Chalmers tekniska högskola/Göteborgs universitet
Matematik, Optimering
Ann-Brith Strömberg, Michael Patriksson, Fredrik Altenstedt 2000–09–22

TMA135/MAN350 Optimering under osäkerhet
Handledning för fallstudier

De modeller som ligger till grund för fallstudierna är beskrivna i [HV] avsnitt 3.2.3 (sid. 27–41). Varje grupp skall välja en av de fem modellerna "Multi-period production planning" (case1), "Budgeting cost of nursing in a hospital" (case3), "Growing maize and sorghum in Kilosa district" (case4), "Product mix problem" (case5) och "Investment planning for electricity generation" (case6). Eventuellt måste fler än en grupp arbeta med samma modell.

För att kunna analysera modellen "Growing maize and sorghum in Kilosa district" behövs även följande information. "Assume that the buying price of maize is 50 guilders per 100 kilograms; the buying price of sorghum is 40 guilders per 100 kilograms; the selling price of maize is 35 guilders per 100 kilograms; the selling price of sorghum is 30 guilders per 100 kilograms; total acreage $a = 1.5$ ha; the preference factor $f = 3$. Assume that the disturbances ε_1 and ε_2 are independent random variables, and that they are independent of the total rainfall R."

För att kunna analysera modellen "Investment planning for electricity generation" behövs även följande information om fördelningarna för de *oberoende* stokastiska variablerna ξ_j (detta ger totalt 27 olika realiseringar, d.v.s., scenarier):

sannolikhet	ξ_1	ξ_2	ξ_3
0.3	3	2	1
0.4	5	3	2
0.3	7	4	3
$\xi_j^{\max}$	7	4	3

Uppgifter

Uppgift 1–5 samt 7–9 utförs för alla modellerna. Uppgift 6 är olika för de olika modellerna.

1. Generera 500[1] scenarier m.h.a. Matlab-programmet.[2] Finn den steg 1-lösning som erhålls då alla stokastiska parametrar ersätts med sina respektive väntevärden samt förväntade värdet av lösningen i steg 2 då steg 1-lösningen enligt ovan används. Tolka resultatet och diskutera speciellt vilken nytta man kan ha av att istället lösa recourse-modellen.
2. Lös recourse-modellen och jämför resultatet med det som erhölls i deluppgift 1. Diskutera speciellt värdet av den stokastiska lösningen (VSS).
3. Beräkna "wait and see"-lösningen och jämför resultatet med dem erhållna i deluppgift 1 och 2. Beräkna EVPI (expected value of perfect information)?
4. [3] Generera nu istället 20 scenarier och utför deluppgifterna 1–3 igen. Jämför resultaten med motsvarande för 500 scenarier. Vilken nytta har vi av att generera många scenarier?

[1] För elektricitets-genererings-modellen genereras alltid 27 scenarier.
[2] Se avsnitt Programvara.
[3] Ej för elektricitets-genererings-modellen.

Optimeringsprogrammet klarar att lösa modeller med mycket fler är 500 scenarier. Testa gärna med några olika antal scenarier och jämför resultaten!

5. Vilken recourse-struktur har modellen? Beskriv hur subproblemen i L-shaped-algoritmen kan lösas.

6. MULTI-PERIOD PRODUCTION PLANNING

   i. Variera övre gränserna för utökning av produktionskapaciteterna up_{jt} (i datafilen, `ubd_xcapac`) och studera hur resultatet från recourse-modellen beror av dessa (känslighetsanalys).

   ii. Diskutera hur resultatet beror av antagandena om efterfrågan på produkterna (oberoende mellan produkter, oberoende över tiden).

   iii. För denna modell är indelningen i steg 1 och steg 2 inte gjord utifrån tidssteg. Av vilken anledning kan man ha valt att modellera på detta sätt?

   iv. För optimallösningen från recourse-modellen (20 scenarier); beräkna sannolikheten att man inte behöver köpa in någon av de två produkterna under någon av de fyra perioderna.

   BUDGETING COST OF NURSING IN A HOSPITAL

   i. Variera lönekostnader, d.v.s., parametrarna $\tilde{r}_i$, o_i och a_i, $i = 1, 2, 3$ (`rtilde`, `over` respektive `ahire` i datafilen) och studera hur resultatet från recourse-modellen beror av dessa (känslighetsanalys). Vad händer t.ex. då (olika typer av) lönerna ökar?

   ii. För optimallösningen från recourse-modellen (20 scenarier); beräkna sannolikheten att den ordinarie arbetsstyrkan är otillräcklig i någon av tidsperioderna.

   iii. Reflektera över möjligheten att sätta upp patienterna på en väntelista. Detta kan inlemmas i recourse-modellen genom att utöka recourse-matrisen med sex ytterligare kolumner (en för varje tidsperiod) och sex nya recourse-variabler. Formulera och förklara den nya recourse-modellen och diskutera recourse-strukturen.

   GROWING MAIZE AND SORGHUM IN KILOSA DISTRICT

   i. Analysera känsligheten hos recourse-modellen med avseende på parametrarna a och f (`area` respektive `f` i datafilen). Diskutera hur antagandet att ε_1 och ε_2 är oberoende påverkar resultatet. Hur ändras lösningen om inköpspriserna (q_m^+ och q_s^+; i datafilen `qbuy`) fördubblas och försäljningspriserna (q_m^- och q_s^-; i datafilen `qsell`) blir noll?

   ii. För optimallösningen från recourse-modellen (20 scenarier); beräkna sannolikheten att skörden är otillräcklig med avseende på minst ett av näringsämnena. Beräkna väntevärdet för hur mycket mera majs än durra (sorghum) familjen kan äta.

   PRODUCT MIX PROBLEM

   i. Analysera känsligheten hos modellen för variationer hos vinsten per enhet av de olika produkterna (c_j; i datafilen `c`). Samma sak för timkostnaderna för inhyrd arbetskraft i de olika avdelningarna (q_k; `q` i datafilen).

   ii. För optimallösningen från recourse-modellen (20 scenarier); beräkna sannolikheten att man inte behöver hyra arbetskraft till någon av avdelningarna.

   INVESTMENT PLANNING FOR ELECTRICITY GENERATION

   i. Antag att det är möjligt att importera elektricitet. Detta kan modelleras med hjälp av en femte "virtuell" teknik med investeringskostnad $c_5 = 0$ (`cost_invest` i datafilen utökas med ett element) och en relativt hög operativ kostnad q_5 (`cost_prod` i datafilen utökas med ett element). Formulera denna modell och analysera den för några rimliga värden på q_5. Tips: variablerna y_{5j}, som beskriver elimporten, skall inte begränsas av "investeringen" i denna "teknik" (bivillkoren `mode_capacity`). Jämför med de tidigare erhållna resultaten.

ii. Utgå från modellen utan elimport. I praktiken visar det sig att av olika anledningar är all utrustning obrukbar då och då. Ett sätt att beakta detta faktum är att inkorporera tillgänglighetsfaktorer $a_i \in [0.9, 1.0]$ för de olika teknikerna i modellen: då teknik i används är endast andelen $a_i x_i$ av den investarade kapaciteten x_i tillgänglig. Justera modellen och lös den för några olika värden på vektorn av tillgänglighetsfaktorer. Jämför resultatet med de tidigare erhållna resultaten.

7. Skriv en rapport som presenterar modellen och frågeställningar samt redovisar resultaten.

8. Ge synpunkter på uppgifterna, datorprogrammen, användarvänlighet, omständigheter i händelse av att något program kraschat, möjliga funktionsfel samt i övrigt vad du vill!

9. Presentera fallstudien muntligt vid ett seminarium för övriga kursdeltagare.

Programvara

För att lösa uppgifterna behövs tre indatafiler—en med den matematiska modellen, en med deterministiska problemdata och en med scenarier för stokastiska data. Alla data- och programfiler som behövs finns i kursbiblioteket `/users/mdstud/ouo00/` under `Fallstudier/` respektive `bin/`. Börja med att kopiera över indatafilerna (`case#mod.pl`, `case#dat.pl` och `case#.m`, där #=1,3,4,5 eller 6) till ditt eget directory.

Med hjälp av Matlab-programmet `case#.m` genereras det antal scenarier du vill ha, enligt de fördelningar som är angivna i respektive uppgift. Scenarierna lagras på en fil `case#ran`. Ge kommandona

```
[] matlab
>> case#
Ange antal scenarier: 100
```

Vi ger här exempel (jordbruksexemplet från föreläsning 3) på hur och i vilken ordning de övriga körningarna görs. Filen med scenarier för stokastiska data heter för detta exempel `farmran`.

Indatafiler till optimeringsprogrammet genereras med hjälp av programmet `plam2`. Ge kommandona `plam2`, `model farm.`, `data farm.`, `build_core.` och `write_smps(farm,farmran).` enligt:

```
[] plam2
Welcome to SWI-Prolog (Version 3.3.2)
Copyright (c) 1990-2000 University of Amsterdam. All rights reserved.
For help, use ?- help(Topic). or ?- apropos(Word).
1 ?- model farm.
% farmmod.pl compiled 0.01 sec, 8,824 bytes
Yes
2 ?- data farm.
% farmdat.pl compiled 0.01 sec, 3,316 bytes
```

```
Yes
3 ?- build_core.
Yes
4 ?- write_smps(farm,farmran).
wrote core file
wrote time file
Yes
5 ?- ^C
Action (h for help) ? exit
```

Då skapas filerna `farm.alias`, `farm.time`, `farm.core` och `farm.stoch`. Slutligen löser vi det deterministiskt ekvivalenta problemet genom att ge kommandot

```
[] b9 farm -deteq
```

varvid följande utskrift ges på skärmen:

```
doing node 0
doing node 1
doing node 2
doing node 3
********************************************************
* deterministic equivalent results                     *
********************************************************
optimal value -108390.000000
----------------------------------------------------------------------
 nod 1 prob 1.000000e+00 var               pl|be varno 0 : 2.500000000e+02
 nod 1 prob 1.000000e+00 var               pl|co varno 1 : 8.000000000e+01
 nod 1 prob 1.000000e+00 var               pl|wh varno 2 : 1.700000000e+02
----------------------------------------------------------------------
 nod 2 prob 3.333333e-01 var              buy|co varno 3 : 0.000000000e+00
 nod 2 prob 3.333333e-01 var              buy|wh varno 4 : 0.000000000e+00
 nod 2 prob 3.333333e-01 var               ex|be varno 5 : 0.000000000e+00
 nod 2 prob 3.333333e-01 var             sell|be varno 6 : 5.000000000e+03
 nod 2 prob 3.333333e-01 var             sell|co varno 7 : 0.000000000e+00
 nod 2 prob 3.333333e-01 var             sell|wh varno 8 : 2.250000000e+02
----------------------------------------------------------------------
 nod 3 prob 3.333333e-01 var              buy|co varno 3 : 0.000000000e+00
 nod 3 prob 3.333333e-01 var              buy|wh varno 4 : 0.000000000e+00
 nod 3 prob 3.333333e-01 var               ex|be varno 5 : 0.000000000e+00
 nod 3 prob 3.333333e-01 var             sell|be varno 6 : 6.000000000e+03
 nod 3 prob 3.333333e-01 var             sell|co varno 7 : 4.800000000e+01
 nod 3 prob 3.333333e-01 var             sell|wh varno 8 : 3.100000000e+02
----------------------------------------------------------------------
 nod 4 prob 3.333333e-01 var              buy|co varno 3 : 4.800000000e+01
 nod 4 prob 3.333333e-01 var              buy|wh varno 4 : 0.000000000e+00
 nod 4 prob 3.333333e-01 var               ex|be varno 5 : 0.000000000e+00
 nod 4 prob 3.333333e-01 var             sell|be varno 6 : 4.000000000e+03
 nod 4 prob 3.333333e-01 var             sell|co varno 7 : 0.000000000e+00
 nod 4 prob 3.333333e-01 var             sell|wh varno 8 : 1.400000000e+02
----------------------------------------------------------------------
 var               pl|be mean :2.500000000e+02
 var               pl|co mean :8.000000000e+01
```

```
var              pl|wh mean :1.700000000e+02
var             buy|co mean :1.600000000e+01
var             buy|wh mean :0.000000000e+00
var              ex|be mean :0.000000000e+00
var            sell|be mean :5.000000000e+03
var            sell|co mean :1.600000000e+01
var            sell|wh mean :2.250000000e+02
```

För att finna lösningen till problemet där alla stokastiska parametrar antar sina respektive väntevärden byts option `-deteq` mot `-avg`. Förväntade värdet av målfunktionen, då den steg 1-lösning som erhållits från förväntade parametervärden används, fås med option `-avg_full` och wait-and-see-lösningen fås med hjälp av option `-was`.

Datafiler

Filen `farmran` innehåller all information om scenarieträdet för jordbruksexemplet, d.v.s, alla stokastiska parametrar och dess värden i alla scenarier som genererats. En nod i scenarieträdet uttrycks som `node(a,b,c,d,[e])`, där `a` = nodens ordningsnummer i trädet, `b` = nodens nivå i trädet, `c` = sannolikheten att scenariet i noden inträffar, `d` = antal efterföljare till noden i trädet och `e` = de stokastiska parametrarna och dess värden i noden.

```
node(1,0,1,3,[]).

node(2,1,0.33333333,0,[
wh_yield-2.5,
co_yield-3,
be_yield-20
]).

node(3,1,0.33333333,0,[
wh_yield-3,
co_yield-3.6,
be_yield-24
]).

node(4,1,0.33333333,0,[
wh_yield-2,
co_yield-2.4,
be_yield-16
]).
```

Filen `farmmod.pl` innehåller den matematiska modellen för jordbruksexemplet och ser ut som följer:

```
:-plam.
set reqcrops.
set ocrops.
set crops.
param plant_cost:crops :- >= 0.
param sell_price:crops :- >= 0.
```

```
param purch_price:reqcrops :- >= 0.
param min_req:reqcrops :- >= 0.
param avail_land.
param quota:ocrops :- >= 0.
param yield:crops :- >= 0.
param penalty:ocrops :- >= 0.
variable sell:crops :- >=0.
variable buy:reqcrops:- >=0.
variable pl:crops:- >=0.
variable ex:ocrops:- >=0.
objective min:profit:-
    0-sum(crops(P),sell(P)*sell_price(P))+sum(reqcrops(P),buy(P)*purch_price(P))
    +sum(ocrops(P),ex(P)*penalty(P))+sum(crops(P),pl(P)*plant_cost(P)).
subject_to mass:- forall(ocrops(P),yield(P)*pl(P) = sell(P)).
subject_to max_land:- sum(crops(P),pl(P)) =< avail_land.
subject_to min_req:- forall(reqcrops(P),yield(P)*pl(P)+buy(P)-sell(P) >= min_req(P)).
subject_to quota:- forall(ocrops(P),yield(P)*pl(P) =< quota(P)+ex(P)).
period(pl,0).
period(sell,1).
period(buy,1).
period(ex,1).
first_period(0).
last_period(1).
```

I filen `farmdat.pl` finns alla deterministiska data till modellen samt deklaration av de stokastiska parametrarna (dess värden finns i filen `farmran`):

```
%%  Data för jordbruksexemplet
%set reqcrops
reqcrops(wh).
reqcrops(co).
%set ocrops.
ocrops(be).
%set crops.
crops(A):-ocrops(A).
crops(A):-reqcrops(A).

s_period(be,0).
s_period(wh,1).
s_period(co,2).
plant_cost(wh,150).
plant_cost(co,230).
plant_cost(be,260).
yield(wh,wh_yield).
yield(co,co_yield).
yield(be,be_yield).
sell_price(wh,170).
sell_price(co,150).
sell_price(be,36).
purch_price(wh,238).
purch_price(co,210).
min_req(wh,200).
min_req(co,240).
quota(be,6000).
avail_land(500).
```

```
penalty(be,26).
mark_random(wh_yield,2.5).
mark_random(co_yield,3).
mark_random(be_yield,20).
```

I filen `farm.alias` kopplas variabel/bivillkorsnamn i .pl-filerna ihop med variabel/bivillkorsnamn i smps-filerna (indata till lösaren):

```
quota|be-C1
min_req|wh-C2
min_req|co-C3
max_land-C4
mass|be-C5
pl|be-X1
pl|co-X2
pl|wh-X3
buy|co-X4
buy|wh-X5
ex|be-X6
sell|be-X7
sell|co-X8
sell|wh-X9
```

Filen `farm.core` innehåller kärnproblemet, d.v.s, koefficienterna i modellen då de stokastiska parametrarna ersätts med sina respektive väntevärden:

```
NAME          NONAME
ROWS
 N  OBJ
 L  C4
 L  C1
 G  C2
 G  C3
 E  C5
COLUMNS
    X1           C4            1.00000000e+00
    X1           OBJ           2.60000000e+02
    X1           C1            2.00000000e+01
    X1           C5            2.00000000e+01
    X2           C4            1.00000000e+00
    X2           OBJ           2.30000000e+02
    X2           C3            3.00000000e+00
    X3           C4            1.00000000e+00
    X3           OBJ           1.50000000e+02
    X3           C2            2.50000000e+00
    X4           OBJ           2.10000000e+02
    X4           C3            1.00000000e+00
    X5           OBJ           2.38000000e+02
    X5           C2            1.00000000e+00
    X6           OBJ           2.60000000e+01
    X6           C1           -1.00000000e+00
    X7           OBJ          -3.60000000e+01
    X7           C5           -1.00000000e+00
```

```
    X8          OBJ           -1.50000000e+02
    X8          C3            -1.00000000e+00
    X9          OBJ           -1.70000000e+02
    X9          C2            -1.00000000e+00
RHS
    RHS         C4            5.00000000e+02
    RHS         C1            6.00000000e+03
    RHS         C2            2.00000000e+02
    RHS         C3            2.40000000e+02
    RHS         C5            0.00000000e+00
BOUNDS
ENDATA
```

I filen `farm.stoch` finns värdena på de stokastiska variablerna för alla scenarier, tillsammans med deras sannolikheter (alltså de parametrar i .core-filen som är stokastiska):

```
STOCH         NONAME
SCENARIOS     DISCRETE
 SC SCEN0       ROOT          3.33333333e-01  PER0
    X1          C1            2.00000000e+01
    X1          C5            2.00000000e+01
    X2          C3            3.00000000e+00
    X3          C2            2.50000000e+00
 SC SCEN1       SCEN0         3.33333333e-01  PER1
    X1          C1            2.40000000e+01
    X1          C5            2.40000000e+01
    X2          C3            3.60000000e+00
    X3          C2            3.00000000e+00
 SC SCEN2       SCEN0         3.33333333e-01  PER1
    X1          C1            1.60000000e+01
    X1          C5            1.60000000e+01
    X2          C3            2.39999999e+00
    X3          C2            2.00000000e+00
ENDATA
```

I filen `farm.time` finns information om vilket tidssteg de olika variablerna tillhör (X1-X3 är steg 1-variabler, X4-X9 är steg 2-variabler):

```
TIME          NONAME
PERIODS       LP
    X1          C4                          PER0
    X4          C1                          PER1
ENDATA
```