = random.randint(Number11 + 1, 57)
                Number1 = random.randint(1, Number2-1)
                Number13 = random.randint(Number12 + 1, 64)
                S=[Number1, Number2 - Number1, Number3 - Number2,Number4 - Number3,Number5 - N
umber4,Number6 - Number5,Number7 - Number6,Number8 - Number7,Number9 - Number8,Number10 - Numb
er9,Number11 - Number10,Number12 - Number11,Number13 - Number12,72 - Number13]
            elif opBD:
                Number7 = random.randint(25, 35)
                Number6 = random.randint(24, Number7-1)
                Number8 = random.randint(Number7+1, 40)
                Number5 = random.randint(19, Number6-1)
                Number9 = random.randint(Number8+1, 41)
                Number4 = random.randint(13, Number5 - 1)
                Number10 = random.randint(Number9 + 1, 48)
                Number3 = random.randint(12, Number4-1)
                Number11 = random.randint(Number10 + 1, 56)
                Number2 = random.randint(8, Number3-1)
                Number12 = random.randint(Number11 + 1, 57)
                Number1 = random.randint(1, Number2-1)
                Number13 = random.randint(Number12 + 1, 64)
```

```
                S=[Number1, Number2 - Number1, Number3 - Number2,Number4 - Number3,Number5 - N
umber4,Number6 - Number5,Number7 - Number6,Number8 - Number7,Number9 - Number8,Number10 - Numb
er9,Number11 - Number10,Number12 - Number11,Number13 - Number12,72 - Number13]
            elif opCD:
                Number7 = random.randint(25, 35)
                Number6 = random.randint(24, Number7-1)
                Number8 = random.randint(Number7+1, 40)
                Number5 = random.randint(19, Number6-1)
                Number9 = random.randint(Number8+1, 41)
                Number4 = random.randint(18, Number5 - 1)
                Number10 = random.randint(Number9 + 1, 48)
                Number3 = random.randint(13, Number4-1)
                Number11 = random.randint(Number10 + 1, 49)
                Number2 = random.randint(8, Number3-1)
                Number12 = random.randint(Number11 + 1, 56)
                Number1 = random.randint(1, Number2-1)
                Number13 = random.randint(Number12 + 1, 64)
                S=[Number1, Number2 - Number1, Number3 - Number2,Number4 - Number3,Number5 - N
umber4,Number6 - Number5,Number7 - Number6,Number8 - Number7,Number9 - Number8,Number10 - Numb
er9,Number11 - Number10,Number12 - Number11,Number13 - Number12,72 - Number13]
            elif opABC:
                Number8 = random.randint(25, 35)
                Number7 = random.randint(19, Number8-1)
                Number9 = random.randint(Number8+1, 40)
                Number6 = random.randint(18, Number7-1)
                Number10 = random.randint(Number9+1, 48)
                Number5 = random.randint(13, Number6 - 1)
                Number11 = random.randint(Number10 + 1, 49)
                Number4 = random.randint(12, Number5-1)
                Number12 = random.randint(Number11 + 1, 56)
                Number3 = random.randint(8, Number4-1)
                Number13 = random.randint(Number12 + 1, 57)
                Number2 = random.randint(7, Number3-1)
                Number14 = random.randint(Number13 + 1, 64)
                Number1 = random.randint(1, Number2-1)
                Number15 = random.randint(Number14 + 1, 65)
                S=[Number1, Number2 - Number1, Number3 - Number2,Number4 - Number3,Number5 - N
umber4,Number6 - Number5,Number7 - Number6,Number8 - Number7,Number9 - Number8,Number10 - Numb
er9,Number11 - Number10,Number12 - Number11,Number13 - Number12,Number14 - Number13,Number15 -
 Number14,72 - Number15]
            elif opABD:
                Number8 = random.randint(25, 35)
                Number7 = random.randint(24, Number8-1)
                Number9 = random.randint(Number8+1, 40)
                Number6 = random.randint(19, Number7-1)
                Number10 = random.randint(Number9+1, 41)
                Number5 = random.randint(13, Number6 - 1)
                Number11 = random.randint(Number10 + 1, 48)
                Number4 = random.randint(12, Number5-1)
                Number12 = random.randint(Number11 + 1, 56)
                Number3 = random.randint(8, Number4-1)
                Number13 = random.randint(Number12 + 1, 57)
                Number2 = random.randint(7, Number3-1)
                Number14 = random.randint(Number13 + 1, 64)
                Number1 = random.randint(1, Number2-1)
                Number15 = random.randint(Number14 + 1, 65)
                S=[Number1, Number2 - Number1, Number3 - Number2,Number4 - Number3,Number5 - N
umber4,Number6 - Number5,Number7 - Number6,Number8 - Number7,Number9 - Number8,Number10 - Numb
er9,Number11 - Number10,Number12 - Number11,Number13 - Number12,Number14 - Number13,Number15 -
 Number14,72 - Number15]
            elif opACD:
```

```
                Number8 = random.randint(25, 35)
                Number7 = random.randint(24, Number8-1)
                Number9 = random.randint(Number8+1, 40)
                Number6 = random.randint(19, Number7-1)
                Number10 = random.randint(Number9+1, 41)
                Number5 = random.randint(18, Number6 - 1)
                Number11 = random.randint(Number10 + 1, 48)
                Number4 = random.randint(13, Number5-1)
                Number12 = random.randint(Number11 + 1, 49)
                Number3 = random.randint(8, Number4-1)
                Number13 = random.randint(Number12 + 1, 56)
                Number2 = random.randint(7, Number3-1)
                Number14 = random.randint(Number13 + 1, 64)
                Number1 = random.randint(1, Number2-1)
                Number15 = random.randint(Number14 + 1, 65)
                S=[Number1, Number2 - Number1, Number3 - Number2,Number4 - Number3,Number5 - N
umber4,Number6 - Number5,Number7 - Number6,Number8 - Number7,Number9 - Number8,Number10 - Numb
er9,Number11 - Number10,Number12 - Number11,Number13 - Number12,Number14 - Number13,Number15 -
 Number14,72 - Number15]
            elif opBCD:
                Number8 = random.randint(25, 35)
                Number7 = random.randint(24, Number8-1)
                Number9 = random.randint(Number8+1, 40)
                Number6 = random.randint(19, Number7-1)
                Number10 = random.randint(Number9+1, 41)
                Number5 = random.randint(18, Number6 - 1)
                Number11 = random.randint(Number10 + 1, 48)
                Number4 = random.randint(13, Number5-1)
                Number12 = random.randint(Number11 + 1, 49)
                Number3 = random.randint(12, Number4-1)
                Number13 = random.randint(Number12 + 1, 56)
                Number2 = random.randint(8, Number3-1)
                Number14 = random.randint(Number13 + 1, 57)
                Number1 = random.randint(1, Number2-1)
                Number15 = random.randint(Number14 + 1, 64)
                S=[Number1, Number2 - Number1, Number3 - Number2,Number4 - Number3,Number5 - N
umber4,Number6 - Number5,Number7 - Number6,Number8 - Number7,Number9 - Number8,Number10 - Numb
er9,Number11 - Number10,Number12 - Number11,Number13 - Number12,Number14 - Number13,Number15 -
 Number14,72 - Number15]
            elif opABCD:
                Number9 = random.randint(25, 35)
                Number8 = random.randint(24, Number9-1)
                Number10 = random.randint(Number9+1, 40)
                Number7 = random.randint(19, Number8-1)
                Number11 = random.randint(Number10+1, 41)
                Number6 = random.randint(18, Number7 - 1)
                Number12 = random.randint(Number11 + 1, 48)
                Number5 = random.randint(13, Number6-1)
                Number13 = random.randint(Number12 + 1, 49)
                Number4 = random.randint(12, Number5-1)
                Number14 = random.randint(Number13 + 1, 56)
                Number3 = random.randint(8, Number4-1)
                Number15 = random.randint(Number14 + 1, 57)
                Number2 = random.randint(7, Number3-1)
                Number16 = random.randint(Number15 + 1, 64)
                Number1 = random.randint(1, Number2-1)
                Number17 = random.randint(Number16 + 1, 65)
                S=[Number1, Number2 - Number1, Number3 - Number2,Number4 - Number3,Number5 - N
umber4,Number6 - Number5,Number7 - Number6,Number8 - Number7,Number9 - Number8,Number10 - Numb
er9,Number11 - Number10,Number12 - Number11,Number13 - Number12,Number14 - Number13,Number15 -
 Number14,Number16 - Number15,Number17 - Number16,72 - Number17]
```

```
            else:
                Number5 = random.randint(25, 35)
                Number4 = random.randint(19, Number5-1)
                Number6 = random.randint(Number5+1, 40)
                Number3 = random.randint(13, Number4-1)
                Number7 = random.randint(Number6+1, 48)
                Number2 = random.randint(8, Number3 - 1)
                Number8 = random.randint(Number7 + 1, 56)
                Number1 = random.randint(1, Number2-1)
                Number9 = random.randint(Number8 + 1, 64)
                S=[Number1, Number2 - Number1, Number3 - Number2,Number4 - Number3,Number5 - N
umber4,Number6 - Number5,Number7 - Number6,Number8 - Number7,Number9 - Number8, 72 - Number9]


        else:
            Number5 = random.randint(25, 35)
            Number4 = random.randint(19, Number5-1)
            Number6 = random.randint(Number5+1, 40)
            Number3 = random.randint(13, Number4-1)
            Number7 = random.randint(Number6+1, 48)
            Number2 = random.randint(8, Number3 - 1)
            Number8 = random.randint(Number7 + 1, 56)
            Number1 = random.randint(1, Number2-1)
            Number9 = random.randint(Number8 + 1, 64)
            S=[Number1, Number2 - Number1, Number3 - Number2,Number4 - Number3,Number5 - Numbe
r4,Number6 - Number5,Number7 - Number6,Number8 - Number7,Number9 - Number8, 72 - Number9]
        return S

    # ################## CrossOver part Function ##############################

    def InsertPiece(Path,Piece,I1):
        I2=I1+len(Piece)
        Children1=Path[:I1]
        Children3=Path[I2:]
        Children=Children1+Piece+Children3
        MissedTask=list(set(Path) - set(Children))
        if MissedTask:
            Children=Children+MissedTask
        for i in Children:
            c=Children.count(i)
            if c>1:
                Children.pop(Children.index(i))
        Children=Order(Children)
        return Children

    # ################## Xover Function ######################################

    def CrossOver(Parent):
        Parent1=Parent[0]
        Parent2=Parent[1]
        R=range(1,len(Parent1)-1)
        i1=choice(R)
        R.pop(R.index(i1))
        i2=choice(R)
        I1=min(i1,i2)
        I2=max(i1,i2)
        Children=InsertPiece(Parent1,Parent2[I1:I2],I1)
        Children=list(set(Children))
        return Children

    # ################### Mutation Function ##################################
```

```
    def Mutation(Path):
        Ind=choice(range(len(Path)))
        NewNode=choice(xrange(1,72))
        Path=InsertPiece(Path,[NewNode],Ind)
        return Path


    # ########################## Grouping Function ##########################

    def Group(A):
        L=[]
        for i in xrange(5):
            a=[]
            b=[]
            a=choice(A)
            b=choice(A)
            L.append([a[1],b[1]])
        return L

    # ################## Generating population ###############################
    bestopt=[]
    ts=time.clock()
    reply=40
    for d in range(reply):
        t1=time.clock()
        cbresponse=[]
        Pop=[]
        h=[]
        Popsize=100
        for i in range(Popsize):
            s=[]
            s=random.sample(xrange(1,73),72)
            h.append(s)
        h=map(Order,h)
        h=map(LengthTasks,h)
        for item in h:
            heapq.heappush(Pop,item)
        for i in range(100):
            Child=[]
            Ind1=[]
            Parent=[]
            Ind1=heapq.nlargest(5, Pop)
            Parent=Group(Ind1)
            Child=map(CrossOver,Parent)
            Child=map(Mutation,Child)
            Child=map(LengthTasks,Child)
            for item in Child:
                heapq.heappushpop(Pop,item)
            if i == 0:
                cbresponse= Pop
                besttime = time.clock()-t1
            else:
                a= Pop[len(Pop)-1][0]
                b= cbresponse[len(cbresponse)-1][0]
                if a > b:
                    cbresponse= Pop
                    besttime = time.clock()-t1
        if d == 0:
            bestopt=Pop[len(Pop)-1]
            opttime=time.clock()-ts
```

```
        else:
            if Pop[len(Pop)-1][0] > bestopt[0]:
                bestopt=Pop[len(Pop)-1]
                opttime=time.clock()-ts
    runnumb=kk+1
    print('##########################################################')
    print('######### Best Result for run ' ,runnumb,' is: ##################')
    print("Broken robots are ",Robotout)
    print("Simulation time = ",opttime)
    print ('Optimum Cycle time',bestopt[0])
    print ('Task Sequence',bestopt[1])
    print ('Task on Station',bestopt[2])
    print ('Time on Stations',bestopt[6])
    print("Robots making Tasks: ")
    for e in bestopt[5]:
        print(e)
    print ('Station capacity',bestopt[3])
    print ('Ideal time',bestopt[4])
    simtime=time.clock()-ts
    print(simtime)
#-------------------------------------------------------------------------
# End of the Genetic algorithm
#-------------------------------------------------------------------------
```