THERMOECONOMIC OPTIMIZATION OF A HEAT PUMP SYSTEM - COMPUTER PROGRAMS

Göran Wall
Physical Resource Theory Group, Chalmers University of Technology and University of Göteborg, S-412 96 Göteborg, Sweden

September, 1985

Abstract - This is a presentation of two programs and calculated results that are used in the Paper *Thermoeconomic Optimization of a Heat Pump System*. The programs are written in Pascal. The first program, consisting of two sub-programs, COSTEQ.PAS and COSTIN.LIS, is only used to generate the diagram in Fig. 3 in the main paper. The second program contains first a documentation, HPMIN.DOK, of the used files and parameters followed by the main program, HPMIN.PAS, which also describes the operation of the program. First it calculates the thermodynamic data for the assumed refrigerant in the sub-programs R12.PAS and PROP.PAS. (Other refrigerants may also be used.) The state equations for the system are presented in HPCYCLE.PAS. These equations are formulated so that iterations are avoided. The minimizing procedure in MIN.PAS is done through a small number of iterations. When the sum of the marginal prices reaches a predefined value the optimization is completed and the result is printed. The values of the fixed decision variables are found in INFILE.LIS. Calculated values for the assumed system and the optimal system are also included.

Note - The programs are also available on disk (Macintosh™) by sending an empty disk to the auther.

Contents

Program in Pascal for calculating the costing relations:

Program in Pascal for optimizing the heat pump system:

Printout of values for the assumed system and related optimal values:

COSTEQ.PAS

```
{Costing equations for each zone}

PROGRAM COSTEQ (COSTIN,COSTOUT);

VAR i,steps:integer;
    etastart,etastop,etastep,V2,p2,p3,mwh,mwc,mr,El,eta:double;
    TZ,cpc:ARRAY[1..5] OF double;
    COSTIN,COSTOUT:text;

BEGIN
  rewrite(COSTOUT);
  reset(COSTIN);
  readln(COSTIN,etastart);
  readln(COSTIN,etastop);
  readln(COSTIN,steps);
  readln(COSTIN,V2);
  readln(COSTIN,p2);
  readln(COSTIN,p3);
  readln(COSTIN,mwh);
  readln(COSTIN,mwc);
  readln(COSTIN,mr);
  readln(COSTIN,El);
  readln(COSTIN,cpc[1]);
  readln(COSTIN,cpc[2]);
  readln(COSTIN,cpc[3]);
  readln(COSTIN,cpc[4]);
  readln(COSTIN,cpc[5]);

  etastep:=(etastop-etastart)/steps;

  FOR i:=0 TO steps DO
  BEGIN
    eta:=etastart+i*etastep;

    TZ[1]:=cpc[1]*V2/(0.9-eta)*p3/p2*ln(p3/p2);
     IF TZ[1]<0 THEN TZ[1]:=0;
    TZ[2]:=cpc[2]*mwh*sqrt(eta/(1-eta));
    TZ[3]:=cpc[3]*mr;
    TZ[4]:=cpc[4]*mwc*sqrt(eta/(1-eta));
    TZ[5]:=cpc[5]*El*eta/(1-eta);

    write(COSTOUT,eta:6:3);
    write(COSTOUT,TZ[1]:10:2);
    write(COSTOUT,TZ[2]:10:2);
    write(COSTOUT,TZ[3]:10:2);
    write(COSTOUT,TZ[4]:10:2);
    write(COSTOUT,TZ[5]:10:2);
    writeln(COSTOUT);
  END;
END.

{**************************************************************}
```

INFILE.LIS

```
0.5        etastart
0.99       etastop
10         steps
2.5E-3     V2, m3/s
3.5        p2, bar
19         p3,bar
0.16       mwh, kg/s in condenser
0.2        mwc, kg/s in evaporator
0.05       mr, kg/s of refrigerant
2500       El, W
7E3        cpc[1] in SEK per flow in m3/s, cost per capacity for compressor
5E3        cpc[2] in SEK per flow in kg/s
5E3        cpc[3] in SEK per flow in kg/s
5E3        cpc[4] in SEK per flow in kg/s
0.1        cpc[5] in SEK/W
```

HPMIN.DOK

{841210 - 850628}

List of files:

CHARTFILE	Data for Microsoft Chart™
CHECK	List of steps in optimization (convergens, etc)
DECVAR	Variable decision variables, renewed after every run
INFILE	Fix decision variables easy to change
MIN	Result: Dec. var. & costs at optimum
OUTFILE	Result: Start values & optimal values
R12	Data for refrigerant R12
SENS	Sensitivity in optimal values

List of parameters:

an[1..5]	Annuity coefficient for each zone
Ar[1..5]	Constants for refrigerant
answer	Character for question: rerun?
b	Constant for refrigerant
Br[1..5]	Constants for refrigerant
calc	Program cycle variable 1,2,...,10
cpc[1..5]	Cost per capacity for each zone
cpr[1..5]	Constants for heat capacity for refrigerant
cprl	Specific heat of refrigerant as liquid
cpw	Specific heat of water
Cr[1..5]	Constants for refrigerant
dpdT	dp/dT
dpdT1	dp/dT1
dpdT4	dp/dT4
Dr[1..7]	constants for refrigerant
dTsc	Supercooling temperature in condenser
dTsh	Superheating temperature in evaporator
D[1..4]	Derivate of each decision variable
dy[1..4]	Step in change of each decision variable
e[1..11]	Specific exergy of each flow, J/kg
EE[1..11]	Exergy of each flow, W
EEmin	Exergy min value, help variable in HPEXERGY.PAS
El	Electricity, W
El0	Electricity, last calculated value, W
eta[1..5]	Efficiency of each zone, fraction
etamax[1..5]	Maximum efficiency of each zone, fraction
etamin[1..5]	Minimum efficiency of each zone, fraction
Fr[1..5]	Constants for refrigerant
f[1..4]	Function for each decision variable
Gr[1..5]	Constants for refrigerant
h[1..11]	Specific enthalpy of each flow, J/kg
HH[1..11]	Enthalpy of each flow, W
hlg	Enthalpy change from liquid to gas of refrigerant
Hout	Enthalpy output as heat from the heat pump
hrs	Numbers of hour in use per year
h3rev	Enthalpy of reversible change at 3
h[1..11]	Specific enthalpy of each flow, J/ks
i	Counter
infile	Text, file for input data
inter [1..5]	Interest factor for each zone
k	Counter for actual decision variable
kappa	
LW[1..5]	Lost Work for each zone, W
LW0[1..5]	Lost Work for each zone, last calculated value, W
LWoptimum[1..5]	Lost Work for each zone at optimum, W
LWTadd[1..4]	Change in total lost work when small increase in decision variable
LWToptimum	Lost Work Total, W
LWTsub[1..4]	Change in total lost work when small decrease in decision variable
M	Molecular weight of refrigerant
MC[1..5]	Marginal Cost of lost work in each zone, SEK/J
MCEl	Marginal Cost of Electricity, SEK/J
MCT	Marginal Cost of lost work tot
mr	Mass flow of refrigerant, kg/s
mwc	Mass flow of water, cold side, kg/s

mwh	Mass flow of water, hot side, kg/s
NTUc	Number of Heat Transfer Units, cold side
NTUh	Number of Heat Transfer Units, hot side
outfile	Text, file for writing results
p[1..11]	Pressure of each flow, Pascal
pc	Critical pressure of refrigerant
pel	Price of electricity, SEK/J
psat	Saturation pressure of refrigerant, Pa
p0	Pressure of the environment, Pa
p[1..11]	Pressure of each flow, Pa
R	Gas constant of refrigerant, J/kg K
RCEl	Ratio cost of electricity, SEK/J
RT	Ratio cost of lost work total, SEK/J
RC[1..5]	Ratio cost of of lost work of each zone, SEK/J
roc	Critical density of refrigerant
rowc	Density of water, cold side, kg/m3
rowh	Density of water, hot side, kg/m3
R12	Text, data for refrigerant 12
s[1..11]	Specific entropy of each flow, J/K/kg
slg	Entropy(gas)-entropy(liquid)
SS[1..11]	Entropy of each flow, W/K
s0r	Zero entropy level of refrigerant
T[1..11]	Temperature of each flow, K
Tc	Critical temperature of refrigerant
TCadd[1..4]	Change in total cost when small increase of decision variable
TCmin	Total Cost minimum value, SEK
TCost	Total Cost, SEK
TCsub[1..4]	Change in total cost when small decrease of decision variable
TC0	Total Cost last calculated value, SEK
TLW	Total Lost Work
TLW0	Total Lost Work, last calculated value
TLWadd	Total Lost Work when small increase in dec. var.
TLWsub	Total Lost Work when small decrease in dec. var.
TLWoptimum	Total Lost Work at optimum, W
T0	Temperature of environment, K
T0r	Zero temperature level of refrigerant
TK	Constant: 273.15, relation between Celsius and Kelvin
Tlmc	Logarithmic mean temperature difference, cold side, K
Tlmh	Logarithmic mean temperature difference, hot side, K
TZ[1..5]	Total cost of each zone, costing functions
u[1..11]	Specific energy of each flow, J/kg
UAc	Overall thermal conductance, cold side, W/K
UAh	Overall thermal conductance, hot side, W/K
UU[1..11]	Energy of each flow, W
ulg	Energy change from gas to liquid of refrigerant
u0r	Zero energy level of refrigerant
v[1..11]	Specifik volume of each flow, m3/kg
vc	Critical specific volume of refrigerant
VV[1..11]	Volume of each flow, m3
x	Gas fraction of gas-liquid mixture of refrigerant
y[1..4]	Actual decision variable
yadd[1..4]	Decision variable + 1%
year[1..5]	Economic life time for each zone
ymax[1..4]	Maximum value for each decision variable
ymin[1..4]	Minimum value for each decision variable
yoptimum[1..4]	Optimum value for each decision variable
ystart[1..4]	Start value for each decision variable
ysub[1..4]	Decision variable - 1%
Z[1..5]	Cost per year for each zone, SEK/yr
Z0[1..5]	Last cost per year for each zone, SEK/yr
Zoptimum[1..5]	Cost per year for each zone at optimum, SEK/yr

HPMIN.PAS

```
PROGRAM HPMIN(output,CHARTFILE,CHECK,DECVAR,INFILE,MIN,OUTFILE,R12,SENS);
{850627}
LABEL 9999;
CONST tab5='     ';                    {5 blanks}
      tab10='          ';              {10 blanks}

VAR k,calc:integer;
    answer:char;
    CHARTFILE,CHECK,DECVAR,INFILE,MIN,OUTFILE,R12,SENS:text;
    T0,p0,cpw,TK,M,R,Tc,pc,roc,vc,kappa,b,T0r,u0r,s0r,dpdT,slg,
    hlg,ulg,psat,dTsh,dTsc,dpdT1,dpdT4,h3rev,x,Hout,hrs,mr,mwh,
    mwc,rowh,rowc,El,cprl,NTUc,NTUh,Tlmc,Tlmh,UAc,UAh,TCost,TC0,
    RCT,MCT,RCEl,MCEl,TCmin,pel,TLW,TLW0,TLWoptimum,El0,step:double;
    ystart,y,yoptimum,dy,D,ymin,ymax,f,yadd,ysub,TCadd,TCsub,
    TLWadd,TLWsub:ARRAY[1..4] OF double;
    cpc,eta,etamin,etamax,Z,Z0,Zoptimum,TZ,LW,LW0,LWoptimum,RC,
    MC,an,inter,year:ARRAY[1..5] OF double;
    T,p,v,u,h,s,e,VV,UU,HH,SS,EE:ARRAY[1..11] OF double;

FUNCTION pwr(x,y:double):double;
  BEGIN pwr:=exp(y*ln(x)); END;

FUNCTION sgn(x:double):double;
  BEGIN sgn:=x/abs(x); END;

%INCLUDE 'R12.PAS/nolist'
%INCLUDE 'PROP.PAS/nolist'
%INCLUDE 'HPINIT.PAS/nolist'
%INCLUDE 'HPCYCLE.PAS/nolist'
%INCLUDE 'HPCOST.PAS/nolist'
%INCLUDE 'HPEXERGY.PAS/nolist'
%INCLUDE 'MIN.PAS/nolist'
%INCLUDE 'HPPRINT.PAS/nolist'
%INCLUDE 'HPCHART.PAS/nolist'

BEGIN
  Refrigerant;
  rewrite(CHARTFILE);
  rewrite(CHECK);
  rewrite(MIN);
  rewrite(OUTFILE);
  rewrite(SENS);
  Init;
  Printstart;
{FOR calc:=0 TO 9 DO}
  BEGIN
    writeln(tab5,'Run # ',calc:1);
    Init;
    Minimum;
    Printoptimum;
    Printmin;
    Chart;
  END;
9999:
END.
```

R12.PAS

```
{ref. Reynolds, W. C.,Thermodynamic properties in SI, Departmant of
 Mechanical Engineering, Stanford University, Stanford, CA 94305, 1979}
{************** Property procedures for Freon-12 ******************}

VAR Gr,Ar,Br,Cr,Fr,cpr:ARRAY[1..5] OF     {Refrigerant constants}
      double;
      Dr:ARRAY[1..7] OF double;           {Refrigerant constants}

PROCEDURE Refrigerant;

VAR i,j:integer;

BEGIN
  reset(R12);
  M:=120.93;
  R:=68.7480;
  FOR i:=2 TO 5 DO readln(R12,Ar[i]);
  FOR i:=2 TO 5 DO readln(R12,Br[i]);
  FOR i:=2 TO 5 DO readln(R12,Cr[i]);
  FOR i:=1 TO 4 DO readln(R12,Gr[i]);
  FOR i:=1 TO 4 DO readln(R12,Fr[i]);
  FOR i:=1 TO 7 DO readln(R12,Dr[i]);
  Tc:=385.17;
  pc:=4.1159;
  roc:=588.08;
  vc:=1/roc;
  T0r:=200;
  b:=4.06366926D-4;
  kappa:=5.475;
  u0r:=1.6970187D5;
  s0r:=8.9448764D2;
END;

{*********************************************************}

FUNCTION pr(T,v:double):double;

VAR i:integer;
    p:double;

BEGIN
  p:=R*T/(v-b);
  FOR i:=2 TO 5 DO
  BEGIN
     p:=p+(Ar[i]+Br[i]*T+Cr[i]*exp(-kappa*T/Tc))/pwr(v-b,i);
  END;
  pr:=p;
END;

{*****************************************************}

FUNCTION ur(T,v:double):double;

VAR i:integer;
  u:double;

BEGIN
  u:=u0r-Gr[5]*(1/T-1/T0r);
  FOR i:=1 TO 4 DO u:=u+Gr[i]*(pwr(T,i)-pwr(T0r,i))/i;
  FOR i:=2 TO 5 DO u:=u+(Ar[i]+Cr[i]*exp(-kappa*T/Tc)*(kappa*T/Tc+1))/((i-1)*pwr(v-b,i-1));
  ur:=u;
END;

{************************************************}

FUNCTION sr(T,v:double):double;

VAR i:integer;
    s:double;
```

```
BEGIN
  s:=s0r+Gr[1]*ln(T/T0r)-Gr[5]*(1/(T*T)-1/(T0r*T0r))/2;
  s:=s+R*ln(v-b);
  FOR i:=2 TO 4 DO s:=s+Gr[i]*(pwr(T,i-1)-pwr(T0r,i-1))/(i-1);
  FOR i:=2 TO 5 DO s:=s-(Br[i]-Cr[i]*kappa*exp(-kappa*T/Tc)/Tc)/((i-1)*pwr(v-b,i-1));
  sr:=s;
END;

{************************************************************}

FUNCTION vsl(T:double):double;

VAR x,help:double;
    i:integer;

BEGIN
  x:=1-T/Tc;
  help:=Dr[6]*pwr(x,0.5)+Dr[7]*x*x;
  FOR i:=1 TO 4 DO help:=help+Dr[i]*pwr(x,(i-1)/3);
  vsl:=1/help;
END;

{***********************************************************}

{ref. Thermophysical Properties of Matter vol. 6, Purdue Univ.}

FUNCTION cpsl(T:double):double;

BEGIN
  cpsl:=40.296648+9.7120682*T-4.070781D-2*T*T+6.2564189D-5*T*T*T;
END;

{***********************************************************}

FUNCTION satpr(T:double):double;

BEGIN
  satpr:=exp(Fr[1]+Fr[2]/T+Fr[3]*ln(T)+Fr[4]*T);
END;

FUNCTION satdpdTr(T,p:double):double;

BEGIN
  satdpdTr:=(-Fr[2]/(T*T)+Fr[3]/T+Fr[4])*p;
END;

{**********************************************************}

PROCEDURE lr(T1,T2:double;VAR dh,ds:double);

VAR i:integer;

BEGIN
  dh:=0;
  FOR i:=1 TO 4 DO dh:=dh+cpr[i]*(pwr(T2,i)-pwr(T1,i))/i;
  ds:=cpr[1]*ln(T2/T1);
  FOR i:=2 TO 4 DO ds:=ds+cpr[i]*(pwr(T2,i-1)-pwr(T1,i-1))/(i-1);
END;

{***********************************************************}
FUNCTION vstart(T:double):double;      {Calculation of a start value for v}
BEGIN
  vstart:=-5.8529E-4*T+0.22025;
END;

{***********************************************************}
```

R12.LIS

```
 -9.16210126d1         Ar[2]
  1.01049598d-1        Ar[3]
 -5.74640225d-5        Ar[4]
  0                    Ar[5]
  7.71136428d-2        Br[2]
 -5.67539138d-5        Br[3]
  0                    Br[4]
  4.08193371d-11       Br[5]
 -1.52524293d3         Cr[2]
  2.19982681           Cr[3]
  0                    Cr[4]
 -1.66307226d-7        Cr[5]
  3.389005260d1        Gr[1]
  2.507020671          Gr[2]
 -3.274505926d-3       Gr[3]
  1.641736815d-6       Gr[4]
  9.33438056d1         Fr[1]
 -4.39618785d3         Fr[2]
 -1.24715223d1         Fr[3]
  1.96060432d-2        Fr[4]
  5.580845400d2        Dr[1]
  8.544458040d2        Dr[2]
  0                    Dr[3]
  2.994077103d2        Dr[4]
  0                    Dr[5]
  3.521500633d2        Dr[6]
 -5.047419739d1        Dr[7]
40.296648              cpslr[1]
  9.7120682            cpslr[2]
 -4.070781d-2          cpslr[3]
  6.2564289d-5         cpslr[4]
  1.126E3              cpslm
```

PROP.PAS

```
{PROCEDURE Property}
{ref. Reynolds, W. C.,Thermodynamic properties in SI, Departmant of
 Mechanical Engineering, Stanford University, Stanford, CA 94305, 1979}
{***************************************************************}

PROCEDURE Property(VAR T,p,v,u,h,s,vc,Tc,pc,R:double;option:integer);

LABEL 9999;

CONST erh=0.00001;
     erp=0.0001;
     ers=0.0001;
     maxiter=50;

TYPE choice=(TT,vv,pp,ss,hh);

VAR
dT,dv,dvbf,pmin,pmax,dvs1,dvs2,v1,v2,T1,T2,vmin,vmax,px,ux,hx,sx,p1,u1,h1,s1,dpdv,dvm,dTm,vt
,dva,dTa, p2,u2,h2,s2,det,dhdT,dhdv,dpdT,dsdT,dsdv:double;
    kbr,count:integer;
    flag,error:boolean;

{PROCEDURE perturb}
{              alter_v}
{              regulate}
{              init}
{FUNCTION no_convergence}
{***************************************************************}


PROCEDURE perturb(op:choice);

BEGIN
  CASE op OF
   TT:BEGIN
     dT:=0.001*T;
     T1:=T+dT;
     v1:=v;
    END;
    vv:BEGIN
     dv:=0.001*v;
     IF v<=vc THEN dv:=-dv;
     v2:=v+dv;
     T2:=T;
    END;
  END;  {CASE}
END;

PROCEDURE alter_v;

BEGIN
  IF v<=vc THEN dv:=-0.05*v
  ELSE dv:=0.2*v;
  IF vmin>0 THEN dv:=0.2*v;
  IF vmax<1d30 THEN dv:=-0.05*v;
END;

PROCEDURE regulate;

BEGIN
  dvm:=0.2*v;
  IF v<dvs1 THEN dvm:=0.5*dvm;
  IF v<dvs2 THEN dvm:=0.5*dvm;
  dTm:=0.1*T;
  dva:=abs(dv);
  dTa:=abs(dT);
  IF dva>dvm THEN dv:=dv*dvm/dva;
  IF dTa>dTm THEN dT:=dT* dTm/dTa;
END;
```

```
PROCEDURE init;

BEGIN
  dT:=0;
  kbr:=0;
  dvbf:=1;
  vmin:=0;
  vmax:=1d30;
  pmin:=0;
  pmax:=1d30;
  dvs1:=2*vc;
  dvs2:=0.7*vc;
  flag:=false;
  error:=false;
END;

FUNCTION no_convergence(op:choice):boolean;

BEGIN
  CASE op OF
    pp:no_convergence:=abs(p-px)>=erp*p;
    ss:no_convergence:=abs(s-sx)>=ers*R;
    hh:no_convergence:=abs(h-hx)>=erh*R*T;
  END;
END;

{****************************************}

BEGIN
  init;
  count:=1;
  CASE option OF

    1:p:=pr(T,v);                {T & v known find p}

    2:BEGIN                      {T & p known find v}
      px:=0;
      WHILE no_convergence(pp) AND (count<=maxiter) DO
      BEGIN
        px:=pr(T,v);
        IF px>=0 THEN
        BEGIN
          perturb(vv);
          p2:=pr(T2,v2);
          dpdv:=(p2-px)/dv;
          IF dpdv>0 THEN flag:=true
          ELSE
          BEGIN
            IF (px>p) AND (v>vmin) THEN
            BEGIN
              vmin:=v;
              pmin:=px;
            END;
            IF (px<p) AND (v<vmax) THEN
            BEGIN
              vmax:=v;
              pmax:=px;
            END;
            IF vmin>=vmax THEN {ERROR!} GOTO 9999;
            IF (vmin>0) AND (vmax<1d30) THEN kbr:=1;
            IF dpdv=0 THEN
            BEGIN
              dvbf:=0.5;
              flag:=true;
            END
            ELSE
            BEGIN
              dvbf:=1;
              dv:=(p-px)/dpdv;
              dT:=0;
            END;
```

```
     END;
   END;
   IF (px<0) OR flag THEN
   BEGIN
     IF kbr=0 THEN alter_v
     ELSE
     BEGIN
       dpdv:=(pmax-pmin)/(vmax-vmin);
       v:=vmax;
       px:=pmax;
       dv:=dvbf*(p-px)/dpdv;
       dT:=0;
       dvbf:=0.5*dvbf;
     END;
   END;
   dvm:=0.2*v;
   IF v<dvs1 THEN dvm:=0.5*dvm;
   IF v<dvs2 THEN dvm:=0.5*dvm;
   dTm:=0.1*T;
   IF kbr<>0 THEN
   BEGIN
     vt:=v+dv;
     IF NOT ((vt>=vmin) AND (vt<=vmax)) THEN dv:=vmin+(p-pmin)*(vmax-vmin)/(pmax-pmin)-v;
   END;
   dva:=abs(dv);
   dTa:=abs(dT);
   IF dva>dvm THEN dv:=dv*dvm/dva;
   IF dTa>dTm THEN dT:=dT*dTm/dTa;
   T:=T+dT;
   v:=v+dv;
   count:=count+1;
  END; {while}
  u:=ur(T,v);
  h:=u+p*v;
  s:=sr(T,v);
END;

3:BEGIN                    {p & v known}
  px:=pr(T,v);
  WHILE no_convergence(pp) AND (count<=maxiter) DO
  BEGIN
   IF px<0 THEN alter_v
   ELSE
   BEGIN
     perturb(TT);
     p1:=pr(T1,v1);
     dpdT:=(p1-px)/dT;
     dT:=(p-px)/dpdT;
     dv:=0;
   END;
   regulate;
   T:=T+dT;
   v:=v+dv;
   count:=count+1;
   px:=pr(T,v);
  END; {while}
  u:=ur(T,v);
  h:=u+p*v;
  s:=sr(T,v);
END;

4:BEGIN               {v & h known}
  px:=pr(T,v);
  ux:=ur(T,v);
  hx:=ux+px*v;
  WHILE no_convergence(hh) AND (count<=maxiter) DO
  BEGIN
   IF px<0 THEN alter_v
   ELSE
   BEGIN
     perturb(TT);
     p1:=pr(T1,v1);
```

```
      u1:=ur(T1,v1);
      h1:=u1+p1*v1;
      dhdT:=(h1-hx)/dT;
      dT:=(h-hx)/dhdT;
      dv:=0;
    END;
    regulate;
    T:=T+dT;
    v:=v+dv;
    count:=count+1;
    px:=pr(T,v);
    ux:=ur(T,v);
    hx:=ux+px*v;
  END; {while}
  p:=px;
  u:=ux;
  h:=hx;
  s:=sr(T,v);
END;

5:BEGIN              {T & h known}
  px:=pr(T,v);
  ux:=ur(T,v);
  hx:=ux+px*v;
  WHILE no_convergence(hh) AND (count<=maxiter) DO
  BEGIN
    IF px<0 THEN alter_v
    ELSE
    BEGIN
      perturb(vv);
      p2:=pr(T2,v2);
      u2:=ur(T,v);
      h2:=u2+p2*v2;
      dhdv:=(h2-hx)/dv;
      dv:=(h-hx)/dhdv;
      dT:=0;
    END;
    regulate;
    T:=T+dT;
    v:=v+dv;
    count:=count+1;
    px:=pr(T,v);
    ux:=ur(T,v);
    hx:=ux+px*v;
  END; {while}
  p:=px;
  u:=ux;
  h:=hx;
  s:=sr(T,v);
END;

6:BEGIN              {v & s known}
  px:=pr(T,v);
  sx:=sr(T,v);
  WHILE no_convergence(ss) AND (count<=maxiter) DO
  BEGIN
    IF px<0 THEN alter_v
    ELSE
    BEGIN
      perturb(TT);
      p1:=pr(T1,v1);
      s1:=sr(T1,v1);
      dsdT:=(s1-sx)/dT;
      dT:=(s-sx)/dsdT;
      dv:=0;
    END;
    regulate;
    T:=T+dT;
    v:=v+dv;
    count:=count+1;
    px:=pr(T,v);
    sx:=sr(T,v);
```

```
  END; {while}
  p:=px;
  u:=ur(T,v);
  h:=u+p*v;
  s:=sx;
END;

7:BEGIN         {T & s known}
  px:=pr(T,v);
  sx:=sr(T,v);
  WHILE no_convergence(ss) AND (count<=maxiter) DO
  BEGIN
    IF px<0 THEN alter_v
    ELSE
    BEGIN
      perturb(vv);
      p2:=pr(T2,v2);
      s2:=sr(T,v);
      dsdv:=(s2-sx)/dv;
      dv:=(s-sx)/dsdv;
      dT:=0;
    END;
    regulate;
    T:=T+dT;
    v:=v+dv;
    count:=count+1;
    px:=pr(T,v);
    sx:=sr(T,v);
  END; {while}
  p:=px;
  u:=ur(T,v);
  h:=u+p*v;
END;

8:BEGIN           {p & s known}
  px:=pr(T,v);
  sx:=sr(T,v);
  WHILE (no_convergence(ss) OR no_convergence(pp)) AND
    (count<=maxiter) DO
  BEGIN
    IF px<0 THEN alter_v
    ELSE
    BEGIN
      perturb(TT);
      p1:=pr(T1,v1);
      s1:=sr(T1,v1);
      perturb(vv);
      p2:=pr(T2,v2);
      s2:=sr(T,v);
      dsdT:=(s1-sx)/dT;
      dsdv:=(s2-sx)/dv;
      dpdT:=(p1-px)/dT;
      dpdv:=(p2-px)/dv;
      det:=dsdT*dpdv-dpdT*dsdv;
      dT:=((s-sx)*dpdv-(p-px)*dsdv)/det;
      dv:=(dsdT*(p-px)-dpdT*(s-sx))/det;
    END;
    regulate;
    T:=T+dT;
    v:=v+dv;
    count:=count+1;
    px:=pr(T,v);
    sx:=sr(T,v);
  END; {while}
  u:=ur(T,v);
  h:=u+p*v;
END;

9:BEGIN          {p & h known}
  px:=pr(T,v);
  ux:=ur(T,v);
  hx:=ux+px*v;
```

```
  WHILE (no_convergence(hh) OR no_convergence(pp)) AND
   (count<=maxiter) DO
  BEGIN
   IF px<0 THEN alter_v
   ELSE
   BEGIN
     perturb(TT);
     p1:=pr(T1,v1);
     u1:=ur(T1,v1);
     h1:=u1+p1*v1;
     perturb(vv);
     p2:=pr(T2,v2);
     u2:=ur(T,v);
     h2:=u2+p2*v2;
     dhdT:=(h1-hx)/dT;
     dhdv:=(h2-hx)/dv;
     dpdT:=(p1-px)/dT;
     dpdv:=(p2-px)/dv;
     det:=dhdT*dpdv-dpdT*dhdv;
     dT:=((h-hx)*dpdv-(p-px)*dhdv)/det;
     dv:=(dhdT*(p-px)-dpdT*(h-hx))/det;
   END;
   regulate;
   T:=T+dT;
   v:=v+dv;
   count:=count+1;
   px:=pr(T,v);
   ux:=ur(T,v);
   hx:=ux+px*v;
  END; {while}
  s:=sr(T,v);
END;

10:BEGIN              {h & s known}
  px:=pr(T,v);
  ux:=ur(T,v);
  sx:=sr(T,v);
  hx:=ux+px*v;
  WHILE (no_convergence(ss) OR no_convergence(hh)) AND
   (count<=maxiter) DO
   BEGIN
    IF px<0 THEN alter_v
    ELSE
    BEGIN
      perturb(TT);
      p1:=pr(T1,v1);
      u1:=ur(T1,v1);
      s1:=sr(T1,v1);
      h1:=u1+p1*v1;
      perturb(vv);
      p2:=pr(T2,v2);
      u2:=ur(T,v);
      s2:=sr(T,v);
      h2:=u2+p2*v2;
      dsdT:=(s1-sx)/dT;
      dsdv:=(s2-sx)/dv;
      dhdT:=(h1-hx)/dT;
      dhdv:=(h2-hx)/dT;
      det:=dhdT*dsdv-dsdT*dhdv;
      dT:=((h-hx)*dsdv-(s-sx)*dhdv)/det;
      dv:=(dhdT*(s-sx)-dsdT*(h-hx))/det;
    END;
    regulate;
    T:=T+dT;
    v:=v+dv;
    count:=count+1;
    px:=pr(T,v);
    ux:=ur(T,v);
    sx:=sr(T,v);
    hx:=ux+px*v;
   END; {while}
   u:=ux;
```

```
      p:=px;
     END;
   END; {CASE}

  IF count>maxiter THEN
  BEGIN
    writeln('Property not convergent for option ',option:0);
    writeln('T = ',T);
    writeln('v = ',v);
    writeln('p = ',p);
    writeln('u = ',u);
    writeln('h = ',h);
    writeln('s = ',s);
  END;

  9999:IF (option=2) AND (vmin>=vmax) THEN      {ERROR MESSAGE}
  BEGIN
    writeln('Error in property');
    writeln('T = ',T);
    writeln('p = ',p);
    writeln('v = ',v);
    writeln('vmin = ',vmin);
    writeln('vmax = ',vmax);
  END;
END;
{****************************************************************}

PROCEDURE sat(VAR T,p,dpdT,vc,Tc,pc,R:double;option:integer);

CONST maxiter=150;

VAR err,px,dTa,dTm,dT:double;
    count:integer;

BEGIN
  CASE option OF

    1:BEGIN
      p:=satpr(T);
      dpdT:=satdpdTr(T,p);
    END;

    2:BEGIN
      count:=0;
      err:=1d-6*p;
      IF T>Tc THEN T:=Tc-0.001;
      px:=satpr(T);
      dpdT:=satdpdTr(T,px);
      WHILE (abs(p-px)>=err) AND (count<=maxiter) DO
      BEGIN
        dT:=(p-px)/dpdT;
        dTa:=abs(dT);
        dTm:=0.1*dT;
        IF dTa>dTm THEN dT:=dT*dTm/dTa;
        T:=T+dT;
        IF T>Tc THEN T:=Tc-0.001;
        px:=satpr(T);
        dpdT:=satdpdTr(T,px);
        count:=count+1;
      END;
      IF count>maxiter THEN writeln('Sat not convergent : p,T=',p,T);
    END;
  END;        {CASE}
END;
{****************************************************************}
```

HPINIT.PAS

```
{PROCEDURE Init}
{************************** Initiation ****************************}

PROCEDURE Init;

VAR i:integer;

BEGIN
 TK:=273.15;                        {Relation between degr C and K}
 T0:=273.15;                        {0 degrees C}
 p0:=101325;                        {1 atm}
 cpw:=4180;                         {cp water J/kg*K}
 rowh:=983.2;                       {Density for water at 60 degr C}
 rowc:=1000;                        {Density for water at 5 degr C}

 {Fixed decision variables}

 reset(INFILE);
 readln(INFILE,step);               {Step between each optimization}
 readln(INFILE,T[8]);               {Water in hot side in degr C}
 readln(INFILE,T[9]);               {Water out hot side in degr C}
 readln(INFILE,T[10]);              {Water in cold side in degr C}
 readln(INFILE,T[11]);              {Water out cold side in degr C}
 FOR i:=8 TO 11 DO T[i]:=T[i]+TK;   {From degr C to K}
 FOR i:=8 TO 11 DO p[i]:=p0;        {pressure for water}
 readln(INFILE,dTsh);               {Superheating in evaporator}
 readln(INFILE,dTsc);               {Supercooling in condenser}
 readln(INFILE,Hout);               {Heat out}
 readln(INFILE,hrs);                {Hours in action per year}
 readln(INFILE,etamin[1]);          {Minimum efficiency for compressor}
 readln(INFILE,etamin[2]);          {Minimum efficiency for condenser}
 readln(INFILE,etamin[4]);          {Minimum efficiency for evaporator}
 readln(INFILE,etamin[5]);          {Minimum efficiency for electric eng.}
 readln(INFILE,etamax[1]);          {Maximum efficiency for compressor}
 readln(INFILE,etamax[2]);          {Maximum efficiency for condenser}
 readln(INFILE,etamax[4]);          {Maximum efficiency for evaporator}
 readln(INFILE,etamax[5]);          {Maximum efficiency for electric eng.}

 FOR i:=1 TO 5 DO
 BEGIN
  readln(INFILE,inter[i]);          {Interest factor for each zone}
  readln(INFILE,year[i]);           {Life time for each zone}
  an[i]:=inter[i]/(1-pwr((1+inter[i]),-year[i]));         {Annuity for each zone}
  readln(INFILE,cpc[i]);            {1: Compressor: SEK per m3/s}
                                    {2: Condenser: SEK per kg/s}
                                    {3: Expansion valve: SEK per kg/s}
                                    {4: Evaporator: SEK per kg/s}
                                    {5: Electric engine: SEK per W}
 END;

 readln(INFILE,pel);                {In SEK/kWh}

 {Calc routine}

 {pel:=pel+step*calc;               {calc=0,1,2, ... 9}

 {T[8]:=T[8]+step*calc;             {calc=0,1,2, ... 9}
 {T[9]:=T[8]+10;                    {10 degrees differens}

 {hrs:=hrs+step*calc;               {calc=0,1,2, ... 9}

 pel:=pel/(1000*3600);              {From SEK/kWh to  SEK/J}

 {Variable decision variables, start values}

 reset(DECVAR);
 readln(DECVAR,eta[1]);             {Efficiency for compressor}
 readln(DECVAR,eta[2]);             {Efficiency for condenser}
 readln(DECVAR,eta[4]);             {Efficiency for evaporator}
```

```
  readln(DECVAR,eta[5]);                  {Efficiency for electric engine}

  ystart[1]:=eta[1];                      {Set start values for the variable       }
  ystart[2]:=eta[2];                      {decision variables, these values        }
  ystart[3]:=eta[4];                      {are kept constant throught the          }
  ystart[4]:=eta[5];                      {program                        }

  FOR k:=1 TO 4 DO y[k]:=ystart[k];       {Set start values}

  FOR k:=1 TO 4 DO dy[k]:=0.01*ystart[k];                         {Step=0.01*ystart}

  {Limits for variable decision variables}

  ymin[1]:=etamin[1]; ymax[1]:=etamax[1];                         {Compressor}
  ymin[2]:=etamin[2]; ymax[2]:=etamax[2];                         {Condenser}
  ymin[3]:=etamin[4]; ymax[3]:=etamax[4];                         {Evaporator}
  ymin[4]:=etamin[5]; ymax[4]:=etamax[5];                         {Electric engine}

END;

{**************************************************************}
```

INFILE.LIS

```
   0     step
  50     T[8]
  60     T[9]
  10     T[10]
   5     T[11]
  10     dTsh
  10     dTsc
6500     Hout
5000     hrs
   0.4   etamin[1], compressor
   0.4   etamin[2], condenser
   0.4   etamin[4], evaporator
   0.6   etamin[5], electric engine
   0.9   etamax[1], compressor
   0.99  etamax[2], condenser
   0.99  etamax[4], evaporator
   0.99  etamax[5], electric engine
    .10  inter[1], interest factor for compressor
  10     year[1], economic life time for compressor
   7E3   cpc[1] in SEK per flow in m3/s, cost per capacity for compressor
    .10  inter[2], dito for condenser
  15     year[2]
   5E3   cpc[2] in SEK per flow in kg/s
    .10  inter[3], dito for expansion valve
  15     year[3]
   5E3   cpc[3] in SEK per flow in kg/s
    .10  inter[4], dito for evaporator
  15     year[4]
   5E3   cpc[4] in SEK per flow in kg/s
    .10  inter[5], dito for electric engine
  15     year[5]
   0.1   cpc[5] in SEK/W
   0.25  pel in SEK/kWh
```

HPCYCLE.PAS

```
{PROCEDURE Cycle}
{************************* Heat pump cycle *************************}

PROCEDURE Cycle;

VAR i:integer;

BEGIN
  T[1]:=T[10]+(T[11]-T[10])/eta[4];              {Evaporator}
  T[4]:=(T[9]-T[8])/eta[2]+T[8];                 {Condenser}
  IF T[4]>Tc THEN
  BEGIN
    writeln('T4 exceeds Tc !!!');
   T[4]:=1/0;                                    {PANIC STOP}
  END;
  sat(T[1],p[1],dpdT1,vc,Tc,pc,R,1);
  v[1]:=vstart(T[1]);
  property(T[1],p[1],v[1],u[1],h[1],s[1],vc,Tc,pc,R,2);
  p[2]:=p[1];
  T[2]:=T[1]+dTsh;
  v[2]:=vstart(T[2]);
  property(T[2],p[2],v[2],u[2],h[2],s[2],vc,Tc,pc,R,2);
  sat(T[4],p[4],dpdT4,vc,Tc,pc,R,1);
  v[4]:=vstart(T[4]);
  property(T[4],p[4],v[4],u[4],h[4],s[4],vc,Tc,pc,R,2);
  p[3]:=p[4];
  T[3]:=T[4]+dTsh;
  v[3]:=vstart(T[3]);
  property(T[3],p[3],v[3],u[3],h3rev,s[2],vc,Tc,pc,R,8);
  h[3]:=h[2]+(h3rev-h[2])/eta[1];
  property(T[3],p[3],v[3],u[3],h[3],s[3],vc,Tc,pc,R,9);
  IF T[3]>Tc THEN
  BEGIN
    writeln('T3 exceeds Tc!!!!');
   T[3]:=1/0;                                    {PANIC STOP}
  END;
  T[5]:=T[4];
  p[5]:=p[4];
  v[5]:=vsl(T[5]);
  slg:=(v[4]-v[5])*dpdT4;
  hlg:=slg*T[5];
  h[5]:=h[4]-hlg;
  u[5]:=h[5];
  ulg:=u[4]-u[5];
  s[5]:=s[4]-slg;
  T[6]:=T[5]-dTsc;
  p[6]:=p[5];
  v[6]:=v[5];
  h[6]:=h[5]-cpr[5]*dTsc;
  u[6]:=h[6];
  s[6]:=s[5]+cpr[5]*ln(T[6]/T[5]);
  h[7]:=h[6];
  T[7]:=T[1];
  p[7]:=p[1];
  slg:=(v[1]-vsl(T[7]))*dpdT1;
  hlg:=slg*T[7];
  x:=(h[7]-h[1])/hlg+1;
  v[7]:=v[6]*(1-x)+v[1]*x;
  s[7]:=s[1]-(1-x)*slg;
  u[7]:=u[1]-(1-x)*ulg;
  mr:=Hout/(h[3]-h[6]);
  El:=mr*(h[3]-h[2])/eta[5];
  FOR i:=1 TO 7 DO e[i]:=h[i]-T0r*s[i];
END;

{**************************************************************}
```

HPCOST.PAS

```
{PROCEDURE Costs}
{               Margcosts}
{Costing equations for each zone and marginal prices for the decision variables}

PROCEDURE Costs;

VAR i:integer;

BEGIN
  TZ[1]:=cpc[1]*VV[2]/(etamax[1]-eta[1])*p[3]/p[2]*ln(p[3]/p[2]);
  TZ[2]:=cpc[2]*mwh*sqrt(eta[2]/(1-eta[2]));
  TZ[3]:=cpc[3]*mr;
  TZ[4]:=cpc[4]*mwc*sqrt(eta[4]/(1-eta[4]));
  TZ[5]:=cpc[5]*El*eta[5]/(1-eta[5]);

  TCost:=0;
  FOR i:=1 TO 5 DO
  BEGIN
   Z[i]:=an[i]*TZ[i];
   TCost:=TCost+Z[i];
  END;
  TCost:=hrs*3600*pel*El+TCost;
END;


PROCEDURE Margcosts;

VAR i:integer;

BEGIN                                       {Calculation of ratio and marginal costs}
  RCT:=TCost/(TLW*hrs*3600);                {Ratio Cost Total}
  MCT:=(TCost-TC0)/((TLW-TLW0)*hrs*3600);   {Marginal Cost Total}
  FOR i:=1 TO 5 DO
  BEGIN
    RC[i]:=Z[i]/(LW[i]*hrs*3600);           {Ratio Costs}
    MC[i]:=(Z[i]-Z0[i])/((LW[i]-LW0[i])*hrs*3600);          {Marginal Costs}
  END;
  RCEl:=pel;                                {pel in SEK/J}
  MCEl:=(TCost-(Z[1]+Z[2]+Z[3]+Z[4]+Z[5])-
       (TC0-(Z0[1]+Z0[2]+Z0[3]+Z0[4]+Z0[5])))/
      ((El-El0)*hrs*3600);                  {Should be equal RCEl}
END;

{**************************************************************}
```

HPEXERGY.PAS

```
{PROCEDURE Exergies}
{********** Exergies for all flows and lost work in all zones **********}

PROCEDURE Exergies;

VAR i:integer;
   EEmin:double;                              {Exergy min value in cycle}

BEGIN
 FOR i:=1 TO 7 DO
 BEGIN
   VV[i]:=mr*v[i];
   UU[i]:=mr*u[i];
   HH[i]:=mr*h[i];
   SS[i]:=mr*s[i];
   EE[i]:=HH[i]-T0*SS[i];
 END;

 EEmin:=EE[1];
 FOR i:=2 TO 7 DO
 BEGIN
  IF EE[i]<EEmin THEN EEmin:=EE[i];
 END;
 FOR i:=1 TO 7 DO EE[i]:=EE[i]-EEmin;         {Zerro level}

 mwh:=Hout/(cpw*(T[9]-T[8]));                 {Enthalpy cons.}
 mwc:=(HH[2]-HH[7])/(cpw*(T[10]-T[11]));      {Enthalpy cons.}

 FOR i:=8 TO 9 DO                             {Water in condenser, hot side}
 BEGIN
  VV[i]:=mwh/rowh;                            {Volume}
   EE[i]:=mwh*cpw*(T[i]-T0-T0*ln(T[i]/T0));   {Total=thermal exergy}
  HH[i]:=mwh*cpw*(T[i]-T0);                   {Enthalpies}
  UU[i]:=HH[i];                               {Energies, water}
 END;

 FOR i:=10 TO 11 DO                           {Water in evaporator, cold side}
 BEGIN
  VV[i]:=mwc/rowc;                            {Volume}
   EE[i]:=mwc*cpw*(T[i]-T0-T0*ln(T[i]/T0));   {Total=thermal exergy}
  HH[i]:=mwc*cpw*(T[i]-T0);                   {Enthalpies}
  UU[i]:=HH[i];                               {Energies, water}
 END;

 {Lost works in each zone 1-5}

 LW[1]:=EE[2]+El*eta[5]-EE[3];
 LW[2]:=EE[3]+EE[8]-EE[6]-EE[9];
 LW[3]:=EE[6]-EE[7];
 LW[4]:=EE[7]+EE[10]-EE[1]-EE[11];            {See comment below}
 LW[5]:=El*(1-eta[5]);
 TLW:=0;                                      {Reset TLW}
 FOR i:=1 TO 5 DO TLW:=TLW+LW[i];

{The work needed for superheating the refrigerant gas after the evaporator is assumed to be available
from losses in other parts of the cycle}

{Data for condenser and evaporator}

 NTUh:=-ln(1-eta[2]);
 UAh:=mwh*cpw*NTUh;
 Tlmh:=(T[9]-T[8])/(ln((T[4]-T[8])/(T[4]-T[9])));

 NTUc:=-ln(1-eta[4]);
 UAc:=mwc*cpw*NTUc;
 Tlmc:=(T[10]-T[11])/(ln((T[10]-T[7])/(T[11]-T[7])));

END;
```

```
{**************************************************************}
```

MIN.PAS

```
{FUNCTION Fi0}
{           DFi0}
{PROCEDURE Save}
{             Minimum }

{**** Objective function and its derivates and its minimum value ****}

FUNCTION Fi0(yk:double;k:integer):double;       {Object function = Total cost}

BEGIN
  y[k]:=yk;

  eta[1]:=y[1];
  eta[2]:=y[2];
  eta[4]:=y[3];
  eta[5]:=y[4];

  Cycle;

  Exergies;

  Costs;

  Fi0:=TCost;
END;


FUNCTION DFi0(yk:double;k:integer):double;   {d(Total cost)/d(decision variable)}
LABEL 10;

VAR n,h0,h:double;
    i,l,m:integer;
    A:ARRAY [0..10,0..10] OF double;

BEGIN
  n:=100;
  h0:=dy[k]/n;
  h:=h0;
  A[0,0]:=(Fi0(yk+h,k)-Fi0(yk-h,k))/(2*h);
  FOR m:=1 TO 10 DO
  BEGIN
    h:=h0/pwr(2,m);
    A[m,0]:=(Fi0(yk+h,k)-Fi0(yk-h,k))/(2*h);
    FOR l:=1 TO m DO
    BEGIN
      A[m,l]:=A[m,l-1]+(A[m,l-1]-A[m-1,l-1])/(pwr(2,2*l)-1);
      IF abs(A[m,l-1]-A[m-1,l-1])<(1+abs(A[m,l]))*0.001 THEN GOTO 10;
    END;
  END;
  writeln('DFi0(',k:0,') not found after 10 loops');
10:DFi0:=A[m,l-1];
END;


PROCEDURE Save;           {Save last values for marginal calculations}

VAR i:integer;

BEGIN
  FOR i:=1 TO 5 DO
  BEGIN
    Z0[i]:=Z[i];
    LW0[i]:=LW[i];
  END;
  TC0:=TCost;
  TLW0:=TLW;
  El0:=El;
END;
```

```
PROCEDURE Minimum;          {Find minimum and calculate costs}

VAR i:integer;
    sumdy,sumD:double;
    y0,y1,D0,D1:ARRAY[1..4]OF double;

BEGIN
  FOR k:=1 TO 4 DO D1[k]:=DFi0(y[k],k);          {Start values}
  FOR k:=1 TO 4 DO y1[k]:=y[k];                  {Save start values}

  FOR k:=1 TO 4 DO
  BEGIN
    y[k]:=y1[k]-sgn(D1[k])*dy[k];                {First step}

    {Limits for variable decision variables}
    IF y[k]>ymax[k] THEN y[k]:=(9*y1[k]+ymax[k])/10;
    IF y[k]<ymin[k] THEN y[k]:=(9*y1[k]+ymin[k])/10;
  END;

  FOR k:=1 TO 4 DO D[k]:=DFi0(y[k],k);           {First derivates}

{Check of convergency}
writeln(CHECK,tab10,'TCost:',TCost:12:6,'SEK/yr');
writeln(CHECK,tab10,'Eff. compressor:',y[1]:12:9,'    Marg. cost:',D[1]:18:6,'SEK/yr');
writeln(CHECK,tab10,'Eff. condenser: ',y[2]:12:9,'    Marg. cost:',D[2]:18:6,'SEK/yr');
writeln(CHECK,tab10,'Eff. evaporator:',y[3]:12:9,'    Marg. cost:',D[3]:18:6,'SEK/yr');
writeln(CHECK,tab10,'Eff. el. engine:',y[4]:12:9,'    Marg. cost:',D[4]:18:6,'SEK/yr');

  REPEAT
   FOR k:=1 TO 4 DO                              {Save old values}
   BEGIN
     D0[k]:=D1[k];
     D1[k]:=D[k];
     y0[k]:=y1[k];
     y1[k]:=y[k];
   END;

   FOR k:=1 TO 4 DO
   BEGIN                                         {New values by Newton-Rapson}
     y[k]:=y1[k]-D1[k]*(y1[k]-y0[k])/(D1[k]-D0[k]);
     {Limits for variable decision variables}
     IF y[k]>ymax[k] THEN y[k]:=(y1[k]+ymax[k])/2;
     IF y[k]<ymin[k] THEN y[k]:=(y1[k]+ymin[k])/2;
   END;
   sumD:=0;
   FOR k:=1 TO 4 DO
   BEGIN
     D[k]:=DFi0(y[k],k);                         {New derivaties}
     f[k]:=Fi0(y[k],k);                          {Reset the y[k]'s in Fi0}
     sumD:=sumD+abs(D[k]);                       {Value for "REPEAT-UNTIL" condition}
   END;


{Check of convergency}
writeln(CHECK,tab10,'TCost:',TCost:12:6,'SEK/yr');
writeln(CHECK,tab10,'Eff. compressor:',y[1]:12:9,'    Marg. cost:',D[1]:18:6,'SEK/yr');
writeln(CHECK,tab10,'Eff. condenser: ',y[2]:12:9,'    Marg. cost:',D[2]:18:6,'SEK/yr');
writeln(CHECK,tab10,'Eff. evaporator:',y[3]:12:9,'    Marg. cost:',D[3]:18:6,'SEK/yr');
writeln(CHECK,tab10,'Eff. el. engine:',y[4]:12:9,'    Marg. cost:',D[4]:18:6,'SEK/yr');

  UNTIL sumD/TCost<0.01;        {The sum of the relative marginal costs < 0.01 SEK/yr}
  TCmin:=TCost;                                  {Save the Total Cost}
  TLWoptimum:=TLW;                               {and Lost Work Total at optimum}
  rewrite(DECVAR);
  FOR k:=1 TO 4 DO
  BEGIN
    yoptimum[k]:=y[k];                           {Save decision var.}
    writeln(DECVAR,y[k]);                        {in file DECVAR}
  END;

  FOR i:=1 TO 5 DO
```

```
  BEGIN
   LWoptimum[i]:=LW[i];                    {lost works and}
   Zoptimum[i]:=Z[i];                      {costs at optimum}
  END;
END;
{***************************************************************}
```

HPPRINT.PAS

```
{PROCEDURE      Printcycle}
{               Printfile}
{               Print}
{               Printsens}
{               Printstart}
{               Printoptimum}
{               Printmin}
{************************* Print procedures ***********************}

PROCEDURE Printcycle;

VAR i:integer;

BEGIN
  Cycle;
  writeln(tab10,'T1 = ',T[1]-273.15:6:1,' C');
  writeln(tab10,'T4 = ',T[4]-273.15:6:1,' C');
  writeln(tab10,'dTsh = ',dTsh:6:1,' C');
  writeln(tab10,'dTsc = ',dTsc:6:1,' C');
  writeln;
  writeln(tab10,' no    T      p      density     h       s       e');
  writeln(tab10,'      (C)     (kPa)     (kg/m3)   (kJ/kg)   (kJ/kg*K)  (kJ/kg)');
  writeln;
  FOR i:=1 TO 7 DO writeln(tab10,i:3,T[i]-273.15:7:1,'   ',p[i]/1000:7:2,'   ',1/v[i]:10:2,'    ',
    h[i]/1000:8:2,'    ',s[i]/1000:8:2,'    ',e[i]/1000:8:2);
  writeln;
  writeln(tab10,'Component      Energy      Exergy');
  writeln(tab10,tab10,'       kJ/kg       kJ/kg');
  writeln(tab10,'Compressor : ',(h[3]-h[2])/1000:10:2,(e[3]-e[2])/1000:10:2);
  writeln(tab10,'Evaporator : ',(h[2]-h[7])/1000:10:2,(e[2]-e[7])/1000:10:2);
  writeln(tab10,'Condenser  : ',(h[6]-h[3])/1000:10:2,(e[6]-e[3])/1000:10:2);
  writeln;
  writeln(tab10,'Lost work');
  writeln(tab10,'Compressor : ',(h[3]-h[2]-(e[3]-e[2]))/1000:8:2,' kJ/kg');
  writeln(tab10,'Exp. valve : ',(e[6]-e[7])/1000:8:2,' kJ/kg');
  writeln;
  writeln(tab10,'Efficiency : ',(e[3]-e[2])/(e[2]-e[7]+h[3]-h[2])*100:5:1,' %');
  writeln(tab10,'COP        : ',(h[3]-h[6])/(h[3]-h[2]):5:2);
  writeln(tab10,'COP Carnot : ',1/(T[4]/T[1]-1):5:2);
  writeln;
END;

PROCEDURE Printfile;

VAR i:integer;

BEGIN
  writeln(OUTFILE,tab10,'Flow    Temp.    Pressure   Volume    Energy  Enthalpy  Exergy');
  writeln(OUTFILE,tab10,'#:    Celsius:    bar:      l/s:       kW:      kW:      kW:');
  writeln(OUTFILE);
  FOR i:=1 TO 11 DO writeln(OUTFILE,tab10,'',i:2,T[i]-TK:10:2,p[i]/1E5:10:3,VV[i]*1000:10:3,
    UU[i]/1000:10:3,HH[i]/1000:10:3,EE[i]/1000:10:3);
  writeln(OUTFILE);
  writeln(OUTFILE,tab10,'Flow of refrigerant: ',mr:6:4,' kg/s');
  writeln(OUTFILE,tab10,'Waterflows: ',mwh:6:3,' kg/s in condenser and ',mwc:6:3,' kg/s in evaporator');
  writeln(OUTFILE);
  writeln(OUTFILE,tab10,'Variable decision variables          Marginal price (SEK/yr/unit)');
  writeln(OUTFILE);
  writeln(OUTFILE,tab10,'Efficiency compressor:     ',y[1]:10:4,D[1]:18:6);
  writeln(OUTFILE,tab10,'Efficiency condenser:      ',y[2]:10:4,D[2]:18:6);
  writeln(OUTFILE,tab10,'Efficiency evaporator:     ',y[3]:10:4,D[3]:18:6);
  writeln(OUTFILE,tab10,'Efficiency electric engine:',y[4]:10:4,D[4]:18:6);
  writeln(OUTFILE);
  writeln(OUTFILE,tab10,'                      Condenser, hot side   Evaporator, cold side');
  writeln(OUTFILE,tab10,'Number of H. T. Units:         ',NTUh:10:4,NTUc:20:4);
  writeln(OUTFILE,tab10,'Overall therm. conduct. [W/K]: ',UAh:10:4,UAc:20:4);
  writeln(OUTFILE,tab10,'Log. mean temp. diff. [K]:     ',Tlmh:10:4,Tlmc:20:4);
  writeln(OUTFILE);
  writeln(OUTFILE,tab10,'Zone            Total   Costs     Lost     Ratio     Marginal');
```

```
  writeln(OUTFILE,tab10,'                costs           Works     Costs       Costs');
  writeln(OUTFILE,tab10,'                 SEK     SEK/yr     W       SEK/MJ      SEK/MJ');
  writeln(OUTFILE);
  writeln(OUTFILE,tab10,'Total:               ',TCost:10:1,TLW:10:1,RCT*1E6:12:6,MCT*1E6:12:6);
  writeln(OUTFILE,tab10,'Electricity:         ',pel*hrs*3600*El:10:1,El:10:1,RCEl*1E6:12:6,
    MCEl*1E6:12:6);
  writeln(OUTFILE,tab10,'Compressor: ',TZ[1]:10:1,Z[1]:10:1,LW[1]:10:1,RC[1]*1E6:12:6,
    MC[1]*1E6:12:6);
  writeln(OUTFILE,tab10,'Condenser:  ',TZ[2]:10:1,Z[2]:10:1,LW[2]:10:1,RC[2]*1E6:12:6,
    MC[2]*1E6:12:6);
  writeln(OUTFILE,tab10,'Exp. valve: ',TZ[3]:10:1,Z[3]:10:1,LW[3]:10:1,RC[3]*1E6:12:6,
    MC[3]*1E6:12:6);
  writeln(OUTFILE,tab10,'Evaporator: ',TZ[4]:10:1,Z[4]:10:1,LW[4]:10:1,RC[4]*1E6:12:6,
     MC[4]*1E6:12:6);
  writeln(OUTFILE,tab10,'El. engine: ',TZ[5]:10:1,Z[5]:10:1,LW[5]:10:1,RC[5]*1E6:12:6,
    MC[5]*1E6:12:6);
  writeln(OUTFILE);
  writeln(OUTFILE);
END;


PROCEDURE Print;

VAR i:integer;

BEGIN
  writeln(tab10,'Flow    Temp.    Pressure   Volume    Energy  Enthalpy  Exergy');
  writeln(tab10,'#:    Celsius:    bar:      l/s:       kW:      kW:      kW:');
  writeln;
  FOR i:=1 TO 11 DO writeln(tab10,'',i:2,T[i]-TK:10:2,p[i]/1E5:10:3,VV[i]*1000:10:3,UU[i]/1000:10:3,
    HH[i]/1000:10:3,EE[i]/1000:10:3);
  writeln;
  writeln(tab10,'Flow of refrigerant: ',mr:6:4,' kg/s');
  writeln(tab10,'Waterflows: ',mwh:6:3,' kg/s in condenser and ',mwc:6:3,' kg/s in evaporator');
  writeln;
  writeln(tab10,'Variable decision variables           Marginal price (SEK/yr/unit)');
  writeln;
  writeln(tab10,'Efficiency compressor:     ',y[1]:10:4,D[1]:18:6);
  writeln(tab10,'Efficiency condenser:      ',y[2]:10:4,D[2]:18:6);
  writeln(tab10,'Efficiency evaporator:     ',y[3]:10:4,D[3]:18:6);
  writeln(tab10,'Efficiency electric engine:',y[4]:10:4,D[4]:18:6);
  writeln;
  writeln(tab10,'                       Condenser, hot side   Evaporator, cold side');
  writeln(tab10,'Number of H. T. Units:        ',NTUh:10:4,NTUc:20:4);
  writeln(tab10,'Overall therm. conduct. [W/K]: ',UAh:10:4,UAc:20:4);
  writeln(tab10,'Log. mean temp. diff. [K]:     ',Tlmh:10:4,Tlmc:20:4);
  writeln;
  writeln(tab10,'Zone             Total   Costs     Lost      Ratio      Marginal');
  writeln(tab10,'                costs           Works     Costs       Costs');
  writeln(tab10,'                 SEK     SEK/yr     W       SEK/MJ      SEK/MJ');
  writeln;
  writeln(tab10,'Total:               ',TCost:10:1,TLW:10:1,RCT*1E6:12:6,MCT*1E6:12:6);
  writeln(tab10,'Electricity:         ',pel*hrs*3600*El:10:1,El:10:1,RCEl*1E6:12:6,MCEl*1E6:12:6);
  writeln(tab10,'Compressor: ',TZ[1]:10:1,Z[1]:10:1,LW[1]:10:1,RC[1]*1E6:12:6,MC[1]*1E6:12:6);
  writeln(tab10,'Condenser:  ',TZ[2]:10:1,Z[2]:10:1,LW[2]:10:1,RC[2]*1E6:12:6,MC[2]*1E6:12:6);
  writeln(tab10,'Exp. valve: ',TZ[3]:10:1,Z[3]:10:1,LW[3]:10:1,RC[3]*1E6:12:6,MC[3]*1E6:12:6);
  writeln(tab10,'Evaporator: ',TZ[4]:10:1,Z[4]:10:1,LW[4]:10:1,RC[4]*1E6:12:6,MC[4]*1E6:12:6);
  writeln(tab10,'El. engine: ',TZ[5]:10:1,Z[5]:10:1,LW[5]:10:1,RC[5]*1E6:12:6,MC[5]*1E6:12:6);
  writeln;
  writeln;
END;


PROCEDURE Printsens;

VAR i:integer;

BEGIN {Calculation and printing (SENS) of the sensitivities in the costs and lost works from changes in
                      the decision variables by 1 % at the optimum}

  FOR k:=1 TO 4 DO
  BEGIN
```

```
   yadd[k]:=y[k]+0.01;                              {Add 1 %}
   ysub[k]:=y[k]-0.01;                              {Subtract 1 %}

   IF yadd[k]>ymax[k] THEN yadd[k]:=(9*y[k]+ymax[k])/10;
   IF ysub[k]<ymin[k] THEN ysub[k]:=(9*y[k]+ymin[k])/10;

   TCadd[k]:=Fi0(yadd[k],k)-TCmin;                  {Diff when adding 1 % to yk}
   TLWadd[k]:=TLW-TLWoptimum;                       {Total lost work}
   TCsub[k]:=Fi0(ysub[k],k)-TCmin;                  {Diff when sub. 1 % from yk}
   TLWsub[k]:=TLW-TLWoptimum;                       {Total lost work}
   y[k]:=yoptimum[k];                               {Reset optimum value for y[k]}
   f[k]:=Fi0(y[k],k);                               {Reset y[k] in FUNCTION Fi0}
  END;

  writeln(SENS,tab10,'Changes in the total cost and lost work from changes in');
  writeln(SENS,tab10,'the decision variables');
  writeln;
  writeln(SENS,tab10,'Dec. var.      Change   Change  in total  Change  Change  in total');
  writeln(SENS,tab10,'               (+1%)     cost   lost work  (-1%)   cost   lost work');
  writeln(SENS,tab10,'                         (SEK/yr)    (W)          (SEK/yr)    (W)');
  writeln;
  writeln(SENS,tab10,'Eff. compressor:',yadd[1]-y[1]:6:3,TCadd[1]:8:2,
    TLWadd[1]:10:2,ysub[1]-y[1]:8:3,TCsub[1]:8:2,TLWsub[1]:10:2);
  writeln(SENS,tab10,'Eff. condenser: ',yadd[2]-y[2]:6:3,TCadd[2]:8:2,
    TLWadd[2]:10:2,ysub[2]-y[2]:8:3,TCsub[2]:8:2,TLWsub[2]:10:2);
  writeln(SENS,tab10,'Eff. evaporator:',yadd[3]-y[3]:6:3,TCadd[3]:8:2,
    TLWadd[3]:10:2,ysub[3]-y[3]:8:3,TCsub[3]:8:2,TLWsub[3]:10:2);
  writeln(SENS,tab10,'Eff. el. engine:',yadd[4]-y[4]:6:3,TCadd[4]:8:2,
    TLWadd[4]:10:2,ysub[4]-y[4]:8:3,TCsub[4]:8:2,TLWsub[4]:10:2);
  writeln(SENS);
  writeln(SENS);
END;


PROCEDURE Printstart;

BEGIN
  FOR k:=1 TO 4 DO D[k]:=DFi0(y[k],k);              {Marginal costs}
  FOR k:=1 TO 4 DO y[k]:=ystart[k]*0.999;           {Small change}
  TCost:=Fi0(y[1],1);                               {Get new value}
  Exergies;
  Save;                                             {Save new values}
  FOR k:=1 TO 4 DO y[k]:=ystart[k];                 {Reset start values}
  TCost:=Fi0(y[1],1);                               {Get start values}
  Exergies;
  Margcosts;                                        {Ratio & Marginal costs}
  writeln(OUTFILE);
  writeln(OUTFILE,tab10,'Start values');
  writeln(OUTFILE,tab10,'********************************************');
  writeln(OUTFILE);
  Printfile;
END;


PROCEDURE Printoptimum;

BEGIN
  FOR k:=1 TO 4 DO D[k]:=DFi0(y[k],k);              {Marginal costs}
  FOR k:=1 TO 4 DO y[k]:=yoptimum[k]*0.999;                   {Small change}
  TCost:=Fi0(y[1],1);                               {Get changed values}
  Exergies;
  Save;                                             {Save changed values}
  FOR k:=1 TO 4 DO y[k]:=yoptimum[k];               {Reset optimum values}
  TCost:=Fi0(y[1],1);                               {Get optimum values}
  Exergies;
  Margcosts;
  writeln(OUTFILE);
  writeln(OUTFILE,tab10,'Optimal values');
  writeln(OUTFILE,tab10,'********************************************');
  writeln(OUTFILE);
  Printfile;
  Printsens;
```

```
END;


PROCEDURE Printmin;

BEGIN
  FOR k:=1 TO 4 DO D[k]:=DFi0(y[k],k);          {Marginal costs}
  FOR k:=1 TO 4 DO y[k]:=yoptimum[k]*0.999;                 {Small change}
  TCost:=Fi0(y[1],1);                           {Get changed values}
  Exergies;
  Save;                                         {Save changed values}
  FOR k:=1 TO 4 DO y[k]:=yoptimum[k];           {Reset optimum values}
  TCost:=Fi0(y[1],1);                           {Get optimum values}
  Exergies;
  Margcosts;
  writeln(MIN);
  writeln(MIN,tab10,'Min values');
  writeln(MIN,tab10,'**********');
  writeln(MIN);
  writeln(MIN,tab10,'pel (SEK/kWh): ',pel*3.6E6:6:2);
  writeln(MIN,tab10,'T[9] (degr. Celsius, T[8]=T[9]-10): ',T[8]-273.15:7:1);
  writeln(MIN,tab10,'hrs (hrs/yr): ',hrs:6:0);
  writeln(MIN);
  writeln(MIN,tab10,'Variable decision variables          Marginal price (SEK/yr/unit)');
  writeln(MIN);
  writeln(MIN,tab10,'Efficiency compressor:     ',y[1]:10:4,D[1]:18:6);
  writeln(MIN,tab10,'Efficiency condenser:      ',y[2]:10:4,D[2]:18:6);
  writeln(MIN,tab10,'Efficiency evaporator:     ',y[3]:10:4,D[3]:18:6);
  writeln(MIN,tab10,'Efficiency electric engine:',y[4]:10:4,D[4]:18:6);
  writeln(MIN);
  writeln(MIN,tab10,'                    Condenser, hot side   Evaporator, cold side');
  writeln(MIN,tab10,'Number of H. T. Units:        ',NTUh:10:4,NTUc:20:4);
  writeln(MIN,tab10,'Overall therm. conduct. [W/K]: ',UAh:10:4,UAc:20:4);
  writeln(MIN,tab10,'Log. mean temp. diff. [K]:     ',Tlmh:10:4,Tlmc:20:4);
  writeln(MIN);
  writeln(MIN,tab10,'Zone            Total   Costs     Lost      Ratio      Marginal');
  writeln(MIN,tab10,'                costs             Works     Costs      Costs');
  writeln(MIN,tab10,'                SEK     SEK/yr    W         SEK/MJ     SEK/MJ');
  writeln(MIN);
  writeln(MIN,tab10,'Total:              ',TCost:10:1,TLW:10:1,RCT*1E6:12:6,MCT*1E6:12:6);
  writeln(MIN,tab10,'Electricity:          ',pel*hrs*3600*El:10:1,El:10:1,RCEl*1E6:12:6,MCEl*1E6:12:6);
  writeln(MIN,tab10,'Compressor: ',TZ[1]:10:1,Z[1]:10:1,LW[1]:10:1,RC[1]*1E6:12:6,MC[1]*1E6:12:6);
  writeln(MIN,tab10,'Condenser:  ',TZ[2]:10:1,Z[2]:10:1,LW[2]:10:1,RC[2]*1E6:12:6,MC[2]*1E6:12:6);
  writeln(MIN,tab10,'Exp. valve: ',TZ[3]:10:1,Z[3]:10:1,LW[3]:10:1,RC[3]*1E6:12:6,MC[3]*1E6:12:6);
  writeln(MIN,tab10,'Evaporator: ',TZ[4]:10:1,Z[4]:10:1,LW[4]:10:1,RC[4]*1E6:12:6,MC[4]*1E6:12:6);
  writeln(MIN,tab10,'El. engine: ',TZ[5]:10:1,Z[5]:10:1,LW[5]:10:1,RC[5]*1E6:12:6,MC[5]*1E6:12:6);
  writeln(MIN);
  writeln(MIN);

  {Calculation and printing of the sensitivities in the costs and lost works from changes in the decision
                        variables by 1 % at the optimum}

  FOR k:=1 TO 4 DO
  BEGIN
    yadd[k]:=y[k]+0.01;                          {Add 1 %}
    ysub[k]:=y[k]-0.01;                          {Subtract 1 %}

    IF yadd[k]>ymax[k] THEN yadd[k]:=(9*y[k]+ymax[k])/10;
    IF ysub[k]<ymin[k] THEN ysub[k]:=(9*y[k]+ymin[k])/10;

    TCadd[k]:=Fi0(yadd[k],k)-TCmin;              {Diff when adding 1 % to yk}
    TLWadd[k]:=TLW-TLWoptimum;                   {Total lost work}
    TCsub[k]:=Fi0(ysub[k],k)-TCmin;              {Diff when sub. 1 % from yk}
    TLWsub[k]:=TLW-TLWoptimum;                   {Total lost work}
    y[k]:=yoptimum[k];                           {Reset optimum value for y[k]}
    f[k]:=Fi0(y[k],k);                           {Reset y[k] in FUNCTION Fi0}
  END;

  writeln(MIN,tab10,'Changes in the total cost and lost work from changes in');
  writeln(MIN,tab10,'the decision variables');
  writeln;
  writeln(MIN,tab10,'Dec. var.      Change   Change  in total  Change  Change  in total');
```

```
  writeln(MIN,tab10,'               (+1%)     cost   lost work  (-1%)   cost   lost work');
  writeln(MIN,tab10,'                      (SEK/yr)    (W)          (SEK/yr)    (W)');
  writeln;
  writeln(MIN,tab10,'Eff. compressor:',yadd[1]-y[1]:6:3,TCadd[1]:8:2,
   TLWadd[1]:10:2,ysub[1]-y[1]:8:3,TCsub[1]:8:2,TLWsub[1]:10:2);
  writeln(MIN,tab10,'Eff. condenser: ',yadd[2]-y[2]:6:3,TCadd[2]:8:2,
   TLWadd[2]:10:2,ysub[2]-y[2]:8:3,TCsub[2]:8:2,TLWsub[2]:10:2);
  writeln(MIN,tab10,'Eff. evaporator:',yadd[3]-y[3]:6:3,TCadd[3]:8:2,
   TLWadd[3]:10:2,ysub[3]-y[3]:8:3,TCsub[3]:8:2,TLWsub[3]:10:2);
  writeln(MIN,tab10,'Eff. el. engine:',yadd[4]-y[4]:6:3,TCadd[4]:8:2,
   TLWadd[4]:10:2,ysub[4]-y[4]:8:3,TCsub[4]:8:2,TLWsub[4]:10:2);
  writeln(MIN);
  writeln(MIN);
END;

{**************************************************************}
```

HPCHART.PAS

```
{PROCEDURE Chart}
{************************ Chart procedure *********************}

PROCEDURE Chart;

BEGIN
  write(CHARTFILE,pel*3.6E6:6:2);
{write(CHARTFILE,T[8]-273.15:7:1);}
{write(CHARTFILE,hrs:6:0);}
  write(CHARTFILE,pel*hrs*3600*El:10:2);
  write(CHARTFILE,Z[1]:10:2);
  write(CHARTFILE,Z[2]:10:2);
  write(CHARTFILE,Z[3]:10:2);
  write(CHARTFILE,Z[4]:10:2);
  write(CHARTFILE,Z[5]:10:2);
  writeln(CHARTFILE);
END;
```

Start values

Flow #:	Temperature Celsius:	Pressure bar:	Volume l/s:	Energy kW:	Enthalpy kW:	Exergy kW:
1	2.86	3.384	2.395	9.464	10.275	0.043
2	12.86	3.384	2.516	9.732	10.584	0.052
3	96.80	16.727	0.587	11.103	12.609	1.619
4	64.29	16.727	0.473	10.501	11.292	1.321
5	64.29	16.727	0.041	6.109	6.109	0.333
6	54.29	16.727	0.041	6.109	6.109	0.333
7	2.86	3.384	1.011	6.883	6.109	0.000
8	50.00	1.013	0.158	32.500	32.500	2.655
9	60.00	1.013	0.158	39.000	39.000	3.744
10	10.00	1.013	0.214	8.949	8.949	0.160
11	5.00	1.013	0.214	4.475	4.475	0.040

Flow of refrigerant: 0.0472 kg/s
Waterflows: 0.156 kg/s in condenser and 0.214 kg/s in evaporator

	Variable decision variables	Marginal price (SEK/yr/unit)
Efficiency compressor:	0.7000	-2918.407506
Efficiency condenser:	0.7000	-1009.627099
Efficiency evaporator:	0.7000	-214.819764
Efficiency electric engine:	0.7000	-4870.683036

	Condenser, hot side	Evaporator, cold side
Number of H. T. Units:	1.2040	1.2040
Overall therm. conduct. [W/K]:	782.5823	1077.4614
Log. mean temp. diff. [K]:	8.3058	4.1529

Zone	Total costs SEK	Costs SEK/yr	Lost Works W	Ratio Costs SEK/MJ	Marginal Costs SEK/MJ
Total:		4220.9	1932.5	0.121343	0.059470
Electricity:		3616.8	2893.4	0.069444	0.069444
Compressor:	695.4	113.2	458.4	0.013715	-0.011225
Condenser:	1187.7	156.1	196.5	0.044157	-0.038727
Exp. valve:	236.1	31.0	333.3	0.005175	-0.003383
Evaporator:	1635.2	215.0	76.3	0.156440	-0.362962
El. engine:	675.1	88.8	868.0	0.005681	-0.001635

Optimal values

Flow #:	Temperature Celsius:	Pressure bar:	Volume l/s:	Energy kW:	Enthalpy kW:	Exergy kW:
1	3.14	3.416	2.448	9.767	10.603	0.050
2	13.14	3.416	2.572	10.044	10.923	0.060
3	87.26	15.928	0.613	11.404	12.676	1.548
4	62.00	15.928	0.516	10.799	11.621	1.323
5	62.00	15.928	0.042	6.176	6.176	0.316
6	52.00	15.928	0.042	6.176	6.176	0.316
7	3.14	3.416	0.989	6.963	6.176	0.000
8	50.00	1.013	0.158	32.500	32.500	2.655
9	60.00	1.014	0.158	39.000	39.000	3.744
10	10.00	1.013	0.227	9.494	9.494	0.170
11	5.00	1.013	0.227	4.747	4.747	0.043

Flow of refrigerant: 0.0487 kg/s
Waterflows: 0.156 kg/s in condenser and 0.227 kg/s in evaporator

	Variable decision variables	Marginal price (SEK/yr/unit)
Efficiency compressor:	0.8035	6.854373
Efficiency condenser:	0.8335	1.048093
Efficiency evaporator:	0.7293	2.489199
Efficiency electric engine:	0.9070	4.063741

	Condenser, hot side	Evaporator, cold side
Number of H. T. Units:	1.7927	1.3069
Overall therm. conduct. [W/K]:	1165.2608	1240.7924
Log. mean temp. diff. [K]:	5.5782	3.8259

Zone	Total costs SEK	Costs SEK/yr	Lost Works W	Ratio Costs SEK/MJ	Marginal Costs SEK/MJ
Total:		3387.5	979.4	0.192148	0.000473
Electricity:		2415.6	1932.5	0.069444	0.069444
Compressor:	1340.3	218.1	264.2	0.045869	-0.067795
Condenser:	1739.6	228.7	143.6	0.088453	-0.129606
Exp. valve:	243.5	32.0	315.6	0.005636	-0.003204
Evaporator:	1864.3	245.1	76.3	0.178349	-0.388686
El. engine:	1885.7	247.9	179.6	0.076678	-0.055508