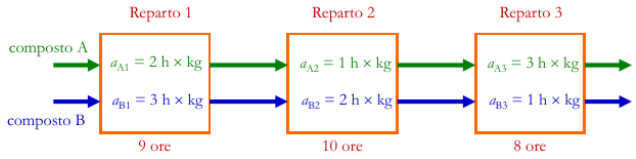
mix produttivo

July 10, 2021

1 mix produttivo

Un'azienda utilizza tre reparti (1, 2 e 3) per produrre due composti (A e B).

Ogni kg di A e B richiede un dato numero di ore di lavorazione in ciascun reparto e ogni reparto ha una data capacità produttiva, ossia è disponibile per un certo numero di ore al giorno (vedi schema).



	Rep 1	Rep 2	Rep 3	Profitto (€/kg)
Composto A	2	1	3	5
Composto B	3	2	1	4
Capacità (ore)	9	10	8	

Considerato che ogni composto ha un dato prezzo di vendita al kg (ultima colonna della tabella), qual è la produzione di A e B che, nel rispetto delle capacità produttive dei reparti, massimizza il profitto dell'azienda?

1.1 formulazione generale del problema di mix produttivo

Un'azienda produce n articoli $P_1,\ldots,P_n$ utilizzando un impianto con m reparti $R_1,\ldots,R_m$.

Ogni unità dell'articolo P_i assicura all'azienda un profitto netto di p_i € e richiede a_{ij} ore di lavorazione nel reparto R_j.

Qual è il mix produttivo giornaliero più redditizio per l'azienda, cioè quali sono le quantità di ogni articolo che l'azienda deve produrre per massimizzare il profitto rispettando le capacità produttive dei reparti?

1.2 modello parametrico

Parametri

- n numero di articoli
- m numero di reparti
- a_{ij} ore di lavorazione in R_j per produrre un pezzo di P_i
- p_i profitto unitario dell'articolo P_i
- b_j capacità produttiva del reparto R_j

variabili decisionali

- $x_i \in \mathbb{R}$: qtà di P_i che si decide di produrre ($i = 1,\ldots, n$)

funzione obiettivo:
Massimizzazione del profitto totale

$$z = \max \sum_{i=1}^{n} p_i x_i$$

vincoli:

- Ogni reparto non può essere utilizzato per un numero di ore superiore alla propria capacità produttiva

$$\sum_{i=1}^{n} a_{ij} x_i \le b_j \qquad \forall j = 1,\ldots,m$$

- I livelli di produzione sono quantità non negative

$$x_i \ge 0 \qquad \forall i = 1,\ldots,n$$

[3]:
```
from pyomo.environ import *
from pyomo.opt import SolverStatus, TerminationCondition

model = ConcreteModel()
model.name = "mix produttivo"

i = [ 'composto A', 'composto B' ]
p = [            5,             4]

j = [ 'rep 1', 'rep 2', 'rep 3']
b = [       9,      10,       8]

a = [[2, 3],
     [1, 2],
     [3, 1]]

model.i = Set(initialize=i)
model.j = Set(initialize=j)

a_dict = {}
for j, mj in enumerate(model.j):
    for i, mi in enumerate(model.i):
        a_dict[mj, mi] = a[j][i]

p = {mi: p[i] for i, mi in enumerate(model.i)}
b = {mj: b[j] for j, mj in enumerate(model.j)}

model.p = Param(model.i, initialize=p)
model.b = Param(model.j, initialize=b)

model.a = Param(model.j, model.i, initialize=a_dict)

model.x = Var(model.i, domain=NonNegativeReals, initialize=0)

obj_expr = sum(model.p[i]*model.x[i] for i in model.i)

model.cost = Objective(expr = obj_expr, sense=maximize)

model.constraints = ConstraintList()

for j in model.j:
    model.constraints.add(expr = sum(model.a[j, i]*model.x[i] for i in model.i) <=␣
 ↪model.b[j])

results = SolverFactory('glpk').solve(model)

model.display()
```

```
Model mix produttivo

  Variables:
    x : Size=2, Index=i
        Key        : Lower : Value            : Upper : Fixed : Stale : Domain
        composto A :     0 : 2.14285714285714 :  None : False : False : NonNegativeReals
        composto B :     0 : 1.57142857142857 :  None : False : False : NonNegativeReals

  Objectives:
    cost : Size=1, Index=None, Active=True
        Key  : Active : Value
        None :   True : 16.999999999999982

  Constraints:
    constraints : Size=3
        Key : Lower : Body             : Upper
          1 :  None : 8.99999999999999 :   9.0
          2 :  None : 5.28571428571428 :  10.0
          3 :  None : 7.99999999999999 :   8.0
```