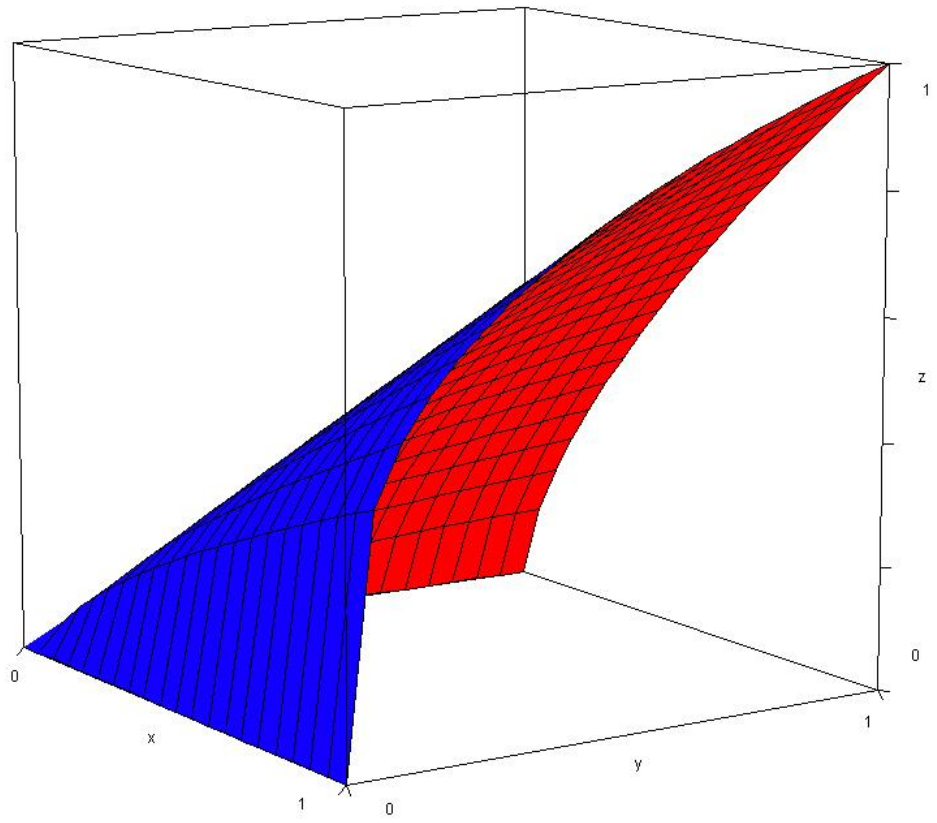
MICROECONOMICS 2018

Mid Sweden University, Sundsvall (Lecture 4)

Peter Lohmander
www.Lohmander.com & Peter@Lohmander.com



Industriell ekonomi:

Fokus på ekonomisk optimal kombination av produktion och lager i tillverkande företag

med hänsyn till produktionskostnadsfunktionens egenskaper, räntekostnader, övriga lagerkostnader samt stokastisk efterfrågan

Särskild motivering

- Flera av gruppernas inlämningsuppgifter har frågor och problemformuleringar som fokuserar på dessa problem.
- De metoder som beskrivs i denna presentation är direkt tillämpbara för att lösa sådana problem.

Några referenser till metodik och tillämpningar

- **Stochastic dynamic programming (From Wikipedia, the free encyclopedia)**
- **https://en.wikipedia.org/wiki/Stochastic_dynamic_programming**

- **Winston, W.L.: Operations Research, Applications and Algorithms. Thomson Brooks/Cole, Belmont (2004)**
- **https://www.amazon.com/Operations-Research-Applications-Algorithms-InfoTrac/dp/0534380581**

- **Lohmander, P., Two Approaches to Optimal Adaptive Control under Large Dimensionality, INTERNATIONAL ROBOTICS AND AUTOMATION JOURNAL, Volume 3, Issue 4, 2017, DOI:10.15406/iratj.2017.03.00062**
  http://medcraveonline.com/IRATJ/IRATJ-03-00062.php
  http://www.Lohmander.com/PL_171204a.pdf
  http://www.Lohmander.com/PL_171204aORIG.pdf
  http://www.Lohmander.com/PL_171204aORIG.docx

- **Lohmander, P., Applications and Mathematical Modeling in Operations Research, In: Cao BY. (ed) Fuzzy Information and Engineering and Decision. IWDS 2016. Advances in Intelligent Systems and Computing, vol 646. Springer, Cham, 2018 Print ISBN 978-3-319-66513-9, Online ISBN 978-3-319-66514-6, eBook Package: Engineering, LAMMOR**
  https://doi.org/10.1007/978-3-319-66514-6_5
- **http://www.lohmander.com/lammor.pdf**

Terminalvillkor. Efter planeringshorisonten har vi inga kostnader eller intäkter.

$$f(T+1,i)=0 \quad \forall i$$

I varje period, t, maximerar vi f(t,i), förväntat nuvärde av vad som händer i aktuell period samt i framtida perioder. Ingående lagernivå är i. Vi väljer u, produktionsnivå. Denna påverkar C, vilket är produktionskostnaden. Lagringskostnad antages här vara proportionell mot lagringsvolymens maximala värde under perioden. Lagringskostnad per enhet och period är h. Esales är förväntad försäljning (förbrukning) av lagret under perioden och beror på ingående lager samt produktionsnivå under perioden (samt efterfrågan). (Efterfrågan är stokastisk och beskrivs på senare bilder.) Diskonteringsfaktorn för en period, är d. Tau betecknar övergångssannolikhet (från en lagernivå till en annan) och beror av efterfrågans sannolikhetsfördelning. Mer detaljer kommer på följande sidor.

$$f(t,i)=\max_{u_t}\left[-C(u_t)-h*(i+u_t)+P*ESales(i+u_t)+d\sum_j \tau(j\,|\,(i+u_t))\,f(t+1,j)\right]$$

$$\forall i, t \le T$$

Modellen (nedan) har programmerats i ett generellt programmeringsprogram som kan laddas ner gratis på internet.

https://en.wikipedia.org/wiki/QB64

http://www.qb64.net/

LA4_ex1

CASE: ORIGINAL CASE
****

```
REM L4A.bas
REM Peter Lohmander 180202
SCREEN 11
CLS

OPEN "__AAA_L4_OUT.txt" FOR OUTPUT AS #1

DIM f(20, 20), u(20, 20), T(20, 20), ProbD(20), ESales(20)
```

```
PRINT ""
PRINT "Stochastic dynamic programming L4"
PRINT "Peter Lohmander 180202"
PRINT ""

PRINT #1, ""
PRINT #1, "Stochastic dynamic programming L4"
PRINT #1, "Peter Lohmander 180202"
PRINT #1, ""
```

```
REM Some parameters
r = 0.05
deltat = 1 / 12
Disc = (1 / (1 + r)) ^ deltat
PRINT " r = "; r; "  deltat = "; deltat; "  Disc = "; Disc
PRINT #1, " r = "; r; "  deltat = "; deltat; "  Disc = "; Disc
```

```
REM Definition of the demand probability distribution matrix
SumProbD = 0
FOR i = 0 TO 20
    ProbD(i) = 5 - i
    IF ProbD(i) < 0 THEN ProbD(i) = 0
    SumProbD = SumProbD + ProbD(i)
NEXT i
FOR i = 0 TO 20
    ProbD(i) = ProbD(i) / SumProbD
NEXT i
```

```
PRINT ""
PRINT "ProbD"
PRINT "    i  Prob"
PRINT ""
FOR i = 0 TO 10
   PRINT USING "##.###"; i, ProbD(i)
NEXT i

PRINT #1, ""
PRINT #1, "ProbD"
PRINT #1, "    i  Prob"
PRINT #1, ""
FOR i = 0 TO 10
   PRINT #1, USING "##.###"; i, ProbD(i)
NEXT i
INPUT zzz
```

```
REM Definition of the expected sales matrix
FOR i = 0 TO 20
    ESales(i) = 0
    FOR j = 0 TO 20
        IF j < i + 0.5 THEN ESales(i) = ESales(i) + ProbD(j) * j
        IF j > i THEN ESales(i) = ESales(i) + ProbD(j) * i
    NEXT j
NEXT i
```

```
PRINT ""
PRINT "ESales"
PRINT "    i  E(S)"
PRINT ""
FOR i = 0 TO 10
   PRINT USING "##.###"; i, ESales(i)
NEXT i

PRINT #1, ""
PRINT #1, "ESales"
PRINT #1, "    i  E(S)"
PRINT #1, ""
FOR i = 0 TO 10
   PRINT #1, USING "##.###"; i, ESales(i)
NEXT i
INPUT zzz
```

```
REM Definition of the transition probability matrix
FOR i = 0 TO 20
   FOR j = 0 TO 20
      T(i, j) = 0
   NEXT j

   FOR j = i TO i - 20 STEP -1
      demand = i - j
      Prob = ProbD(demand)
      IF j > -0.5 THEN T(i, j) = Prob
      IF j < -0.5 THEN T(i, 0) = T(i, 0) + Prob
   NEXT j
NEXT i
```

```
PRINT ""
PRINT "Transition probability matrix"
PRINT "j =          0    1    2    3    4    5    6    7
8    9    10 "
PRINT ""
FOR i = 0 TO 10
   PRINT "i= ";
   PRINT USING "###"; i;
   PRINT " *  ";
   FOR j = 0 TO 10
      PRINT USING "##.###"; T(i, j);
   NEXT j
   PRINT ""
NEXT i
```

```
PRINT #1, ""
PRINT #1, "Transition probability matrix"
PRINT #1, "j =          0    1    2    3    4    5    6
7    8    9    10 "
PRINT #1, ""
FOR i = 0 TO 10
   PRINT #1, "i= ";
   PRINT #1, USING "###"; i;
   PRINT #1, " *  ";
   FOR j = 0 TO 10
      PRINT #1, USING "##.###"; T(i, j);
   NEXT j
   PRINT #1, ""
NEXT i
INPUT zzz
```

```
REM The stochastic dynamic programming algorithm

FOR T = 0 TO 20
   FOR i = 0 TO 20
      f(T, i) = 0
      u(T, i) = 0
   NEXT i
NEXT T

FOR i = 0 TO 20
   f(20, i) = 0
   u(20, i) = 0
NEXT i
```

```
FOR T = 19 TO 0 STEP -1

   FOR i = 0 TO 20
      fopt = -1000
      uopt = 0
      FOR u = 0 TO 20 - i
         Cprod = 10 * u + 1 * u * u
         Cmaxlager = 0.1 * (i + u)
         ERev = 25 * ESales(i + u)
         Efuture = 0
         FOR m = 0 TO 20
            Efuture = Efuture + T(i + u, m) * f(T + 1, m)
         NEXT m
         fev = -Cprod - Cmaxlager + ERev + Disc * Efuture
         IF fev > fopt THEN uopt = u
         IF fev > fopt THEN fopt = fev
      NEXT u
      f(T, i) = fopt
      u(T, i) = uopt
   NEXT i

NEXT T
```

```
PRINT ""
PRINT "f(t,i)"
PRINT "i =         0   1   2   3   4   5   6   7   8   9   10 "
PRINT ""
FOR T = 0 TO 20
   PRINT "t = ";
   PRINT USING "###"; T;
   PRINT " *  ";
   FOR i = 0 TO 10
      PRINT USING "#####"; f(T, i);
   NEXT i
   PRINT ""
NEXT T
INPUT zzz
```

```
PRINT ""
PRINT "u(t,i)"
PRINT "i =        0   1   2   3   4   5   6   7   8   9   10 "
PRINT ""
FOR T = 0 TO 20
   PRINT "t= ";
   PRINT USING "###"; T;
   PRINT " *  ";
   FOR i = 0 TO 10
      PRINT USING "#####"; u(T, i);
   NEXT i
   PRINT ""
NEXT T
```

```
PRINT #1, ""
PRINT #1, "f(t,i)"
PRINT #1, "i =        0   1   2   3   4   5   6   7   8   9   10 "
PRINT #1, ""
FOR T = 0 TO 20
   PRINT #1, "t = ";
   PRINT #1, USING "###"; T;
   PRINT #1, " *  ";
   FOR i = 0 TO 10
      PRINT #1, USING "#####"; f(T, i);
   NEXT i
   PRINT #1, ""
NEXT T
INPUT zzz
```

```
PRINT #1, ""
PRINT #1, "u(t,i)"
PRINT #1, "i =        0   1   2   3   4   5   6   7   8   9   10 "
PRINT #1, ""
FOR T = 0 TO 20
   PRINT #1, "t= ";
   PRINT #1, USING "###"; T;
   PRINT #1, " *  ";
   FOR i = 0 TO 10
      PRINT #1, USING "#####"; u(T, i);
   NEXT i
   PRINT #1, ""
NEXT T
INPUT zzz
```

Stochastic dynamic programming L4
Peter Lohmander 180202

r = .05 deltat = 8.333334E-02 Disc = .9959424

ProbD

i	Prob
0.000	0.333
1.000	0.267
2.000	0.200
3.000	0.133
4.000	0.067
5.000	0.000
6.000	0.000
7.000	0.000
8.000	0.000
9.000	0.000
10.000	0.000

RESULTS

ESales

i	E(S)
0.000	0.000
1.000	0.667
2.000	1.067
3.000	1.267
4.000	1.333
5.000	1.333
6.000	1.333
7.000	1.333
8.000	1.333
9.000	1.333
10.000	1.333

Transition probability matrix

j =	0	1	2	3	4	5	6	7	8	9	10
i= 0 *	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
i= 1 *	0.667	0.333	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
i= 2 *	0.400	0.267	0.333	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
i= 3 *	0.200	0.200	0.267	0.333	0.000	0.000	0.000	0.000	0.000	0.000	0.000
i= 4 *	0.067	0.133	0.200	0.267	0.333	0.000	0.000	0.000	0.000	0.000	0.000
i= 5 *	0.000	0.067	0.133	0.200	0.267	0.333	0.000	0.000	0.000	0.000	0.000
i= 6 *	0.000	0.000	0.067	0.133	0.200	0.267	0.333	0.000	0.000	0.000	0.000
i= 7 *	0.000	0.000	0.000	0.067	0.133	0.200	0.267	0.333	0.000	0.000	0.000
i= 8 *	0.000	0.000	0.000	0.000	0.067	0.133	0.200	0.267	0.333	0.000	0.000
i= 9 *	0.000	0.000	0.000	0.000	0.000	0.067	0.133	0.200	0.267	0.333	0.000
i= 10 *	0.000	0.000	0.000	0.000	0.000	0.000	0.067	0.133	0.200	0.267	0.333

f(t,i)

i =	0	1	2	3	4	5	6	7	8	9	10
t = 0 *	306	321	335	348	361	373	385	396	408	419	430
t = 1 *	290	305	319	332	345	357	369	381	392	403	414
t = 2 *	274	289	303	316	329	341	353	365	376	387	398
t = 3 *	258	273	287	300	313	325	337	348	360	371	382
t = 4 *	242	257	271	284	297	309	321	332	343	354	365
t = 5 *	226	241	255	268	281	293	305	316	327	338	349
t = 6 *	210	225	239	252	264	277	288	300	311	322	333
t = 7 *	193	208	222	235	248	260	272	283	294	305	316
t = 8 *	177	192	206	219	232	244	255	267	278	289	299
t = 9 *	161	176	190	203	215	227	239	250	261	272	283
t = 10 *	145	160	173	186	199	211	222	233	244	255	265
t = 11 *	128	143	157	170	182	194	205	216	227	238	247
t = 12 *	112	127	140	153	166	177	189	200	210	220	228
t = 13 *	95	110	124	137	149	161	172	182	192	200	207
t = 14 *	79	94	107	120	132	143	154	164	172	179	184
t = 15 *	63	78	91	104	115	126	136	144	151	155	158
t = 16 *	47	62	75	87	98	108	116	122	126	128	128
t = 17 *	32	46	59	70	81	88	93	96	97	97	97
t = 18 *	18	31	42	53	60	63	65	65	65	65	65
t = 19 *	6	17	26	31	33	33	33	33	33	32	32
t = 20 *	0	0	0	0	0	0	0	0	0	0	0

u(t,i)

i =	0	1	2	3	4	5	6	7	8	9	10
t= 0 *	3	2	2	1	1	1	1	1	1	1	0
t= 1 *	3	2	2	1	1	1	1	1	1	1	0
t= 2 *	3	2	2	1	1	1	1	1	1	0	0
t= 3 *	3	2	2	1	1	1	1	1	1	0	0
t= 4 *	3	2	2	1	1	1	1	1	1	0	0
t= 5 *	3	2	2	1	1	1	1	1	1	0	0
t= 6 *	3	2	2	1	1	1	1	1	1	0	0
t= 7 *	3	2	2	1	1	1	1	1	1	0	0
t= 8 *	3	2	2	1	1	1	1	1	0	0	0
t= 9 *	3	2	2	1	1	1	1	1	0	0	0
t= 10 *	3	2	2	1	1	1	1	1	0	0	0
t= 11 *	3	2	2	1	1	1	1	0	0	0	0
t= 12 *	3	2	2	1	1	1	1	0	0	0	0
t= 13 *	3	2	2	1	1	1	0	0	0	0	0
t= 14 *	3	2	2	1	1	0	0	0	0	0	0
t= 15 *	3	2	2	1	1	0	0	0	0	0	0
t= 16 *	2	2	1	1	0	0	0	0	0	0	0
t= 17 *	2	2	1	0	0	0	0	0	0	0	0
t= 18 *	2	1	1	0	0	0	0	0	0	0	0
t= 19 *	1	0	0	0	0	0	0	0	0	0	0
t= 20 *	0	0	0	0	0	0	0	0	0	0	0

LA4_ex2

CASE: HIGH RATE OF INTEREST

****

```
REM Some parameters
r = 0.10
deltat = 1 / 12
Disc = (1 / (1 + r)) ^ deltat
PRINT " r = "; r; "  deltat = "; deltat; "  Disc = "; Disc
PRINT #1, " r = "; r; "  deltat = "; deltat; "   Disc = "; Disc
```

Stochastic dynamic programming L4
Peter Lohmander 180202

r = .1 deltat = 8.333334E-02 Disc = .9920889

f(t,i)

i =	0	1	2	3	4	5	6	7	8	9	10
t = 0 *	293	308	322	335	348	360	371	383	394	405	415
t = 1 *	278	293	307	320	333	345	357	368	379	390	401
t = 2 *	264	279	292	305	318	330	342	353	364	375	386
t = 3 *	249	264	278	291	303	315	327	338	349	360	371
t = 4 *	234	249	263	276	288	300	312	323	334	345	356
t = 5 *	219	234	247	260	273	285	297	308	319	330	340
t = 6 *	203	218	232	245	258	270	281	292	303	314	325
t = 7 *	188	203	217	230	242	254	266	277	288	299	310
t = 8 *	173	188	201	214	227	239	250	261	272	283	294
t = 9 *	157	172	186	199	211	223	234	245	256	267	278
t = 10 *	141	156	170	183	195	207	219	230	240	251	261
t = 11 *	126	141	154	167	180	191	202	213	224	234	244
t = 12 *	110	125	138	151	164	175	186	197	207	217	225
t = 13 *	94	109	122	135	147	159	170	180	190	198	205
t = 14 *	78	93	106	119	131	142	153	163	171	178	182
t = 15 *	62	77	90	103	114	125	135	143	150	154	157
t = 16 *	47	61	74	87	98	108	116	121	125	127	128
t = 17 *	32	46	59	70	80	88	93	95	96	97	97
t = 18 *	18	31	42	53	60	63	65	65	65	65	65
t = 19 *	6	17	26	31	33	33	33	33	33	32	32
t = 20 *	0	0	0	0	0	0	0	0	0	0	0

u(t,i)

i =	0	1	2	3	4	5	6	7	8	9	10
t= 0 *	3	2	2	1	1	1	1	1	1	0	0
t= 1 *	3	2	2	1	1	1	1	1	1	0	0
t= 2 *	3	2	2	1	1	1	1	1	0	0	0
t= 3 *	3	2	2	1	1	1	1	1	0	0	0
t= 4 *	3	2	2	1	1	1	1	1	0	0	0
t= 5 *	3	2	2	1	1	1	1	1	0	0	0
t= 6 *	3	2	2	1	1	1	1	1	0	0	0
t= 7 *	3	2	2	1	1	1	1	1	0	0	0
t= 8 *	3	2	2	1	1	1	1	1	0	0	0
t= 9 *	3	2	2	1	1	1	1	1	0	0	0
t= 10 *	3	2	2	1	1	1	1	0	0	0	0
t= 11 *	3	2	2	1	1	1	1	0	0	0	0
t= 12 *	3	2	2	1	1	1	1	0	0	0	0
t= 13 *	3	2	2	1	1	1	0	0	0	0	0
t= 14 *	3	2	2	1	1	0	0	0	0	0	0
t= 15 *	3	2	2	1	1	0	0	0	0	0	0
t= 16 *	2	2	1	1	0	0	0	0	0	0	0
t= 17 *	2	2	1	0	0	0	0	0	0	0	0
t= 18 *	2	1	1	0	0	0	0	0	0	0	0
t= 19 *	1	0	0	0	0	0	0	0	0	0	0
t= 20 *	0	0	0	0	0	0	0	0	0	0	0

LA4_ex3

CASE: VERY SHORT PERIODS

****

```
REM Some parameters
r = 0.05
deltat = 1 / 365
Disc = (1 / (1 + r)) ^ deltat
PRINT " r = "; r; "  deltat = "; deltat; "  Disc = "; Disc
PRINT #1, " r = "; r; "  deltat = "; deltat; "   Disc = "; Disc
```

Stochastic dynamic programming L4
Peter Lohmander 180202

r = .05 deltat = 2.739726E-03 Disc = .9998664

f(t,i)

i =	0	1	2	3	4	5	6	7	8	9	10
t = 0 *	319	334	349	362	375	387	400	411	423	434	445
t = 1 *	302	317	331	344	357	370	382	394	406	417	428
t = 2 *	285	300	314	327	340	353	365	377	388	399	411
t = 3 *	268	283	297	310	323	336	348	359	371	382	393
t = 4 *	251	266	280	293	306	318	330	342	353	365	376
t = 5 *	233	248	263	276	289	301	313	325	336	347	358
t = 6 *	216	231	245	258	271	284	296	307	319	330	341
t = 7 *	199	214	228	241	254	267	278	290	301	312	323
t = 8 *	182	197	211	224	237	249	261	272	284	295	305
t = 9 *	165	180	194	207	220	232	244	255	266	277	288
t = 10 *	148	163	177	190	203	215	226	237	248	259	270
t = 11 *	131	146	160	173	185	197	209	220	231	241	251
t = 12 *	114	129	143	156	168	180	191	202	213	222	231
t = 13 *	97	112	126	139	151	162	173	184	194	203	210
t = 14 *	80	95	109	122	134	145	156	166	174	181	186
t = 15 *	64	79	92	105	116	127	137	145	152	156	159
t = 16 *	47	62	75	88	99	109	117	123	126	128	129
t = 17 *	32	46	59	70	81	88	93	96	97	97	97
t = 18 *	18	31	42	53	60	63	65	65	65	65	65
t = 19 *	6	17	26	31	33	33	33	33	33	32	32
t = 20 *	0	0	0	0	0	0	0	0	0	0	0

u(t,i)

i =	0	1	2	3	4	5	6	7	8	9	10
t= 0 *	3	2	2	2	1	1	1	1	1	1	1
t= 1 *	3	2	2	2	1	1	1	1	1	1	1
t= 2 *	3	2	2	2	1	1	1	1	1	1	1
t= 3 *	3	2	2	2	1	1	1	1	1	1	1
t= 4 *	3	2	2	1	1	1	1	1	1	1	0
t= 5 *	3	2	2	1	1	1	1	1	1	1	0
t= 6 *	3	2	2	1	1	1	1	1	1	1	0
t= 7 *	3	2	2	1	1	1	1	1	1	0	0
t= 8 *	3	2	2	1	1	1	1	1	1	0	0
t= 9 *	3	2	2	1	1	1	1	1	1	0	0
t= 10 *	3	2	2	1	1	1	1	1	0	0	0
t= 11 *	3	2	2	1	1	1	1	1	0	0	0
t= 12 *	3	2	2	1	1	1	1	0	0	0	0
t= 13 *	3	2	2	1	1	1	0	0	0	0	0
t= 14 *	3	2	2	1	1	1	0	0	0	0	0
t= 15 *	3	2	2	1	1	0	0	0	0	0	0
t= 16 *	2	2	1	1	0	0	0	0	0	0	0
t= 17 *	2	2	1	0	0	0	0	0	0	0	0
t= 18 *	2	1	1	0	0	0	0	0	0	0	0
t= 19 *	1	0	0	0	0	0	0	0	0	0	0
t= 20 *	0	0	0	0	0	0	0	0	0	0	0

LA4_ex4

CASE: HIGH PRODUCTION SETUP COST

****

Cprod = 10 * u + 1 * u * u

IF u > 0 THEN Cprod = Cprod + 10

f(t,i)

i =	0	1	2	3	4	5	6	7	8	9	10
t = 0 *	191	207	224	240	256	273	289	305	321	336	352
t = 1 *	181	197	214	230	246	263	279	295	311	326	342
t = 2 *	170	187	203	220	236	252	269	285	300	316	332
t = 3 *	160	177	193	209	226	242	258	274	290	306	322
t = 4 *	150	167	183	199	216	232	248	264	280	296	311
t = 5 *	140	157	173	189	206	222	238	254	270	286	301
t = 6 *	130	146	163	179	195	212	228	244	260	275	291
t = 7 *	119	136	152	168	185	201	217	233	249	265	280
t = 8 *	109	126	142	158	175	191	207	223	239	254	270
t = 9 *	98	115	132	148	164	181	197	213	228	244	259
t = 10 *	88	105	121	137	154	170	186	202	218	233	247
t = 11 *	78	94	111	127	143	160	176	191	207	221	235
t = 12 *	67	84	100	116	133	149	165	180	195	208	220
t = 13 *	56	73	89	105	122	138	154	169	182	194	203
t = 14 *	46	63	79	95	112	128	143	156	167	176	182
t = 15 *	36	52	67	85	101	116	130	141	149	154	157
t = 16 *	26	42	57	75	91	104	114	121	125	127	128
t = 17 *	16	31	48	65	78	87	93	95	97	97	97
t = 18 *	6	22	40	52	60	63	65	65	65	65	65
t = 19 *	0	17	26	31	33	33	33	33	33	32	32
t = 20 *	0	0	0	0	0	0	0	0	0	0	0

u(t,i)

i =	0	1	2	3	4	5	6	7	8	9	10
t= 0 *	3	3	3	3	0	0	0	0	0	0	0
t= 1 *	3	3	3	3	0	0	0	0	0	0	0
t= 2 *	3	3	3	3	0	0	0	0	0	0	0
t= 3 *	3	3	3	3	0	0	0	0	0	0	0
t= 4 *	3	3	3	3	0	0	0	0	0	0	0
t= 5 *	3	3	3	3	0	0	0	0	0	0	0
t= 6 *	3	3	3	3	0	0	0	0	0	0	0
t= 7 *	3	3	3	3	0	0	0	0	0	0	0
t= 8 *	3	3	3	3	0	0	0	0	0	0	0
t= 9 *	3	3	3	3	0	0	0	0	0	0	0
t= 10 *	3	3	3	3	0	0	0	0	0	0	0
t= 11 *	3	3	3	3	0	0	0	0	0	0	0
t= 12 *	3	3	3	0	0	0	0	0	0	0	0
t= 13 *	3	3	3	0	0	0	0	0	0	0	0
t= 14 *	3	3	3	0	0	0	0	0	0	0	0
t= 15 *	3	3	3	0	0	0	0	0	0	0	0
t= 16 *	3	3	0	0	0	0	0	0	0	0	0
t= 17 *	3	2	0	0	0	0	0	0	0	0	0
t= 18 *	2	0	0	0	0	0	0	0	0	0	0
t= 19 *	0	0	0	0	0	0	0	0	0	0	0
t= 20 *	0	0	0	0	0	0	0	0	0	0	0

LA4_ex5

CASE: VERY CONVEX PRODUCTION COST FUNCTION

****

$$Cprod = 10 * u + 3 * u * u$$

f(t,i)

i =	0	1	2	3	4	5	6	7	8	9	10
t = 0 *	223	244	263	282	300	317	333	349	364	378	392
t = 1 *	211	232	251	270	288	305	321	336	351	365	379
t = 2 *	199	220	239	258	275	292	309	324	338	352	366
t = 3 *	187	207	226	245	263	280	296	311	326	339	353
t = 4 *	175	195	214	233	251	268	283	298	313	326	339
t = 5 *	163	183	202	221	238	255	271	285	299	313	326
t = 6 *	151	171	190	209	226	242	258	272	286	300	313
t = 7 *	139	159	178	196	213	230	245	259	273	286	299
t = 8 *	126	146	165	184	201	217	232	246	259	272	285
t = 9 *	114	134	153	171	188	204	218	232	246	259	271
t = 10 *	102	122	141	159	175	191	205	219	232	245	257
t = 11 *	90	110	129	146	162	177	191	205	218	230	241
t = 12 *	78	97	116	134	149	164	177	190	203	215	225
t = 13 *	66	85	104	121	136	150	163	176	188	197	206
t = 14 *	54	73	92	108	123	136	149	161	170	178	183
t = 15 *	43	62	80	95	109	122	133	143	150	155	158
t = 16 *	31	50	67	82	95	106	115	121	125	128	128
t = 17 *	20	39	55	68	80	88	93	96	97	97	97
t = 18 *	11	28	41	53	60	63	65	65	65	65	65
t = 19 *	4	17	26	31	33	33	33	33	33	32	32
t = 20 *	0	0	0	0	0	0	0	0	0	0	0

u(t,i)

i =	0	1	2	3	4	5	6	7	8	9	10
t= 0 *	2	2	2	1	1	1	1	1	1	1	1
t= 1 *	2	2	2	1	1	1	1	1	1	1	1
t= 2 *	2	2	1	1	1	1	1	1	1	1	1
t= 3 *	2	2	1	1	1	1	1	1	1	1	1
t= 4 *	2	2	1	1	1	1	1	1	1	1	1
t= 5 *	2	2	1	1	1	1	1	1	1	1	1
t= 6 *	2	2	1	1	1	1	1	1	1	1	1
t= 7 *	2	2	1	1	1	1	1	1	1	1	0
t= 8 *	2	2	1	1	1	1	1	1	1	1	0
t= 9 *	2	2	1	1	1	1	1	1	1	0	0
t= 10 *	2	2	1	1	1	1	1	1	0	0	0
t= 11 *	2	2	1	1	1	1	1	1	0	0	0
t= 12 *	2	2	1	1	1	1	1	0	0	0	0
t= 13 *	2	2	1	1	1	1	0	0	0	0	0
t= 14 *	2	1	1	1	1	1	0	0	0	0	0
t= 15 *	2	1	1	1	1	0	0	0	0	0	0
t= 16 *	2	1	1	1	0	0	0	0	0	0	0
t= 17 *	1	1	1	0	0	0	0	0	0	0	0
t= 18 *	1	1	0	0	0	0	0	0	0	0	0
t= 19 *	1	0	0	0	0	0	0	0	0	0	0
t= 20 *	0	0	0	0	0	0	0	0	0	0	0