Aggregate Production Planning: Numerical example - Solution

The numerical example and the notation are given on the slides of the lecture. Here the solution with the software LINGO is presented. An educational version can be downloaded from www.lindo.com.

Model programmed in LINGO

```
Model:
Sets:
Period /1..12/:C,O,u;
Product /1..2/:IInit;
ProdPer (Product,Period): X,I,D,h;
Endsets

! Objective Function;
MIN = @Sum(ProdPer: h*I) + @Sum(Period: u * O);

!Inventory Balance Equations;
@for(ProdPer(J,T) | T #NE# 1:
I(J,T) - I(J,T-1) - X(J,T)  + D(J,T)   = 0);
@for(ProdPer(J,T) | T #EQ# 1:
I(J,T) - IInit(J) - X(J,T)  + D(J,T)   = 0);

!Capacity Constraints;
@for(Period(T):
@Sum(Product(J): X(J,T)) - O(T) < C(T));

!Definition Regular Capacity;
@for(Period(T):
C(T) = 200);

!Definition Safety Stock;
@for(Period(T):
I(1,T) > 10);
@for(Period(T):
I(2,T) > 20);


Data:
D=100,90,85,70,105,120,140,120,120,110,105,100,
120,140,120,120,110,105,100,100,90,85,70,105;
h=4,4,4,4,4,4,4,4,4,4,4,4,
5,5,5,5,5,5,5,5,5,5,5,5;
u=20,20,20,20,20,20,20,20,20,20,20,20;
IInit=20,80;

Enddata

End
```

Model formulation, generated by LINGO

```
MODEL:
 [_1] MIN= 4 * I_1_1 + 4 * I_1_2 + 4 * I_1_3 + 4 * I_1_4 + 4 * I_1_5 + 4
 * I_1_6 + 4 * I_1_7 + 4 * I_1_8 + 4 * I_1_9 + 4 * I_1_10 + 4 * I_1_11 +
 4 * I_1_12 + 5 * I_2_1 + 5 * I_2_2 + 5 * I_2_3 + 5 * I_2_4 + 5 * I_2_5 +
 5 * I_2_6 + 5 * I_2_7 + 5 * I_2_8 + 5 * I_2_9 + 5 * I_2_10 + 5 * I_2_11
 + 5 * I_2_12 + 20 * O_1 + 20 * O_2 + 20 * O_3 + 20 * O_4 + 20 * O_5 + 20
 * O_6 + 20 * O_7 + 20 * O_8 + 20 * O_9 + 20 * O_10 + 20 * O_11 + 20 *
 O_12 ;
 [_2] - I_1_1 - X_1_2 + I_1_2 = - 90 ;
 [_3] - I_1_2 - X_1_3 + I_1_3 = - 85 ;
 [_4] - I_1_3 - X_1_4 + I_1_4 = - 70 ;
 [_5] - I_1_4 - X_1_5 + I_1_5 = - 105 ;
 [_6] - I_1_5 - X_1_6 + I_1_6 = - 120 ;
 [_7] - I_1_6 - X_1_7 + I_1_7 = - 140 ;
 [_8] - I_1_7 - X_1_8 + I_1_8 = - 120 ;
 [_9] - I_1_8 - X_1_9 + I_1_9 = - 120 ;
 [_10] - I_1_9 - X_1_10 + I_1_10 = - 110 ;
 [_11] - I_1_10 - X_1_11 + I_1_11 = - 105 ;
 [_12] - I_1_11 - X_1_12 + I_1_12 = - 100 ;
 [_13] - I_2_1 - X_2_2 + I_2_2 = - 140 ;
 [_14] - I_2_2 - X_2_3 + I_2_3 = - 120 ;
 [_15] - I_2_3 - X_2_4 + I_2_4 = - 120 ;
 [_16] - I_2_4 - X_2_5 + I_2_5 = - 110 ;
 [_17] - I_2_5 - X_2_6 + I_2_6 = - 105 ;
 [_18] - I_2_6 - X_2_7 + I_2_7 = - 100 ;
 [_19] - I_2_7 - X_2_8 + I_2_8 = - 100 ;
 [_20] - I_2_8 - X_2_9 + I_2_9 = - 90 ;
 [_21] - I_2_9 - X_2_10 + I_2_10 = - 85 ;
 [_22] - I_2_10 - X_2_11 + I_2_11 = - 70 ;
 [_23] - I_2_11 - X_2_12 + I_2_12 = - 105 ;
 [_24] - X_1_1 + I_1_1 = - 80 ;
 [_25] - X_2_1 + I_2_1 = - 40 ;
 [_26] X_1_1 + X_2_1 - O_1 <= 200 ;
 [_27] X_1_2 + X_2_2 - O_2 <= 200 ;
 [_28] X_1_3 + X_2_3 - O_3 <= 200 ;
 [_29] X_1_4 + X_2_4 - O_4 <= 200 ;
 [_30] X_1_5 + X_2_5 - O_5 <= 200 ;
 [_31] X_1_6 + X_2_6 - O_6 <= 200 ;
 [_32] X_1_7 + X_2_7 - O_7 <= 200 ;
 [_33] X_1_8 + X_2_8 - O_8 <= 200 ;
 [_34] X_1_9 + X_2_9 - O_9 <= 200 ;
 [_35] X_1_10 + X_2_10 - O_10 <= 200 ;
 [_36] X_1_11 + X_2_11 - O_11 <= 200 ;
 [_37] X_1_12 + X_2_12 - O_12 <= 200 ;
 [_50] I_1_1 >= 10 ;
 [_51] I_1_2 >= 10 ;
 [_52] I_1_3 >= 10 ;
 [_53] I_1_4 >= 10 ;
 [_54] I_1_5 >= 10 ;
 [_55] I_1_6 >= 10 ;
 [_56] I_1_7 >= 10 ;
 [_57] I_1_8 >= 10 ;
 [_58] I_1_9 >= 10 ;
 [_59] I_1_10 >= 10 ;
 [_60] I_1_11 >= 10 ;
 [_61] I_1_12 >= 10 ;
 [_62] I_2_1 >= 20 ;
 [_63] I_2_2 >= 20 ;
 [_64] I_2_3 >= 20 ;
```

```
 [_65] I_2_4 >= 20 ;
 [_66] I_2_5 >= 20 ;
 [_67] I_2_6 >= 20 ;
 [_68] I_2_7 >= 20 ;
 [_69] I_2_8 >= 20 ;
 [_70] I_2_9 >= 20 ;
 [_71] I_2_10 >= 20 ;
 [_72] I_2_11 >= 20 ;
 [_73] I_2_12 >= 20 ;
END
```

This is a Linear Programming (LP) model: All variables are real-valued, all functions are linear.

Comments on the Syntax:

- The Right-hand side (RHS) is always a constant (no variables are allowed on the RHS in this notation).
- “<” and “<=” are considered identical due to the real-valued variables.
- The model can also be input directly (without having it generated) in a straightforward (LINDO) syntax.

Model Solution

```
Global optimal solution found.
Objective value:                              3880.000
Infeasibilities:                              0.000000
Total solver iterations:                            34


                     Variable           Value        Reduced Cost
                        C( 1)        200.0000            0.000000
                        C( 2)        200.0000            0.000000
                        C( 3)        200.0000            0.000000
                        C( 4)        200.0000            0.000000
                        C( 5)        200.0000            0.000000
                        C( 6)        200.0000            0.000000
                        C( 7)        200.0000            0.000000
                        C( 8)        200.0000            0.000000
                        C( 9)        200.0000            0.000000
                       C( 10)        200.0000            0.000000
                       C( 11)        200.0000            0.000000
                       C( 12)        200.0000            0.000000
                        O( 1)        0.000000            20.00000
                        O( 2)        0.000000            16.00000
                        O( 3)        0.000000            12.00000
                        O( 4)        0.000000            8.000000
                        O( 5)        0.000000            4.000000
                        O( 6)        25.00000            0.000000
                        O( 7)        40.00000            0.000000
                        O( 8)        20.00000            0.000000
                        O( 9)        10.00000            0.000000
                       O( 10)        0.000000            20.00000
                       O( 11)        0.000000            20.00000
                       O( 12)        0.000000            16.00000
                        U( 1)        20.00000            0.000000
```

```
          U( 2)        20.00000            0.000000
          U( 3)        20.00000            0.000000
          U( 4)        20.00000            0.000000
          U( 5)        20.00000            0.000000
          U( 6)        20.00000            0.000000
          U( 7)        20.00000            0.000000
          U( 8)        20.00000            0.000000
          U( 9)        20.00000            0.000000
         U( 10)        20.00000            0.000000
         U( 11)        20.00000            0.000000
         U( 12)        20.00000            0.000000
      IINIT( 1)        20.00000            0.000000
      IINIT( 2)        80.00000            0.000000
       X( 1, 1)        130.0000            0.000000
       X( 1, 2)        60.00000            0.000000
       X( 1, 3)        80.00000            0.000000
       X( 1, 4)        80.00000            0.000000
       X( 1, 5)        90.00000            0.000000
       X( 1, 6)        120.0000            0.000000
       X( 1, 7)        140.0000            0.000000
       X( 1, 8)        120.0000            0.000000
       X( 1, 9)        120.0000            0.000000
      X( 1, 10)        110.0000            0.000000
      X( 1, 11)        110.0000            0.000000
      X( 1, 12)        95.00000            0.000000
       X( 2, 1)        60.00000            0.000000
       X( 2, 2)        140.0000            0.000000
       X( 2, 3)        120.0000            0.000000
       X( 2, 4)        120.0000            0.000000
       X( 2, 5)        110.0000            0.000000
       X( 2, 6)        105.0000            0.000000
       X( 2, 7)        100.0000            0.000000
       X( 2, 8)        100.0000            0.000000
       X( 2, 9)        90.00000            0.000000
      X( 2, 10)        85.00000            0.000000
      X( 2, 11)        70.00000            0.000000
      X( 2, 12)        105.0000            0.000000
       I( 1, 1)        50.00000            0.000000
       I( 1, 2)        20.00000            0.000000
       I( 1, 3)        15.00000            0.000000
       I( 1, 4)        25.00000            0.000000
       I( 1, 5)        10.00000            0.000000
       I( 1, 6)        10.00000            0.000000
       I( 1, 7)        10.00000            0.000000
       I( 1, 8)        10.00000            0.000000
       I( 1, 9)        10.00000            0.000000
      I( 1, 10)        10.00000            0.000000
      I( 1, 11)        15.00000            0.000000
      I( 1, 12)        10.00000            0.000000
       I( 2, 1)        20.00000            0.000000
       I( 2, 2)        20.00000            0.000000
       I( 2, 3)        20.00000            0.000000
       I( 2, 4)        20.00000            0.000000
       I( 2, 5)        20.00000            0.000000
       I( 2, 6)        20.00000            0.000000
       I( 2, 7)        20.00000            0.000000
       I( 2, 8)        20.00000            0.000000
       I( 2, 9)        20.00000            0.000000
      I( 2, 10)        20.00000            0.000000
      I( 2, 11)        20.00000            0.000000
      I( 2, 12)        20.00000            0.000000
       D( 1, 1)        100.0000            0.000000
```

D(1, 2)	90.00000	0.000000
D(1, 3)	85.00000	0.000000
D(1, 4)	70.00000	0.000000
D(1, 5)	105.0000	0.000000
D(1, 6)	120.0000	0.000000
D(1, 7)	140.0000	0.000000
D(1, 8)	120.0000	0.000000
D(1, 9)	120.0000	0.000000
D(1, 10)	110.0000	0.000000
D(1, 11)	105.0000	0.000000
D(1, 12)	100.0000	0.000000
D(2, 1)	120.0000	0.000000
D(2, 2)	140.0000	0.000000
D(2, 3)	120.0000	0.000000
D(2, 4)	120.0000	0.000000
D(2, 5)	110.0000	0.000000
D(2, 6)	105.0000	0.000000
D(2, 7)	100.0000	0.000000
D(2, 8)	100.0000	0.000000
D(2, 9)	90.00000	0.000000
D(2, 10)	85.00000	0.000000
D(2, 11)	70.00000	0.000000
D(2, 12)	105.0000	0.000000
H(1, 1)	4.000000	0.000000
H(1, 2)	4.000000	0.000000
H(1, 3)	4.000000	0.000000
H(1, 4)	4.000000	0.000000
H(1, 5)	4.000000	0.000000
H(1, 6)	4.000000	0.000000
H(1, 7)	4.000000	0.000000
H(1, 8)	4.000000	0.000000
H(1, 9)	4.000000	0.000000
H(1, 10)	4.000000	0.000000
H(1, 11)	4.000000	0.000000
H(1, 12)	4.000000	0.000000
H(2, 1)	5.000000	0.000000
H(2, 2)	5.000000	0.000000
H(2, 3)	5.000000	0.000000
H(2, 4)	5.000000	0.000000
H(2, 5)	5.000000	0.000000
H(2, 6)	5.000000	0.000000
H(2, 7)	5.000000	0.000000
H(2, 8)	5.000000	0.000000
H(2, 9)	5.000000	0.000000
H(2, 10)	5.000000	0.000000
H(2, 11)	5.000000	0.000000
H(2, 12)	5.000000	0.000000

Row	Slack or Surplus	Dual Price
1	3880.000	-1.000000
2	0.000000	4.000000
3	0.000000	8.000000
4	0.000000	12.00000
5	0.000000	16.00000
6	0.000000	20.00000
7	0.000000	20.00000
8	0.000000	20.00000
9	0.000000	20.00000
10	0.000000	0.000000
11	0.000000	0.000000
12	0.000000	4.000000
13	0.000000	4.000000

14	0.000000	8.000000
15	0.000000	12.00000
16	0.000000	16.00000
17	0.000000	20.00000
18	0.000000	20.00000
19	0.000000	20.00000
20	0.000000	20.00000
21	0.000000	0.000000
22	0.000000	0.000000
23	0.000000	4.000000
24	0.000000	0.000000
25	0.000000	0.000000
26	10.00000	0.000000
27	0.000000	4.000000
28	0.000000	8.000000
29	0.000000	12.00000
30	0.000000	16.00000
31	0.000000	20.00000
32	0.000000	20.00000
33	0.000000	20.00000
34	0.000000	20.00000
35	5.000000	0.000000
36	20.00000	0.000000
37	0.000000	4.000000
38	0.000000	0.000000
39	0.000000	4.000000
40	0.000000	8.000000
41	0.000000	12.00000
42	0.000000	16.00000
43	0.000000	20.00000
44	0.000000	20.00000
45	0.000000	20.00000
46	0.000000	20.00000
47	0.000000	0.000000
48	0.000000	0.000000
49	0.000000	4.000000
50	40.00000	0.000000
51	10.00000	0.000000
52	5.000000	0.000000
53	15.00000	0.000000
54	0.000000	0.000000
55	0.000000	-4.000000
56	0.000000	-4.000000
57	0.000000	-4.000000
58	0.000000	-24.00000
59	0.000000	-4.000000
60	5.000000	0.000000
61	0.000000	-8.000000
62	0.000000	-1.000000
63	0.000000	-1.000000
64	0.000000	-1.000000
65	0.000000	-1.000000
66	0.000000	-1.000000
67	0.000000	-5.000000
68	0.000000	-5.000000
69	0.000000	-5.000000
70	0.000000	-25.00000
71	0.000000	-5.000000
72	0.000000	-1.000000
73	0.000000	-9.000000

Interpretation of the optimal solution

- Objective function value (total costs) = 3880
- Values of the variables are self-explanatory
- Slack or Surplus and Dual Price: Assigned to each constraint.
- Slack or Surplus: Amount by which the Left-hand side (LHS) deviates from the Right-hand side (RHS) (e.g., unused capacity for capacity constraints).
- Dual Price: Change of the objective function value if the RHS is changed by 1. In the case of a capacity constraint this is the cost improvement if capacity is increased by 1. This is also the maximum price a rational decision maker would be willing to pay for 1 additional unit of the respective capacity, therefore "shadow price". This is a marginal change, the dual price only holds within certain (calculable) bounds of the RHS.
- Reduced Cost: Assigned to each variable. Formally this is the dual price of the non-negativity constraint. For instance, if a product is not produced in a certain period in the optimal solution, the Reduced Cost is the cost increase if 1 unit of the product is forced into the production program.

Slack or Surplus and *Dual Price* of each constraint are linked by C*omplementary slackness*! The essential insight:

- If in the optimal solution a resource (say, resource R_i) has positive slack, that is, the resource is not fully utilized, then the dual price of R_i is zero.
- If the dual price of R_i is greater than zero, then R_i is scarce and is fully utilized in the optimal solution, that is, the slack of R_i is zero.