# Project

## ISEN 622 – Linear Programming

Harish Chellappa

Vishal Venkatesh Ganesh

Tushar Nahar

# Section A

**Indices**:

$N = 1,..,n$

**Parameters**:

$p_t$: unit production cost in period $t$, $t \in N$.

$h_t$: unit storage cost in period $t$, $t \in N$.

$d_t$: demand in period $t$, $t \in N$.

$C$: maximum production capacity in any period (all periods have identical production capacity)

$s_0$: initial inventory

**Variables**:

$x_t$: number of product units produced in period t

$s_t$: number of product units carried in inventory form period t to t+1

**Formulations**:

Min $\sum_{t=1}^{n}(p_t x_t + h_t s_t)$

s.t

$s_t = s_{t-1} + x_t - d_t$ $\quad$ t=1,..,n

$s_0 = 0$

$0 \leq x_t \leq C$ $\quad$ t=1,..,n

$S_t \geq 0$ $\quad$ t=1..,n

# Section B

**Model File**

```
# AMPL model file for LSC
# No. of periods
param n;
set N := {1..n};
# Demand in period t,t∈N
param d {t in N};
# Unit production cost in period t,t∈N
param p {t in N};
# Unit storage cost in period t,t∈N
param h {t in N};
# Maximum production capacity in any period
param C;
# No. units produced in period t,t∈N
var x {t in N} >= 0;
# Inventory at the end of each period
var s {t in 0..n} >= 0;
# Sum of cost of producing x[t] units and storing s[t] units in time t,t∈N
minimize z: sum {t in N} (p[t]*x[t] + h[t]*s[t]);
# Initial inventory is 0
subject to initial_inventory: s[0] = 0;
# Inventory at time t,t∈N
subject to inventory {t in N}: s[t] = s[t-1] + x[t] - d[t];
# Production Limit at time t,t∈N (which is constant for all t,t∈N
subject to production_limit {t in N}: x[t]<=C;
```

**Data File**

```
param n := 90;
param p :=
1	92
2	92
3	104
4	81
5	108
6	98
7	99
8	88
9	113
10	83
11	93
12	88
13	101
14	118
15	85
16	88
17	89
18	109
19	113
20	104
21	90
22	94
23	99
24	102
25	112
26	91
27	93
28	102
29	113
30	103
31	89
32	95
33	110
34	105
35	110
36	87
37	113
38	101
39	118
40	118
41	90
42	113
43	82
44	104
45	91
```

| | |
|---|---|
| 46 | 89 |
| 47 | 108 |
| 48 | 84 |
| 49 | 88 |
| 50 | 107 |
| 51 | 84 |
| 52 | 99 |
| 53 | 114 |
| 54 | 112 |
| 55 | 104 |
| 56 | 91 |
| 57 | 83 |
| 58 | 107 |
| 59 | 85 |
| 60 | 116 |
| 61 | 101 |
| 62 | 106 |
| 63 | 104 |
| 64 | 91 |
| 65 | 101 |
| 66 | 98 |
| 67 | 86 |
| 68 | 112 |
| 69 | 102 |
| 70 | 115 |
| 71 | 83 |
| 72 | 99 |
| 73 | 118 |
| 74 | 83 |
| 75 | 108 |
| 76 | 93 |
| 77 | 91 |
| 78 | 113 |
| 79 | 84 |
| 80 | 96 |
| 81 | 87 |
| 82 | 96 |
| 83 | 111 |
| 84 | 86 |
| 85 | 85 |
| 86 | 104 |
| 87 | 103 |
| 88 | 94 |
| 89 | 96 |
| 90 | 99; |

param h :=

| | |
|---|---|
| 1 | 10 |
| 2 | 10 |
| 3 | 10 |
| 4 | 10 |

| | |
|---|---|
| 5 | 10 |
| 6 | 10 |
| 7 | 10 |
| 8 | 10 |
| 9 | 10 |
| 10 | 10 |
| 11 | 10 |
| 12 | 10 |
| 13 | 10 |
| 14 | 10 |
| 15 | 10 |
| 16 | 10 |
| 17 | 10 |
| 18 | 10 |
| 19 | 10 |
| 20 | 10 |
| 21 | 10 |
| 22 | 10 |
| 23 | 10 |
| 24 | 10 |
| 25 | 10 |
| 26 | 10 |
| 27 | 10 |
| 28 | 10 |
| 29 | 10 |
| 30 | 10 |
| 31 | 10 |
| 32 | 10 |
| 33 | 10 |
| 34 | 10 |
| 35 | 10 |
| 36 | 10 |
| 37 | 10 |
| 38 | 10 |
| 39 | 10 |
| 40 | 10 |
| 41 | 10 |
| 42 | 10 |
| 43 | 10 |
| 44 | 10 |
| 45 | 10 |
| 46 | 10 |
| 47 | 10 |
| 48 | 10 |
| 49 | 10 |
| 50 | 10 |
| 51 | 10 |
| 52 | 10 |
| 53 | 10 |
| 54 | 10 |
| 55 | 10 |

```
56	10
57	10
58	10
59	10
60	10
61	10
62	10
63	10
64	10
65	10
66	10
67	10
68	10
69	10
70	10
71	10
72	10
73	10
74	10
75	10
76	10
77	10
78	10
79	10
80	10
81	10
82	10
83	10
84	10
85	10
86	10
87	10
88	10
89	10
90	10;

param d :=
1	134
2	118
3	33
4	185
5	147
6	144
7	123
8	181
9	47
10	74
11	16
12	115
13	109
14	107
```

| | |
|---|---|
| 15 | 109 |
| 16 | 126 |
| 17 | 175 |
| 18 | 144 |
| 19 | 122 |
| 20 | 70 |
| 21 | 135 |
| 22 | 176 |
| 23 | 174 |
| 24 | 31 |
| 25 | 115 |
| 26 | 170 |
| 27 | 108 |
| 28 | 153 |
| 29 | 17 |
| 30 | 149 |
| 31 | 148 |
| 32 | 47 |
| 33 | 73 |
| 34 | 149 |
| 35 | 161 |
| 36 | 46 |
| 37 | 179 |
| 38 | 17 |
| 39 | 40 |
| 40 | 121 |
| 41 | 40 |
| 42 | 155 |
| 43 | 116 |
| 44 | 89 |
| 45 | 46 |
| 46 | 37 |
| 47 | 80 |
| 48 | 38 |
| 49 | 86 |
| 50 | 94 |
| 51 | 114 |
| 52 | 94 |
| 53 | 81 |
| 54 | 76 |
| 55 | 165 |
| 56 | 100 |
| 57 | 149 |
| 58 | 72 |
| 59 | 61 |
| 60 | 102 |
| 61 | 26 |
| 62 | 100 |
| 63 | 24 |
| 64 | 131 |
| 65 | 99 |

| | |
|---|---|
| 66 | 91 |
| 67 | 20 |
| 68 | 76 |
| 69 | 104 |
| 70 | 112 |
| 71 | 160 |
| 72 | 37 |
| 73 | 105 |
| 74 | 63 |
| 75 | 159 |
| 76 | 39 |
| 77 | 144 |
| 78 | 62 |
| 79 | 171 |
| 80 | 56 |
| 81 | 68 |
| 82 | 46 |
| 83 | 104 |
| 84 | 40 |
| 85 | 15 |
| 86 | 61 |
| 87 | 96 |
| 88 | 185 |
| 89 | 133 |
| 90 | 22; |

param C := 180;

**Run File**

```
reset;

option solver cplex;

#option cplex_options 'timing 4';    # For displaying the Input, Output and

                                       #Solve Times in Ticks. Put 'timing 1'
                                     #for getting the value in seconds

option cplex_options 'sensitivity';  # For performing sensitivity analysis

#option 'timelimit= 0.00000023728752'; # For terminating the search after the

                                        #given amount of time

option presolve 0;

model lsc.mod;

data lsc.dat;
```

```
#expand z,initial_inventory,inventory,production_limit;

solve;

display z,x,s>'lsc.out';

display production_limit.up, production_limit.current,
production_limit.down>>'lsc.out';
```

**Output**

z = 853945

| t | x | s |
|---|---|---|
| 1 | 134 | 0 |
| 2 | 156 | 38 |
| 3 | 0 | 5 |
| 4 | 180 | 0 |
| 5 | 147 | 0 |
| 6 | 144 | 0 |
| 7 | 124 | 1 |
| 8 | 180 | 0 |
| 9 | 47 | 0 |
| 10 | 74 | 0 |
| 11 | 16 | 0 |
| 12 | 180 | 65 |
| 13 | 151 | 107 |
| 14 | 0 | 0 |
| 15 | 109 | 0 |
| 16 | 180 | 54 |
| 17 | 180 | 59 |
| 18 | 85 | 0 |
| 19 | 122 | 0 |
| 20 | 70 | 0 |
| 21 | 135 | 0 |
| 22 | 176 | 0 |
| 23 | 174 | 0 |
| 24 | 31 | 0 |
| 25 | 115 | 0 |
| 26 | 170 | 0 |
| 27 | 108 | 0 |
| 28 | 170 | 17 |
| 29 | 0 | 0 |
| 30 | 149 | 0 |

| | | |
|---|---|---|
| 31 | 148 | 0 |
| 32 | 120 | 73 |
| 33 | 0 | 0 |
| 34 | 149 | 0 |
| 35 | 161 | 0 |
| 36 | 180 | 134 |
| 37 | 45 | 0 |
| 38 | 57 | 40 |
| 39 | 0 | 0 |
| 40 | 121 | 0 |
| 41 | 180 | 140 |
| 42 | 15 | 0 |
| 43 | 180 | 64 |
| 44 | 25 | 0 |
| 45 | 46 | 0 |
| 46 | 117 | 80 |
| 47 | 0 | 0 |
| 48 | 38 | 0 |
| 49 | 180 | 94 |
| 50 | 0 | 0 |
| 51 | 180 | 66 |
| 52 | 109 | 81 |
| 53 | 0 | 0 |
| 54 | 76 | 0 |
| 55 | 165 | 0 |
| 56 | 100 | 0 |
| 57 | 180 | 31 |
| 58 | 41 | 0 |
| 59 | 163 | 102 |
| 60 | 0 | 0 |
| 61 | 26 | 0 |
| 62 | 100 | 0 |
| 63 | 24 | 0 |
| 64 | 180 | 49 |
| 65 | 50 | 0 |
| 66 | 91 | 0 |
| 67 | 96 | 76 |
| 68 | 0 | 0 |
| 69 | 180 | 76 |
| 70 | 36 | 0 |
| 71 | 180 | 20 |
| 72 | 122 | 105 |
| 73 | 0 | 0 |

| | | |
|---|---|---|
| 74 | 180 | 117 |
| 75 | 42 | 0 |
| 76 | 39 | 0 |
| 77 | 180 | 36 |
| 78 | 26 | 0 |
| 79 | 180 | 9 |
| 80 | 47 | 0 |
| 81 | 68 | 0 |
| 82 | 150 | 104 |
| 83 | 0 | 0 |
| 84 | 40 | 0 |
| 85 | 76 | 61 |
| 86 | 0 | 0 |
| 87 | 101 | 5 |
| 88 | 180 | 0 |
| 89 | 133 | 0 |
| 90 | 22 | 0 |

| t | production_limit.up | production_limit.current | production_limit.down |
|---|---|---|---|
| 1 | 1e+20 | 180 | 134 |
| 2 | 1e+20 | 180 | 156 |
| 3 | 1e+20 | 180 | 0 |
| 4 | 185 | 180 | 156 |
| 5 | 1e+20 | 180 | 147 |
| 6 | 1e+20 | 180 | 144 |
| 7 | 1e+20 | 180 | 124 |
| 8 | 181 | 180 | 124 |
| 9 | 1e+20 | 180 | 47 |
| 10 | 1e+20 | 180 | 74 |
| 11 | 1e+20 | 180 | 16 |
| 12 | 331 | 180 | 151 |
| 13 | 1e+20 | 180 | 151 |
| 14 | 1e+20 | 180 | 0 |
| 15 | 1e+20 | 180 | 109 |
| 16 | 265 | 180 | 126 |
| 17 | 265 | 180 | 121 |
| 18 | 1e+20 | 180 | 85 |
| 19 | 1e+20 | 180 | 122 |
| 20 | 1e+20 | 180 | 70 |
| 21 | 1e+20 | 180 | 135 |
| 22 | 1e+20 | 180 | 176 |

| | | | |
|---|---|---|---|
| 23 | 1e+20 | 180 | 174 |
| 24 | 1e+20 | 180 | 31 |
| 25 | 1e+20 | 180 | 115 |
| 26 | 1e+20 | 180 | 170 |
| 27 | 1e+20 | 180 | 108 |
| 28 | 1e+20 | 180 | 170 |
| 29 | 1e+20 | 180 | 0 |
| 30 | 1e+20 | 180 | 149 |
| 31 | 1e+20 | 180 | 148 |
| 32 | 1e+20 | 180 | 120 |
| 33 | 1e+20 | 180 | 0 |
| 34 | 1e+20 | 180 | 149 |
| 35 | 1e+20 | 180 | 161 |
| 36 | 225 | 180 | 46 |
| 37 | 1e+20 | 180 | 45 |
| 38 | 1e+20 | 180 | 57 |
| 39 | 1e+20 | 180 | 0 |
| 40 | 1e+20 | 180 | 121 |
| 41 | 195 | 180 | 40 |
| 42 | 1e+20 | 180 | 15 |
| 43 | 205 | 180 | 116 |
| 44 | 1e+20 | 180 | 25 |
| 45 | 1e+20 | 180 | 46 |
| 46 | 1e+20 | 180 | 117 |
| 47 | 1e+20 | 180 | 0 |
| 48 | 1e+20 | 180 | 38 |
| 49 | 180 | 180 | 38 |
| 50 | 1e+20 | 180 | 0 |
| 51 | 289 | 180 | 114 |
| 52 | 1e+20 | 180 | 109 |
| 53 | 1e+20 | 180 | 0 |
| 54 | 1e+20 | 180 | 76 |
| 55 | 1e+20 | 180 | 165 |
| 56 | 1e+20 | 180 | 100 |
| 57 | 221 | 180 | 149 |
| 58 | 1e+20 | 180 | 41 |
| 59 | 1e+20 | 180 | 163 |
| 60 | 1e+20 | 180 | 0 |
| 61 | 1e+20 | 180 | 26 |
| 62 | 1e+20 | 180 | 100 |
| 63 | 1e+20 | 180 | 24 |
| 64 | 230 | 180 | 131 |
| 65 | 1e+20 | 180 | 50 |

| | | | |
|---|---|---|---|
| 66 | 1e+20 | 180 | 91 |
| 67 | 1e+20 | 180 | 96 |
| 68 | 1e+20 | 180 | 0 |
| 69 | 216 | 180 | 104 |
| 70 | 1e+20 | 180 | 36 |
| 71 | 302 | 180 | 160 |
| 72 | 1e+20 | 180 | 122 |
| 73 | 1e+20 | 180 | 0 |
| 74 | 222 | 180 | 63 |
| 75 | 1e+20 | 180 | 42 |
| 76 | 1e+20 | 180 | 39 |
| 77 | 206 | 180 | 144 |
| 78 | 1e+20 | 180 | 26 |
| 79 | 227 | 180 | 171 |
| 80 | 1e+20 | 180 | 47 |
| 81 | 1e+20 | 180 | 68 |
| 82 | 1e+20 | 180 | 150 |
| 83 | 1e+20 | 180 | 0 |
| 84 | 1e+20 | 180 | 40 |
| 85 | 1e+20 | 180 | 76 |
| 86 | 1e+20 | 180 | 0 |
| 87 | 1e+20 | 180 | 101 |
| 88 | 185 | 180 | 101 |
| 89 | 1e+20 | 180 | 133 |
| 90 | 1e+20 | 180 | 22 |

# Section C

1. **What was the version of CPLEX you used in solving LSC problem?**

   CPLEX 12.7.1.0

2. **What was the specifications of the computer you used to run LSC AMPL model?**

   Intel Core i7-7$^{th}$ generation, 2.81GHz

   8 GB DDR-4 RAM

   4 GB DDR-5 NVIDIA GEFORCE GTX 1050

3. **What was the input time, solve time, and output time to solve LSC on this computer? What was the total time?**

   Times (seconds):

   Input = 0
   Solve = 0
   Output =0

   Total = 0

   Times (ticks);

   Input = 0.00516224
   Solve = 0.468297
   Output = 0.0011158

   Total = 0.47457504

   1 tick = $10^{-6}$ seconds

4. **What is the optimal objective function value for LSC?**

   z = 853945

5. **Use the sensitivity ranging capability of AMPLCPLEX to determine for what range of parameter $CC$, the optimal solution remains optimal.**

```
option cplex_options 'sensitivity';  # For performing sensitivity
                                           #analysis

option presolve 0;

display production_limit.up, production_limit.current,
production_limit.down>>'lsc.out';
```

**<u>Output for the same</u>**

| t | production_limit.up | production_limit.current | production_limit.down |
|---|---|---|---|
| 1 | 1e+20 | 180 | 134 |
| 2 | 1e+20 | 180 | 156 |
| 3 | 1e+20 | 180 | 0 |
| 4 | 185 | 180 | 156 |
| 5 | 1e+20 | 180 | 147 |
| 6 | 1e+20 | 180 | 144 |
| 7 | 1e+20 | 180 | 124 |
| 8 | 181 | 180 | 124 |
| 9 | 1e+20 | 180 | 47 |
| 10 | 1e+20 | 180 | 74 |
| 11 | 1e+20 | 180 | 16 |
| 12 | 331 | 180 | 151 |
| 13 | 1e+20 | 180 | 151 |
| 14 | 1e+20 | 180 | 0 |
| 15 | 1e+20 | 180 | 109 |
| 16 | 265 | 180 | 126 |
| 17 | 265 | 180 | 121 |
| 18 | 1e+20 | 180 | 85 |
| 19 | 1e+20 | 180 | 122 |
| 20 | 1e+20 | 180 | 70 |
| 21 | 1e+20 | 180 | 135 |
| 22 | 1e+20 | 180 | 176 |
| 23 | 1e+20 | 180 | 174 |
| 24 | 1e+20 | 180 | 31 |
| 25 | 1e+20 | 180 | 115 |
| 26 | 1e+20 | 180 | 170 |
| 27 | 1e+20 | 180 | 108 |
| 28 | 1e+20 | 180 | 170 |
| 29 | 1e+20 | 180 | 0 |
| 30 | 1e+20 | 180 | 149 |

| | | | |
|---|---|---|---|
| 31 | 1e+20 | 180 | 148 |
| 32 | 1e+20 | 180 | 120 |
| 33 | 1e+20 | 180 | 0 |
| 34 | 1e+20 | 180 | 149 |
| 35 | 1e+20 | 180 | 161 |
| 36 | 225 | 180 | 46 |
| 37 | 1e+20 | 180 | 45 |
| 38 | 1e+20 | 180 | 57 |
| 39 | 1e+20 | 180 | 0 |
| 40 | 1e+20 | 180 | 121 |
| 41 | 195 | 180 | 40 |
| 42 | 1e+20 | 180 | 15 |
| 43 | 205 | 180 | 116 |
| 44 | 1e+20 | 180 | 25 |
| 45 | 1e+20 | 180 | 46 |
| 46 | 1e+20 | 180 | 117 |
| 47 | 1e+20 | 180 | 0 |
| 48 | 1e+20 | 180 | 38 |
| 49 | 180 | 180 | 38 |
| 50 | 1e+20 | 180 | 0 |
| 51 | 289 | 180 | 114 |
| 52 | 1e+20 | 180 | 109 |
| 53 | 1e+20 | 180 | 0 |
| 54 | 1e+20 | 180 | 76 |
| 55 | 1e+20 | 180 | 165 |
| 56 | 1e+20 | 180 | 100 |
| 57 | 221 | 180 | 149 |
| 58 | 1e+20 | 180 | 41 |
| 59 | 1e+20 | 180 | 163 |
| 60 | 1e+20 | 180 | 0 |
| 61 | 1e+20 | 180 | 26 |
| 62 | 1e+20 | 180 | 100 |
| 63 | 1e+20 | 180 | 24 |
| 64 | 230 | 180 | 131 |
| 65 | 1e+20 | 180 | 50 |
| 66 | 1e+20 | 180 | 91 |
| 67 | 1e+20 | 180 | 96 |
| 68 | 1e+20 | 180 | 0 |
| 69 | 216 | 180 | 104 |
| 70 | 1e+20 | 180 | 36 |
| 71 | 302 | 180 | 160 |
| 72 | 1e+20 | 180 | 122 |
| 73 | 1e+20 | 180 | 0 |

| | | | |
|---|---|---|---|
| 74 | 222 | 180 | 63 |
| 75 | 1e+20 | 180 | 42 |
| 76 | 1e+20 | 180 | 39 |
| 77 | 206 | 180 | 144 |
| 78 | 1e+20 | 180 | 26 |
| 79 | 227 | 180 | 171 |
| 80 | 1e+20 | 180 | 47 |
| 81 | 1e+20 | 180 | 68 |
| 82 | 1e+20 | 180 | 150 |
| 83 | 1e+20 | 180 | 0 |
| 84 | 1e+20 | 180 | 40 |
| 85 | 1e+20 | 180 | 76 |
| 86 | 1e+20 | 180 | 0 |
| 87 | 1e+20 | 180 | 101 |
| 88 | 185 | 180 | 101 |
| 89 | 1e+20 | 180 | 133 |
| 90 | 1e+20 | 180 | 22 |

6. **Set the time limit of CPLEX to half the total run time you report in part 3 and re-solve your model. Report the objective value for the best solution found within this time limit.**

   Objective value found within this time limit is same as in question C.3, i.e. z=853945

# Section D

**Indices**:

$N = 1,..,n$

**Parameters**:

$p_t$: unit production cost in period $t$, $t \in N$.

$h_t$: unit storage cost in period $t$, $t \in N$.

$d_t$: demand in period $t$, $t \in N$.

$C$: maximum production capacity of each module

$f_t$: unit module cost in each period t, $t \in N$

$s_0$: initial inventory

**Variables**:

$x_t$: number of product units produced in period t

$s_t$: number of product units carried in inventory form period t to t+1

$y_t$: number of modules in each period (takes only integer values)

**Formulations**:

Min $\sum_{t=1}^{n}(p_t x_t + h_t s_t + y_t f_t)$

s.t

$s_t = s_{t-1} + x_t - d_t$ $\quad$ t=1,..,n

$s_0 = 0$

$0 \leq x_t \leq y_t * C$ $\quad$ t=1,..,n

$S_t \geq 0$ $\quad$ t=1..,n

# Section E

**Model File**

```
# AMPL model file for LSCM

# No. of periods

param n;

set N := {1..n};

# Demand in period t,t∈N

param d {t in N};

# Unit production cost in period t,t∈N

param p {t in N};

# Unit module cost in period t,t∈N

param f {t in N};

# Unit storage cost in period t,t∈N

param h {t in N};

# Maximum production capacity in any period

param C;

# No. units produced in period t,t∈N

var x {t in N} >= 0;

# Inventory at the end of each period

var s {t in 0..n} >= 0;

# Number of capacity modules installed in period t,t∈N

var y {t in N} >=0 integer;

# Sum of cost of producing x[t] units, storing s[t] units and cost of y[t] in
time t,t∈N

minimize z: sum {t in N} (p[t]*x[t] + h[t]*s[t] + y[t]*f[t]);

# Initial inventory is 0

subject to initial_inventory: s[0] = 0;

# Inventory at time t,t∈N

subject to inventory {t in N}: s[t] = s[t-1] + x[t] - d[t];

# Production Limit at time t,t∈N

subject to production_limit {t in N}: x[t]<=y[t]*C;
```

**Data File**

param n := 90;
param p :=

| | |
|---|---|
| 1 | 92 |
| 2 | 92 |
| 3 | 104 |
| 4 | 81 |
| 5 | 108 |
| 6 | 98 |
| 7 | 99 |
| 8 | 88 |
| 9 | 113 |
| 10 | 83 |
| 11 | 93 |
| 12 | 88 |
| 13 | 101 |
| 14 | 118 |
| 15 | 85 |
| 16 | 88 |
| 17 | 89 |
| 18 | 109 |
| 19 | 113 |
| 20 | 104 |
| 21 | 90 |
| 22 | 94 |
| 23 | 99 |
| 24 | 102 |
| 25 | 112 |
| 26 | 91 |
| 27 | 93 |
| 28 | 102 |
| 29 | 113 |
| 30 | 103 |
| 31 | 89 |
| 32 | 95 |
| 33 | 110 |
| 34 | 105 |
| 35 | 110 |
| 36 | 87 |
| 37 | 113 |
| 38 | 101 |
| 39 | 118 |
| 40 | 118 |
| 41 | 90 |
| 42 | 113 |
| 43 | 82 |
| 44 | 104 |
| 45 | 91 |

```
46	89
47	108
48	84
49	88
50	107
51	84
52	99
53	114
54	112
55	104
56	91
57	83
58	107
59	85
60	116
61	101
62	106
63	104
64	91
65	101
66	98
67	86
68	112
69	102
70	115
71	83
72	99
73	118
74	83
75	108
76	93
77	91
78	113
79	84
80	96
81	87
82	96
83	111
84	86
85	85
86	104
87	103
88	94
89	96
90	99;

param h :=
1	10
2	10
3	10
4	10
```

| | |
|---|---|
| 5 | 10 |
| 6 | 10 |
| 7 | 10 |
| 8 | 10 |
| 9 | 10 |
| 10 | 10 |
| 11 | 10 |
| 12 | 10 |
| 13 | 10 |
| 14 | 10 |
| 15 | 10 |
| 16 | 10 |
| 17 | 10 |
| 18 | 10 |
| 19 | 10 |
| 20 | 10 |
| 21 | 10 |
| 22 | 10 |
| 23 | 10 |
| 24 | 10 |
| 25 | 10 |
| 26 | 10 |
| 27 | 10 |
| 28 | 10 |
| 29 | 10 |
| 30 | 10 |
| 31 | 10 |
| 32 | 10 |
| 33 | 10 |
| 34 | 10 |
| 35 | 10 |
| 36 | 10 |
| 37 | 10 |
| 38 | 10 |
| 39 | 10 |
| 40 | 10 |
| 41 | 10 |
| 42 | 10 |
| 43 | 10 |
| 44 | 10 |
| 45 | 10 |
| 46 | 10 |
| 47 | 10 |
| 48 | 10 |
| 49 | 10 |
| 50 | 10 |
| 51 | 10 |
| 52 | 10 |
| 53 | 10 |
| 54 | 10 |
| 55 | 10 |

```
56	10
57	10
58	10
59	10
60	10
61	10
62	10
63	10
64	10
65	10
66	10
67	10
68	10
69	10
70	10
71	10
72	10
73	10
74	10
75	10
76	10
77	10
78	10
79	10
80	10
81	10
82	10
83	10
84	10
85	10
86	10
87	10
88	10
89	10
90	10;

param d :=
1	134
2	118
3	33
4	185
5	147
6	144
7	123
8	181
9	47
10	74
11	16
12	115
13	109
14	107
```

| | |
|---|---|
| 15 | 109 |
| 16 | 126 |
| 17 | 175 |
| 18 | 144 |
| 19 | 122 |
| 20 | 70 |
| 21 | 135 |
| 22 | 176 |
| 23 | 174 |
| 24 | 31 |
| 25 | 115 |
| 26 | 170 |
| 27 | 108 |
| 28 | 153 |
| 29 | 17 |
| 30 | 149 |
| 31 | 148 |
| 32 | 47 |
| 33 | 73 |
| 34 | 149 |
| 35 | 161 |
| 36 | 46 |
| 37 | 179 |
| 38 | 17 |
| 39 | 40 |
| 40 | 121 |
| 41 | 40 |
| 42 | 155 |
| 43 | 116 |
| 44 | 89 |
| 45 | 46 |
| 46 | 37 |
| 47 | 80 |
| 48 | 38 |
| 49 | 86 |
| 50 | 94 |
| 51 | 114 |
| 52 | 94 |
| 53 | 81 |
| 54 | 76 |
| 55 | 165 |
| 56 | 100 |
| 57 | 149 |
| 58 | 72 |
| 59 | 61 |
| 60 | 102 |
| 61 | 26 |
| 62 | 100 |
| 63 | 24 |
| 64 | 131 |
| 65 | 99 |

```
66	91
67	20
68	76
69	104
70	112
71	160
72	37
73	105
74	63
75	159
76	39
77	144
78	62
79	171
80	56
81	68
82	46
83	104
84	40
85	15
86	61
87	96
88	185
89	133
90	22;

param C := 180;

param f :=
1	5000
2	5000
3	5000
4	5000
5	5000
6	5000
7	5000
8	5000
9	5000
10	5000
11	5000
12	5000
13	5000
14	5000
15	5000
16	5000
17	5000
18	5000
19	5000
20	5000
21	5000
22	5000
```

| | |
|---|---|
| 23 | 5000 |
| 24 | 5000 |
| 25 | 5000 |
| 26 | 5000 |
| 27 | 5000 |
| 28 | 5000 |
| 29 | 5000 |
| 30 | 5000 |
| 31 | 5000 |
| 32 | 5000 |
| 33 | 5000 |
| 34 | 5000 |
| 35 | 5000 |
| 36 | 5000 |
| 37 | 5000 |
| 38 | 5000 |
| 39 | 5000 |
| 40 | 5000 |
| 41 | 5000 |
| 42 | 5000 |
| 43 | 5000 |
| 44 | 5000 |
| 45 | 5000 |
| 46 | 5000 |
| 47 | 5000 |
| 48 | 5000 |
| 49 | 5000 |
| 50 | 5000 |
| 51 | 5000 |
| 52 | 5000 |
| 53 | 5000 |
| 54 | 5000 |
| 55 | 5000 |
| 56 | 5000 |
| 57 | 5000 |
| 58 | 5000 |
| 59 | 5000 |
| 60 | 5000 |
| 61 | 5000 |
| 62 | 5000 |
| 63 | 5000 |
| 64 | 5000 |
| 65 | 5000 |
| 66 | 5000 |
| 67 | 5000 |
| 68 | 5000 |
| 69 | 5000 |
| 70 | 5000 |
| 71 | 5000 |
| 72 | 5000 |
| 73 | 5000 |

74	5000
75	5000
76	5000
77	5000
78	5000
79	5000
80	5000
81	5000
82	5000
83	5000
84	5000
85	5000
86	5000
87	5000
88	5000
89	5000
90	5000;

**Run File**

```
reset;

option solver cplex;

option cplex_options 'timing 1';  # For displaying the Input, Output andSolve Times

model lscm.mod;

data lscm.dat;

#expand z,initial_inventory,inventory,production_limit;

solve;

display z,x,s,y>'lscm.out';
```

**Output**

z = 1134640

| t | x | s | y |
|---|---|---|---|
| 1 | 164 | 30 | 1 |
| 2 | 180 | 92 | 1 |
| 3 | 0 | 59 | 0 |
| 4 | 360 | 234 | 2 |
| 5 | 0 | 87 | 0 |
| 6 | 180 | 123 | 1 |

| | | | |
|---|---|---|---|
| 7 | 0 | 0 | 0 |
| 8 | 318 | 137 | 2 |
| 9 | 0 | 90 | 0 |
| 10 | 0 | 16 | 0 |
| 11 | 0 | 0 | 0 |
| 12 | 357 | 242 | 2 |
| 13 | 0 | 133 | 0 |
| 14 | 0 | 26 | 0 |
| 15 | 180 | 97 | 1 |
| 16 | 180 | 151 | 1 |
| 17 | 360 | 336 | 2 |
| 18 | 0 | 192 | 0 |
| 19 | 0 | 70 | 0 |
| 20 | 0 | 0 | 0 |
| 21 | 135 | 0 | 1 |
| 22 | 176 | 0 | 1 |
| 23 | 174 | 0 | 1 |
| 24 | 146 | 115 | 1 |
| 25 | 0 | 0 | 0 |
| 26 | 170 | 0 | 1 |
| 27 | 163 | 55 | 1 |
| 28 | 180 | 82 | 1 |
| 29 | 0 | 65 | 0 |
| 30 | 180 | 96 | 1 |
| 31 | 180 | 128 | 1 |
| 32 | 0 | 81 | 0 |
| 33 | 0 | 8 | 0 |
| 34 | 180 | 39 | 1 |
| 35 | 180 | 58 | 1 |
| 36 | 180 | 192 | 1 |
| 37 | 0 | 13 | 0 |
| 38 | 180 | 176 | 1 |
| 39 | 0 | 136 | 0 |
| 40 | 0 | 15 | 0 |
| 41 | 180 | 155 | 1 |
| 42 | 0 | 0 | 0 |
| 43 | 251 | 135 | 2 |
| 44 | 0 | 46 | 0 |
| 45 | 0 | 0 | 0 |
| 46 | 160 | 123 | 1 |
| 47 | 0 | 43 | 0 |
| 48 | 0 | 5 | 0 |
| 49 | 180 | 99 | 1 |

| | | | |
|---|---|---|---|
| 50 | 0 | 5 | 0 |
| 51 | 360 | 251 | 2 |
| 52 | 0 | 157 | 0 |
| 53 | 0 | 76 | 0 |
| 54 | 0 | 0 | 0 |
| 55 | 165 | 0 | 1 |
| 56 | 150 | 50 | 1 |
| 57 | 180 | 81 | 1 |
| 58 | 0 | 9 | 0 |
| 59 | 180 | 128 | 1 |
| 60 | 0 | 26 | 0 |
| 61 | 0 | 0 | 0 |
| 62 | 124 | 24 | 1 |
| 63 | 0 | 0 | 0 |
| 64 | 321 | 190 | 2 |
| 65 | 0 | 91 | 0 |
| 66 | 0 | 0 | 0 |
| 67 | 132 | 112 | 1 |
| 68 | 0 | 36 | 0 |
| 69 | 180 | 112 | 1 |
| 70 | 0 | 0 | 0 |
| 71 | 302 | 142 | 2 |
| 72 | 0 | 105 | 0 |
| 73 | 0 | 0 | 0 |
| 74 | 287 | 224 | 2 |
| 75 | 0 | 65 | 0 |
| 76 | 0 | 26 | 0 |
| 77 | 180 | 62 | 1 |
| 78 | 0 | 0 | 0 |
| 79 | 305 | 134 | 2 |
| 80 | 0 | 78 | 0 |
| 81 | 0 | 10 | 0 |
| 82 | 180 | 144 | 1 |
| 83 | 0 | 40 | 0 |
| 84 | 0 | 0 | 0 |
| 85 | 177 | 162 | 1 |
| 86 | 0 | 101 | 0 |
| 87 | 0 | 5 | 0 |
| 88 | 180 | 0 | 1 |
| 89 | 155 | 22 | 1 |
| 90 | 0 | 0 | 0 |

# Section F

**1.Solve LSCM on the same computer on which you solved LSC and compare its total running time with that of LSC.**

Input = 0

Solve = 4.78125

Output = 0

**2. What is the optimal objective function value for LSCM? Compare this optimal objective value with the optimal objective value of LSC plus 5000$n$ (since in LSC every period has a capacity of $C$, or equivalent to one module, the total cost of capacity installation would be 5000$n$ if we assume each module in LSC also costs 5000). Theoretically argue why the optimal objective of LSCM is larger or smaller than that of LSC plus 5000$n$.**

The optimal objective value of LSCM is 1134640

The optimal objective value of LSC + 5000*n is 1303940

The optimal objective of LSCM will always be equal to or smaller than the optimal objective cost of LSC plus 5000n.The LSC after adding 5000 n can be represented as the current LSCM with an additional constraint , $y_t$=1. Adding a constraint to an LP either reduces or doesn't change the optimal objective value. In the current case (LSC plus 5000n/LSCM with $y_t$=1) since the feasible region reduces (when compared with LSCM) and the optimal solution of LSCM is not a part of the current feasible region of LSC(LSCM with $y_t$=1) and hence the objective value of LSC(LSCM with $y_t$=1) is larger than LSCM.
When comparing LSC with LSCM for the current data, we can see that it is advantageous to produce the items in periods with lower production costs and hold it as inventory for satisfying demand in subsequent periods rather than setting up production and producing items in every period. Therefore the cost of LSCM is lower than the cost of LSC.