```
Rem Opt_Brigade 260227_1444
Rem Peter Lohmander

Dim Xa(50), Xc(50), Xm(50)
Dim Ya(50), Yc(50), Ym(50)
Dim Optgraph(100)

Screen _NewImage(1800, 2000, 256)
Open "BrigOut.txt" For Output As #2

GoSub initial
GoSub increments
GoSub AttrCoef
GoSub Exponents
GoSub Objective_parameters
GoSub Decisions

For fYa = -15 To -5 Step 5
   Print ""
   Print #2, ""
   Print "  Parameter fYa = "; fYa
   Print #2, "  Parameter fYa = "; fYa

   Rem *** Optimization ***
   opt_value = -999999999
   opt_uca = 0
   opt_ucm = 0
   For uca = .01 To .99 Step 0.01
      ucm_max = 1 - uca
      For ucm = 0.01 To ucm_max Step .01
         ucc = 1 - uca - ucm
         For t = 0 To topt
            t2 = t + 1

            AttrXa = Ga * (vaa * Ya(t)) ^ raa * (vca * Yc(t)) ^ rca
            If AttrXa > Xa(t) Then AttrXa = Xa(t)
            Xa(t2) = Xa(t) + iXa - AttrXa

            AttrXc = Gc * (vac * Ya(t)) ^ rac * (vcc * Yc(t)) ^ rcc
            If AttrXc > Xc(t) Then AttrXc = Xc(t)
            Xc(t2) = Xc(t) + iXc - AttrXc

            AttrXm = Gm * (vam * Ya(t)) ^ ram * (vcm * Yc(t)) ^ rcm * (vmm * Ym(t)) ^ rmm
            If AttrXm > Xm(t) Then AttrXm = Xm(t)
            Xm(t2) = Xm(t) + iXm - AttrXm

            AttrYa = Ha * (uaa * Xa(t)) ^ saa * (uca * Xc(t)) ^ sca
            If AttrYa > Ya(t) Then AttrYa = Ya(t)
            Ya(t2) = Ya(t) + iYa - AttrYa

            AttrYc = Hc * (uac * Xa(t)) ^ sac * (ucc * Xc(t)) ^ scc
            If AttrYc > Yc(t) Then AttrYc = Yc(t)
            Yc(t2) = Yc(t) + iYc - AttrYc

            AttrYm = Hm * (uam * Xa(t)) ^ sam * (ucm * Xc(t)) ^ scm * (umm * Xm(t)) ^ smm
            If AttrYm > Ym(t) Then AttrYm = Ym(t)
```

```
            Ym(t2) = Ym(t) + iYm - AttrYm

          Next t
          GoSub Objective_function
          If f_value > opt_value Then GoSub New_Opt
        Next ucm
    Next uca

    Print ""
    Print "    Optimal Solution:"
    Print ""
    Print "     objective function = "; opt_value
    Print ""
    Print "     uca      ucc      ucm"
    Print Using "######.##"; opt_uca; opt_ucc; opt_ucm
    Print ""
    Print "              Art       Cav       Mec "
    Print "   X(topt) = "; Xaopt; Xcopt; Xmopt
    Print "   Y(topt) = "; Yaopt; Ycopt; Ymopt
    Print " "

    Print #2, ""
    Print #2, "    Optimal Solution:"
    Print #2, ""
    Print #2, "     objective function = "; opt_value
    Print #2, ""
    Print #2, "     uca      ucc      ucm"
    Print #2, Using "######.##"; opt_uca; opt_ucc; opt_ucm
    Print #2, ""
    Print #2, "              Art       Cav       Mec "
    Print #2, "   X(topt) = "; Xaopt; Xcopt; Xmopt
    Print #2, "   Y(topt) = "; Yaopt; Ycopt; Ymopt
    Print #2, " "

Next fYa
Close #2
End

New_Opt:
opt_uca = uca
opt_ucc = ucc
opt_ucm = ucm
opt_value = f_value
Xaopt = Xa(topt)
Xcopt = Xc(topt)
Xmopt = Xm(topt)
Yaopt = Ya(topt)
Ycopt = Yc(topt)
Ymopt = Ym(topt)
Return

Objective_function:
f_value = fYa * Ya(topt) + fYc * Yc(topt) + fYm * Ym(topt)
Return

Objective_parameters:
```

topt = 10
fYa = -10
fYc = -5
fYm = -10
Return

initial:
Xa(0) = 6
Xc(0) = 6
Xm(0) = 18
Ya(0) = 2
Yc(0) = 2
Ym(0) = 6
Return

increments:
iXa = 1
iXc = 1
iXm = 3
iYa = 3
iYc = 3
iYm = 9
Return

AttrCoef:
Ga = .2
Gc = .1
Gm = .1
Ha = .5
Hc = .3
Hm = .3
Return

Exponents:
raa = .5
rca = 1 - raa
ram = 1 / 3
rcm = 1 / 3
rmm = 1 / 3
rac = .5
rcc = 1 - rac
saa = .5
sca = 1 - saa
sam = 1 / 3
scm = 1 / 3
smm = 1 / 3
sac = .5
scc = 1 - sac
Return

Decisions:
vaa = 1 / 3
vam = 1 / 3
vac = 1 / 3
vca = 1 / 3

vcm = 1 / 3
vcc = 1 / 3
vmm = 1
uaa = 1 / 3
uam = 1 / 3
uac = 1 / 3
uca = 1 / 3
ucm = 1 / 3
ucc = 1 / 3
umm = 1
Return