# DHINV
## *- Dynamic distribution net optimization software*
Peter Lohmander 2010-08-13

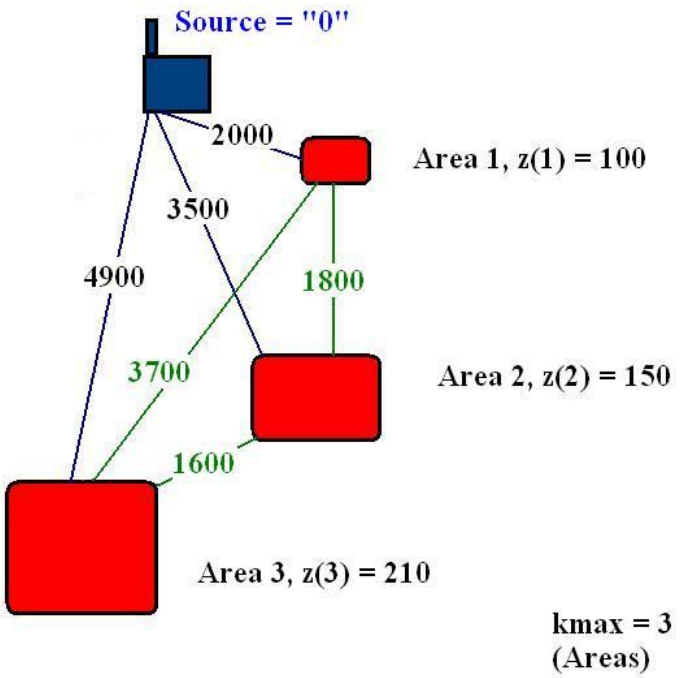


**Figure 1.**
The distribution net optimization problem. The number of units in each area, k, are denoted z(k).
The lines with figures show possible investments in connecting cables or pipelines and the corresponding costs.
The parameters should be written in the input file, DHIN.txt. Then, the software DHINV22.exe or DHINV22.bas can be used to obtain the results.

The input file DHIN.txt is included below.
The parameter Less should have the value 1 if we want reduced output. With value 0, we get complete output.
The paremeter kmax is the total number of areas. The parameter tmax is the total number of periods (normally defined as years, but longer or shorter periods may be used). The rate of interest in the capital market is denoted rate. The level of economic net improvement thanks to the investment, per year and unit (for instance per house in an area) is denoted p. The cost of connection of one unit within an area (for instance one house) is denoted concos. The meaning of the other parameters should be understandable if you compare Figure 1.. Each period, it is possible to select zero or one investment.

```
”DHIN  Peter Lohmander 2010_08_12  A small dynamic distribution net optimization example”
1      ”Less”
3      ”kmax”
25     ”tmax”
.05    ”rate”
10     ”p”
40     ”concos”
100    ”z1”
150    ”z2”
210    ”z3”
2000   ”c1”
3500   ”c2”
4900   ”c3”
0      ”cc11”
1800   ”cc12”
3700   ”cc13”
0      ”cc22”
1600   ”cc23”
3700   ”cc31”
1600   ”cc32”
0      ”cc33”
```

The optimal strategy and results are reported below (File = DHOUT.txt).
Figure 2. illustrates the meaning of most results. E(PV) denotes the total present value.

```
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     34830.   5       1    0     0 0 0

 t =    2
 i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
 --------------------------------------------------------
   5     40538.   7       2    1     1 0 0

 t =    3
 i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
 --------------------------------------------------------
   7     45062.   8       3    2     1 1 0

 t =    4
 i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
 --------------------------------------------------------
   8     51517.   8                  1 1 1
```

```
t =   5
i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
---------------------------------------------------------
  8     47751.   8                 1 1 1

t =   6
i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
---------------------------------------------------------
  8     44168.   8                 1 1 1

t =   7
i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
---------------------------------------------------------
  8     40761.   8                 1 1 1

t =   8
i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
---------------------------------------------------------
  8     37519.   8                 1 1 1

t =   9
i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
---------------------------------------------------------
  8     34436.   8                 1 1 1

t =  10
i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
---------------------------------------------------------
  8     31502.   8                 1 1 1
```

t = 11

| i(t) | E(PV) | i(t+1) | DEC | CVIA | Entering Partial States |
|---|---|---|---|---|---|
| 8 | 28712. | 8 | | | 1 1 1 |

t = 12

| i(t) | E(PV) | i(t+1) | DEC | CVIA | Entering Partial States |
|---|---|---|---|---|---|
| 8 | 26058. | 8 | | | 1 1 1 |

t = 13

| i(t) | E(PV) | i(t+1) | DEC | CVIA | Entering Partial States |
|---|---|---|---|---|---|
| 8 | 23534. | 8 | | | 1 1 1 |

t = 14

| i(t) | E(PV) | i(t+1) | DEC | CVIA | Entering Partial States |
|---|---|---|---|---|---|
| 8 | 21133. | 8 | | | 1 1 1 |

t = 15

| i(t) | E(PV) | i(t+1) | DEC | CVIA | Entering Partial States |
|---|---|---|---|---|---|
| 8 | 18848. | 8 | | | 1 1 1 |

t = 16

| i(t) | E(PV) | i(t+1) | DEC | CVIA | Entering Partial States |
|---|---|---|---|---|---|
| 8 | 16675. | 8 | | | 1 1 1 |

```
t =  17
i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
---------------------------------------------------------
  8     14608.   8                 1 1 1

t =  18
i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
---------------------------------------------------------
  8     12642.   8                 1 1 1

t =  19
i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
---------------------------------------------------------
  8     10772.   8                 1 1 1

t =  20
i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
---------------------------------------------------------
  8      8993.   8                 1 1 1

t =  21
i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
---------------------------------------------------------
  8      7301.   8                 1 1 1

t =  22
i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
---------------------------------------------------------
  8      5691.   8                 1 1 1
```

```
t =  23
i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
--------------------------------------------------------
  8      4160.   8                1 1 1

t =  24
i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
--------------------------------------------------------
  8      2703.   8                1 1 1

t =  25
i(t)  E(PV)   i(t+1)  DEC  CVIA  Entering Partial States
--------------------------------------------------------
  8      1318.   8                1 1 1
```

Figure 2. illustrates the optimal expansion plan.

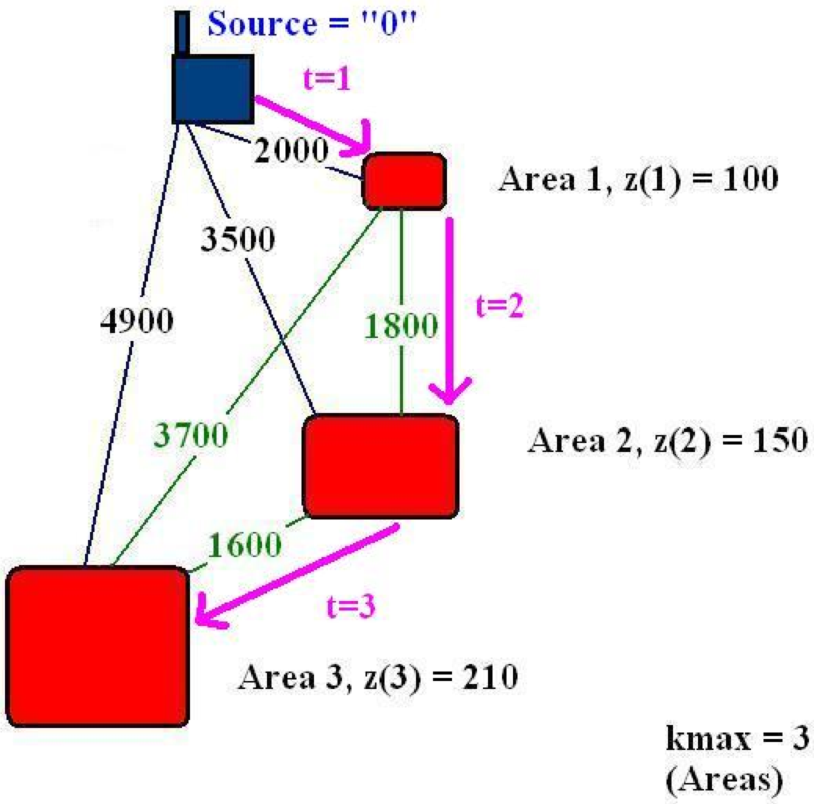


**Figure 2.**
The optimal dynamic investment plan.
Compare the results reported in the output file.

## 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
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
```

---

**Peter Lohmander**
**2010-08-13**