L 2 (By Peter Lohmander 2009-09-09)

Linear programming. (Linear objective function and linear constraints.) General theory, application examples, analytical solutions and numerical solutions via computer programming.

General LP theory:

In the primal problem, we maximize the variable profit, Π.
We can produce n different products. The production volumes of these products are q_1 to q_n.
The net prices (product prices – variable costs) of these products are p_1 to p_n.
Our machines and other resources (we have m such constraints) have the capacities c_1 to c_m.
$a_{i,j}$ denotes the amount of the resource i that is needed per unit of the product j.

Here is the primal problem:

$$\max \Pi = p_1 q_1 + p_2 q_2 + ... + p_n q_n$$

s.t.

$$a_{1,1} q_1 + a_{1,2} q_2 + ... + a_{1,n} q_n <= c_1$$
$$a_{2,1} q_1 + a_{2,2} q_2 + ... + a_{2,n} q_n <= c_2$$
.....
$$a_{m,1} q_1 + a_{m,2} q_2 + ... + a_{m,n} q_n <= c_m$$

We introduce ”slack variables”, s_i, in the different resource constraints. Of course, the slack variables may not be negative. When this has been done, we have replaced the inequalities by equalities or equations.

$$a_{1,1} q_1 + a_{1,2} q_2 + ... + a_{1,n} q_n + 1 s_1 = c_1$$
$$a_{2,1} q_1 + a_{2,2} q_2 + ... + a_{2,n} q_n + 1 s_2 = c_2$$
.....
$$a_{m,1} q_1 + a_{m,2} q_2 + ... + a_{m,n} q_n + 1 s_m = c_m$$

We may multiply each coefficient in equation i by λ_i.
(We introduce the name “dual variable” or “marginal capacity value” or “shadow price” to denote λ_i.)

$$a_{1,1} q_1 \lambda_1 + a_{1,2} q_2 \lambda_1 + ... + a_{1,n} q_n \lambda_1 + 1 s_1 \lambda_1 = c_1 \lambda_1$$
$$a_{2,1} q_1 \lambda_2 + a_{2,2} q_2 \lambda_2 + ... + a_{2,n} q_n \lambda_2 + 1 s_2 \lambda_2 = c_2 \lambda_2$$
.....
$$a_{m,1} q_1 \lambda_m + a_{m,2} q_2 \lambda_m + ... + a_{m,n} q_n \lambda_m + 1 s_m \lambda_m = c_m \lambda_m$$

We may obtain this equation from the constraints in the primal problem.

$$\sum_i \sum_j a_{i,j} q_j \lambda_i + \sum_i s_i \lambda_i = \sum_i c_i \lambda_i$$

Below, we find the dual problem.

There we minimize $\overline{\Pi}$, the "total capacity value". Right now, it may seem strange that we are interested in minimizing this value. Soon, however, we will see that the minimum of $\overline{\Pi}$ is the maximum of Π.
Π is the sum of "capacities" times "marginal capacity values".
Each product, that is possible to produce, gives a lower bound on the values of the different resources. (Since the resources may be used to produce a product, the total value of the resources needed to produce such a product can not be lower than the net price of such a product.)

$$\min \overline{\Pi} = c_1\lambda_1 + c_2\lambda_2 + ... + c_m\lambda_m$$

s.t.

$$a_{1,1}\lambda_1 + a_{2,1}\lambda_2 + ... + a_{m,1}\lambda_m >= p_1$$
$$a_{1,2}\lambda_1 + a_{2,2}\lambda_2 + ... + a_{m,2}\lambda_m >= p_2$$
.....
$$a_{1,n}\lambda_1 + a_{2,n}\lambda_2 + ... + a_{m,n}\lambda_m >= p_n$$

Below, we replace the inequalities by equalities, introducing "dual slack variables" $\overline{s_j}$. Of course, the dual slack variables may not be negative.

$$a_{1,1}\lambda_1 + a_{2,1}\lambda_2 + ... + a_{m,1}\lambda_m - 1\overline{s_1} = p_1$$
$$a_{1,2}\lambda_1 + a_{2,2}\lambda_2 + ... + a_{m,2}\lambda_m - 1\overline{s_2} = p_2$$
.....
$$a_{1,n}\lambda_1 + a_{2,n}\lambda_2 + ... + a_{m,n}\lambda_m - 1\overline{s_n} = p_n$$

We may multiply each coefficient in equation j by q_j.

$$a_{1,1}q_1\lambda_1 + a_{2,1}q_1\lambda_2 + ... + a_{m,1}q_1\lambda_m - 1\overline{s_1}q_1 = p_1q_1$$
$$a_{1,2}\lambda_1q_2 + a_{2,2}\lambda_2q_2 + ... + a_{m,2}\lambda_mq_2 - 1\overline{s_2}q_2 = p_2q_2$$
.....
$$a_{1,n}\lambda_1q_n + a_{2,n}\lambda_2q_n + ... + a_{m,n}\lambda_mq_n - 1\overline{s_n}q_n = p_nq_n$$

We may obtain this equation from the constraints in the dual problem.

$$\sum_i \sum_j a_{i,j}\lambda_i q_j - \sum_j \overline{s}_j q_j = \sum_j p_j q_j$$

We note the following:

$$\Pi = \sum_j p_j q_j$$

$$\overline{\Pi} = \sum_i c_i \lambda_i$$

$$\Pi = \sum_i \sum_j a_{i,j}\lambda_i q_j - \sum_j \overline{s}_j q_j$$
$$\overline{\Pi} = \sum_i \sum_j a_{i,j} q_j \lambda_i + \sum_i s_i \lambda_i$$

Let us calculate the difference between the total capacity value, $\overline{\Pi}$, and the total variable profit, Π:

$$\overline{\Pi} - \Pi = \sum_i s_i \lambda_i + \sum_j \overline{s}_j q_j$$

We find that the total capacity value, $\overline{\Pi}$, can never be lower than the total variable profit, Π.

$$\left(s_i \geq 0; \lambda_i \geq 0; \overline{s}_j \geq 0; q_j \geq 0\right) \quad \forall i, j \quad \Rightarrow \quad \left(\overline{\Pi} - \Pi\right) \geq 0 \quad \Rightarrow \quad \left(\overline{\Pi} \geq \Pi\right)$$

Since we are interested in the highest possible value of the total variable profit, Π, it is important to be aware that:

$$\min \overline{\Pi} = \max \Pi$$

Hence, if we are interested in the maximum value of the total variable profit, Π, we may as well search for the minimum value of the total capacity value, $\overline{\Pi}$.

Furthermore, if we, somehow, have found feasible solutions such that $\Pi = \overline{\Pi}$, we already know that we have found the solutions $\min \overline{\Pi} = \max \Pi$.

$$\left(\overline{\Pi} = \Pi, s_i \geq 0; \lambda_i \geq 0; \overline{s}_j \geq 0; q_j \geq 0\right) \quad \Leftrightarrow \quad \left[\left(s_i \lambda_i = 0\right) \forall i \quad \wedge \quad \left(\overline{s}_j q_j = 0\right) \forall j\right]$$

We also know that if we have found a feasible solution, where $\min \overline{\Pi} = \max \Pi$, then, for each constraint, the “slack variable” times the “shadow price” is zero.

Furthermore, if we have found a feasible solution where $\min \bar{\Pi} = \max \Pi$, then, for each product, the "dual slack variable" times the "production volume" is zero.

(These findings are sometimes called the complementary slackness conditions of Kuhn-Tucker programming.)

These findings may also be used in the reversed order:

Assume that we have a feasible solution.
If we, for each resource constraint, find that the "slack variable" times the "shadow price" is zero and that we, for each product, find that the "dual slack variable" times the "production volume" is zero:
Then we have found the optimal solutions to both the primal and the dual problems:
$\min \bar{\Pi} = \max \Pi$

Reference:
The central ideas have been very well expressed in this book:

Baumol, W.J., Economic Theory and Operations Analysis, Prentice/Hall International editions, 4 ed., 1977.

Now, if we have used some numerical software to obtain these solutions:

- The primal objective function value
- The dual objective function value
- The production volumes
- The dual slack variables
- The shadow prices
- The slack variables

Then, we may check these solutions, using the general results expressed above.

Ex1:

```
model:

max = prof;

prof = p1*q1 + p2*q2;

[machine_1] a11*q1 + a12*q2 <= cap1;

[machine_2] a21*q1 + a22*q2 <= cap2;

p1 = 100;
p2 = 100;

cap1 = 100;
cap2 = 100;

a11 = 1;
a12 = 2;
a21 = 2;
a22 = 1;

end
```

```
Global optimal solution found.
  Objective value:                              6666.667
  Infeasibilities:                              0.000000
  Total solver iterations:                             2


                        Variable           Value        Reduced Cost
                            PROF        6666.667            0.000000
                              P1        100.0000            0.000000
                              Q1        33.33333            0.000000
                              P2        100.0000            0.000000
                              Q2        33.33333            0.000000
                             A11        1.000000            0.000000
                             A12        2.000000            0.000000
                            CAP1        100.0000            0.000000
                             A21        2.000000            0.000000
                             A22        1.000000            0.000000
                            CAP2        100.0000            0.000000

                             Row    Slack or Surplus      Dual Price
                        MACHINE_1        0.000000            33.33333
                        MACHINE_2        0.000000            33.33333
```

Ex2:

```
model:

sets:
prod/1..2/:p,q;
res/1..2/:cap;
resprod(res,prod):a;
endsets

max = prof;
prof = @sum(prod(j): p(j)*q(j));

[machine_1] @sum(prod(j): a(1,j)*q(j)) <= cap(1);
[machine_2] @sum(prod(j): a(2,j)*q(j)) <= cap(2);

data:
p =   100 100;
cap = 100 100;
a =     1   2
        2   1;
enddata

end
```

```
  Global optimal solution found.
  Objective value:                              6666.667
  Infeasibilities:                              0.000000
  Total solver iterations:                             2


                       Variable           Value        Reduced Cost
                           PROF        6666.667            0.000000
                         P( 1)         100.0000            0.000000
                         P( 2)         100.0000            0.000000
                         Q( 1)         33.33333            0.000000
                         Q( 2)         33.33333            0.000000
                        CAP( 1)        100.0000            0.000000
                        CAP( 2)        100.0000            0.000000
                      A( 1, 1)         1.000000            0.000000
                      A( 1, 2)         2.000000            0.000000
                      A( 2, 1)         2.000000            0.000000
                      A( 2, 2)         1.000000            0.000000

                            Row    Slack or Surplus      Dual Price
                      MACHINE_1         0.000000            33.33333
                      MACHINE_2         0.000000            33.33333
```

Ex3:

```
model:

sets:
prod/1..2/:p,q;
res/1..2/:cap;
resprod(res,prod):a;
endsets

max = prof;
prof = @sum(prod(j): p(j)*q(j));

@for(res(i): [machine]@sum(prod(j): a(i,j)*q(j)) <= cap(i));

data:
p =   100 100;
cap = 100 100;
a =     1   2
        2   1;
enddata

end
```

```
  Global optimal solution found.
  Objective value:                              6666.667
  Infeasibilities:                              0.000000
  Total solver iterations:                             2


                       Variable           Value        Reduced Cost
                           PROF        6666.667            0.000000
                           P( 1)       100.0000            0.000000
                           P( 2)       100.0000            0.000000
                           Q( 1)       33.33333            0.000000
                           Q( 2)       33.33333            0.000000
                         CAP( 1)       100.0000            0.000000
                         CAP( 2)       100.0000            0.000000
                        A( 1, 1)       1.000000            0.000000
                        A( 1, 2)       2.000000            0.000000
                        A( 2, 1)       2.000000            0.000000
                        A( 2, 2)       1.000000            0.000000

                             Row    Slack or Surplus      Dual Price
                      MACHINE( 1)       0.000000           33.33333
                      MACHINE( 2)       0.000000           33.33333
```

Ex4:

```
model:

sets:
prod/1..5/:p,q;
res/1..5/:cap;
resprod(res,prod):a;
endsets

max = prof;
prof = @sum(prod(j): p(j)*q(j));

@for(res(i): [machine]@sum(prod(j): a(i,j)*q(j)) <= cap(i));

data:
p =   100 100 100 100 100;
cap = 100 100 100 100 100;
a =     1   2   3   4   5
        5   1   2   3   4
        4   5   1   2   3
        3   4   5   1   2
        2   3   4   5   1;
enddata

end
```

```
Global optimal solution found.
  Objective value:                              3333.333
  Infeasibilities:                              0.000000
  Total solver iterations:                             5


                      Variable           Value        Reduced Cost
                          PROF        3333.333            0.000000
                          P( 1)       100.0000            0.000000
                          P( 2)       100.0000            0.000000
                          P( 3)       100.0000            0.000000
                          P( 4)       100.0000            0.000000
                          P( 5)       100.0000            0.000000
                          Q( 1)       6.666667            0.000000
                          Q( 2)       6.666667            0.000000
                          Q( 3)       6.666667            0.000000
                          Q( 4)       6.666667            0.000000
                          Q( 5)       6.666667            0.000000
                        CAP( 1)       100.0000            0.000000
                        CAP( 2)       100.0000            0.000000
                        CAP( 3)       100.0000            0.000000
                        CAP( 4)       100.0000            0.000000
                        CAP( 5)       100.0000            0.000000
                       A( 1, 1)       1.000000            0.000000
                       A( 1, 2)       2.000000            0.000000
                       A( 1, 3)       3.000000            0.000000
                       A( 1, 4)       4.000000            0.000000
                       A( 1, 5)       5.000000            0.000000
                       A( 2, 1)       5.000000            0.000000
                       A( 2, 2)       1.000000            0.000000
                       A( 2, 3)       2.000000            0.000000
```

A(2, 4)	3.000000	0.000000
A(2, 5)	4.000000	0.000000
A(3, 1)	4.000000	0.000000
A(3, 2)	5.000000	0.000000
A(3, 3)	1.000000	0.000000
A(3, 4)	2.000000	0.000000
A(3, 5)	3.000000	0.000000
A(4, 1)	3.000000	0.000000
A(4, 2)	4.000000	0.000000
A(4, 3)	5.000000	0.000000
A(4, 4)	1.000000	0.000000
A(4, 5)	2.000000	0.000000
A(5, 1)	2.000000	0.000000
A(5, 2)	3.000000	0.000000
A(5, 3)	4.000000	0.000000
A(5, 4)	5.000000	0.000000
A(5, 5)	1.000000	0.000000

Row	Slack or Surplus	Dual Price
MACHINE(1)	0.000000	6.666667
MACHINE(2)	0.000000	6.666667
MACHINE(3)	0.000000	6.666667
MACHINE(4)	0.000000	6.666667
MACHINE(5)	0.000000	6.666667

Ex5:

```
model:

sets:
prod/1..5/:p,q;
res/1..5/:cap;
resprod(res,prod):a;
endsets

max = prof;
prof = @sum(prod(j): p(j)*q(j));

@for(res(i): [machine]@sum(prod(j): a(i,j)*q(j)) <= cap(i));

data:
p = @OLE('ex5.XLS');
cap = @OLE('ex5.XLS');
a =   @OLE('ex5.XLS');
@OLE('ex5.XLS') = PROF, Q, MACHINE;

enddata

end
```

ex5

	A	B	C	D	E	F	G	H
1	**ex5**							
2				1	2	3	4	5
3		**p**	**cap**	**a**				
4	1	100	100	1	2	3	4	5
5	2	100	100	5	1	2	3	4
6	3	100	100	4	5	1	2	3
7	4	100	100	3	4	5	1	2
8	5	100	100	2	3	4	5	1
9								
10								
11	**PROF**		**Q**		**MACHINE**			
12	3333,333		6,666667		0			
13			6,666667		0			
14			6,666667		0			
15			6,666667		0			
16			6,666667		0			
17								

```
  Global optimal solution found.
  Objective value:                              3333.333
  Infeasibilities:                              0.000000
  Total solver iterations:                             5
```

```
Export Summary Report
---------------------
Transfer Method:       OLE BASED
Workbook:              ex5.XLS
Ranges Specified:               3
    PROF
    Q
    MACHINE
Ranges Found:                   3
Range Size Mismatches:          0
Values Transferred:            11
```

Variable	Value	Reduced Cost
PROF	3333.333	0.000000
P(1)	100.0000	0.000000
P(2)	100.0000	0.000000
P(3)	100.0000	0.000000
P(4)	100.0000	0.000000
P(5)	100.0000	0.000000
Q(1)	6.666667	0.000000
Q(2)	6.666667	0.000000
Q(3)	6.666667	0.000000
Q(4)	6.666667	0.000000
Q(5)	6.666667	0.000000
CAP(1)	100.0000	0.000000
CAP(2)	100.0000	0.000000
CAP(3)	100.0000	0.000000
CAP(4)	100.0000	0.000000
CAP(5)	100.0000	0.000000
A(1, 1)	1.000000	0.000000
A(1, 2)	2.000000	0.000000
A(1, 3)	3.000000	0.000000
A(1, 4)	4.000000	0.000000
A(1, 5)	5.000000	0.000000
A(2, 1)	5.000000	0.000000
A(2, 2)	1.000000	0.000000
A(2, 3)	2.000000	0.000000
A(2, 4)	3.000000	0.000000
A(2, 5)	4.000000	0.000000
A(3, 1)	4.000000	0.000000
A(3, 2)	5.000000	0.000000
A(3, 3)	1.000000	0.000000
A(3, 4)	2.000000	0.000000
A(3, 5)	3.000000	0.000000
A(4, 1)	3.000000	0.000000
A(4, 2)	4.000000	0.000000
A(4, 3)	5.000000	0.000000
A(4, 4)	1.000000	0.000000
A(4, 5)	2.000000	0.000000
A(5, 1)	2.000000	0.000000
A(5, 2)	3.000000	0.000000
A(5, 3)	4.000000	0.000000
A(5, 4)	5.000000	0.000000
A(5, 5)	1.000000	0.000000

Row	Slack or Surplus	Dual Price
1	3333.333	1.000000
2	0.000000	1.000000
MACHINE(1)	0.000000	6.666667
MACHINE(2)	0.000000	6.666667
MACHINE(3)	0.000000	6.666667
MACHINE(4)	0.000000	6.666667
MACHINE(5)	0.000000	6.666667