DHOUT Version 22 Peter Lohmander 2010_08_11 1446

News:

This version (below) of the software should be OK for up to 25 time periods (years) and up to 8 different areas. It is OK in combination with the original "Qbasic" software.

Compare the complete example below!

It is also possible to obtain alternatives to the original "QBasic", for instance "PowerBASIC". With such alternatives, much larger problems should be possible to solve with practically the same code (as the code in this document).

Peter Lohmander

```
"DHIN  Peter Lohmander 2010_08_10  Version 22"
1      "Less"
8      "kmax"
25     "tmax"
.05    "rate"
50     "p"
40     "concos"
100    "z1"
150    "z2"
210    "z3"
160    "z4"
150    "z5"
140    "z6"
220    "z7"
230    "z8"
2000   "c1"
3500   "c2"
4900   "c3"
5500   "c4"
5700   "c5"
5900   "c6"
```

```
6200   "c7"
6700   "c8"
0      "cc11"
1800   "cc12"
3700   "cc13"
4100   "cc14"
4200   "cc15"
4300   "cc16"
4400   "cc17"
4800   "cc18"
1800   "cc21"
0      "cc22"
3700   "cc23"
4100   "cc24"
4200   "cc25"
4300   "cc26"
4400   "cc27"
4800   "cc28"
1900   "cc31"
1800   "cc32"
0      "cc33"
4100   "cc34"
4200   "cc35"
4300   "cc36"
4400   "cc37"
4800   "cc38"
2100   "cc41"
1800   "cc42"
3700   "cc43"
0      "cc44"
4200   "cc45"
4300   "cc46"
4400   "cc47"
4800   "cc48"
2200   "cc51"
1800   "cc52"
3700   "cc53"
4100   "cc54"
0      "cc55"
4300   "cc56"
4400   "cc57"
4800   "cc58"
2800   "cc61"
1800   "cc62"
3700   "cc63"
4100   "cc64"
4200   "cc65"
0      "cc66"
4400   "cc67"
4800   "cc68"
3200   "cc71"
1800   "cc72"
3700   "cc73"
4100   "cc74"
4200   "cc75"
4300   "cc76"
0      "cc77"
4800   "cc78"
3400   "cc81"
1800   "cc82"
3700   "cc83"
4100   "cc84"
4200   "cc85"
4300   "cc86"
```

DHOUT Version 22 Peter Lohmander 2010_08_11 1446
4400   "cc87"
0   "cc88"

OPTIMAL RESULTS FROM DHINV
Software by
Peter Lohmander 2010

OPTIMAL TIME AND STATE DEPENDENT DECISIONS AND EXPECTED PRESENT VALUES

t = 1

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
1	649538.	33	3	0	0 0 0 0 0 0 0 0

t = 2

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
33	662190.	34	8	3	0 0 1 0 0 0 0 0

t = 3

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
34	664361.	36	7	3	0 0 1 0 0 0 0 1

t = 4

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
36	656185.	100	2	0	0 0 1 0 0 0 1 1

t = 5

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
100	636945.	116	4	2	0 1 1 0 0 0 1 1

t = 6

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
116	611789.	124	5	2	0 1 1 1 0 0 1 1

t = 7

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
124	581638.	128	6	2	0 1 1 1 1 0 1 1

t = 8

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
128	547390.	256	1	2	0 1 1 1 1 1 1 1

t = 9

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
256	509048.	256			1 1 1 1 1 1 1 1

t = 10

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
256	465689.	256			1 1 1 1 1 1 1 1

t = 11

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
256	424445.	256			1 1 1 1 1 1 1 1

t = 12

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
256	385212.	256			1 1 1 1 1 1 1 1

t = 13

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
256	347893.	256			1 1 1 1 1 1 1 1

t = 14

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
256	312394.	256			1 1 1 1 1 1 1 1

t = 15

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
256	278626.	256			1 1 1 1 1 1 1 1

t = 16

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
256	246505.	256			1 1 1 1 1 1 1 1

t = 17

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
256	215951.	256			1 1 1 1 1 1 1 1

t = 18

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
256	186887.	256			1 1 1 1 1 1 1 1

t = 19

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
256	159240.	256			1 1 1 1 1 1 1 1

t = 20

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
256	132942.	256			1 1 1 1 1 1 1 1

t = 21

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
256	107926.	256			1 1 1 1 1 1 1 1

t = 22

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
256	84130.	256			1 1 1 1 1 1 1 1

t = 23

i(t)	E(PV)	i(t+1)	DEC	CVIA	Entering Partial States
256	61495.	256			1 1 1 1 1 1 1 1

```
        DHOUT Version 22  Peter Lohmander 2010_08_11  1446
 t =  24
 i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
 ------------------------------------------------------
 256     39964. 256                1 1 1 1 1 1 1 1

 t =  25
 i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
 ------------------------------------------------------
 256     19482. 256                1 1 1 1 1 1 1 1
```

SOFTWARE
****

```
REM
REM DHInv22
REM Peter Lohmander
REM 2010_08_11_1437
CLS

OPEN "DHOut.txt" FOR OUTPUT AS #1
OPEN "DHIN.txt" FOR INPUT AS #2

DIM W(256, 26), M(8, 256), z(8), c(8, 256)
DIM cc(8, 8), MEX(10)
DIM jopt(256, 26), cvia(8, 256)

INPUT #2, Info$
INPUT #2, Less, a$
INPUT #2, kmax, a$
INPUT #2, tmax, a$
INPUT #2, rate, a$
INPUT #2, p, a$
INPUT #2, concos, a$

imax = 2 ^ kmax
jmax = imax

FOR k = 1 TO kmax
INPUT #2, z(k), a$
NEXT k

REM
REM ***** Connection Costs via the Primary Source *****
REM
FOR k = 1 TO kmax
INPUT #2, c(k, 1), a$
c(k, 1) = c(k, 1) + concos * z(k)
NEXT k


REM
REM ***** Costs of connecting one area via another area *****
REM
FOR k = 1 TO kmax
FOR M = 1 TO kmax
INPUT #2, cc(k, M), a$
NEXT M
NEXT k

FOR k1 = 1 TO kmax
  FOR k2 = 1 TO kmax
    IF k2 = k1 THEN GOTO 444
```

```
    cc(k1, k2) = cc(k1, k2) + concos * z(k1)
444 REM
  NEXT k2
NEXT k1


PRINT #1, ""
PRINT #1, "OPTIMAL RESULTS FROM DHINV"
PRINT #1, "Software by "
PRINT #1, "Peter Lohmander 2010"
REM PRINT #1, ""
REM PRINT #1, "tmax = ", tmax, " kmax = ", kmax, " imax = jmax = ";
imax


REM
REM ***** Terminal conditions *****
REM
FOR i = 1 TO imax
 W(i, (tmax + 1)) = 0
NEXT i

REM
REM ***** Calculation of the membership function *****
REM
  mnum = 0
  FOR k = kmax TO 1 STEP -1
    value = 0
    mnum = mnum + 1
    mm = 2 ^ (mnum - 1)
    count = 0
     FOR i = 1 TO imax
       count = count + 1
       M(k, i) = value
       change = 0
       IF count = mm THEN change = 1
       IF change = 1 THEN count = 0
       chdown = 0
       IF value = 1 THEN chdown = 1
       chup = 0
       IF value = 0 THEN chup = 1
       IF (change = 1 AND chdown = 1) THEN value = 0
       IF (change = 1 AND chup = 1) THEN value = 1
  NEXT i
NEXT k


REM
REM ***** Calculation of State Dependent Partial *****
REM ***** Investment Cost Functions               *****
REM
FOR i = 2 TO imax
  FOR k = 1 TO kmax
    IF M(k, i) = 1 THEN c(k, i) = 0
    IF M(k, i) = 1 THEN GOTO 222
    c(k, i) = c(k, 1)
      FOR kconect = 1 TO kmax
        IF M(kconect, i) = 0 THEN GOTO 333
        IF kconect = k THEN GOTO 333
        clok = cc(k, kconect)
        IF clok < c(k, i) THEN cvia(k, i) = kconect
        IF clok < c(k, i) THEN c(k, i) = clok
333 REM
      NEXT kconect
```

```
222 REM
  NEXT k
NEXT i


REM
REM ***** Dynamic Programming via Backward Recursion *****
REM
FOR t = tmax TO 1 STEP -1
    d = EXP(-rate * t)
  FOR i = 1 TO imax
    optF = -999999
    optJ = 0
      FOR j = 1 TO jmax
        neginv = 0
        numinv = 0
          FOR k = 1 TO kmax
            IF (M(k, j) - M(k, i)) = 1 THEN numinv = numinv + 1
            IF (M(k, j) - M(k, i)) < 0 THEN neginv = neginv + 1
          NEXT k
        IF neginv > 0 THEN GOTO 100
        IF numinv > 1 THEN GOTO 100
          net = 0
          FOR k = 1 TO kmax
            net = net + p * M(k, i) * z(k)
          NEXT k
          FOR k = 1 TO kmax
            IF (M(k, j) - M(k, i)) = 1 THEN net = net - c(k, i)
          NEXT k
        F = d * net + W(j, (t + 1))
        IF F > optF THEN optJ = j
        IF F > optF THEN optF = F
100 REM
      NEXT j
    W(i, t) = optF
REM PRINT #1, "t = "; t; " i = "; i; " optF = "; optF; " optJ = ";
optJ
jopt(i, t) = optJ
  NEXT i
NEXT t



PRINT #1, ""
PRINT #1, "OPTIMAL TIME AND STATE DEPENDENT DECISIONS AND EXPECTED
PRESENT VALUES"
instate = 1
FOR t = 1 TO tmax
PRINT #1, ""
PRINT #1, " t = ";
PRINT #1, USING "###"; t

PRINT #1, " i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial
States"
PRINT #1, "
----------------------------------------------------"
FOR i = 1 TO imax

IF (i < instate OR i > instate) AND (Less = 1) THEN GOTO 888


 FOR k = 1 TO kmax
 MEX(k) = M(k, i)
 NEXT k
```

```
PRINT #1, USING "####"; i;

PRINT #1, USING "##########."; W(i, t);


invnumb = 0
FOR k = 1 TO kmax
IF (M(k, jopt(i, t)) - M(k, i)) > 0 THEN invnumb = k
NEXT k
PRINT #1, USING "####"; jopt(i, t);
PRINT #1, "    ";

IF invnumb > 0 THEN PRINT #1, USING "###"; invnumb;
IF invnumb = 0 THEN PRINT #1, "   ";

IF invnumb > 0 THEN PRINT #1, USING "#####"; cvia(invnumb, i);
IF invnumb = 0 THEN PRINT #1, "     ";

PRINT #1, "    ";
FOR k = 1 TO kmax
PRINT #1, USING "##"; MEX(k);
NEXT k
PRINT #1, ""

888 REM

NEXT i
instate = jopt(instate, t)

NEXT t


CLOSE #1
CLOSE #2

END
```