

pianificazione di investimenti

July 8, 2021

1 pianificazione di investimenti

Si dispone di un budget di $b\text{€}$ per realizzare n progetti.

Ogni progetto ha un costo c_i e un guadagno atteso di $p_i\text{€}$. Quali progetti occorre selezionare per massimizzare il guadagno atteso rispettando il vincolo di budget?

	guadagno p (K€)	costo c (K€)
progetto 1	60	3
progetto 2	65	5
progetto 3	45	1
progetto 4	55	2

budget: $b = 7,000 \text{ €}$

1.1 modello di programmazione matematica

$$Z = \max(60P_1 + 65P_2 + 45P_3 + 55P_4)$$

$$C1 : 3P_1 + 5P_2 + P_3 + 2P_4 \leq 7$$

$$C2 : P_1 \geq 0$$

$$C3 : P_2 \geq 0$$

$$C4 : P_3 \geq 0$$

$$C5 : P_4 \geq 0$$

```
[4]: import glpk

lp = glpk.LPX()
lp.name = 'pianificazione investimenti'

lp.obj.maximize = True
lp.rows.add(1)

lp.rows[0].name = 'profitto'

lp.rows[0].bounds = (None, 7.0)

lp.cols.add(4)

lp.cols[0].name = 'P1'
lp.cols[1].name = 'P2'
lp.cols[2].name = 'P3'
lp.cols[3].name = 'P4'

lp.cols[0].bounds = (0.0, None)
lp.cols[1].bounds = (0.0, None)
lp.cols[2].bounds = (0.0, None)
lp.cols[3].bounds = (0.0, None)

lp.cols[0].kind = int
lp.cols[1].kind = int
lp.cols[2].kind = int
lp.cols[3].kind = int

lp.obj[:] = [ 60, 65, 45, 55 ]

lp.matrix = [ 3, 5, 1, 2 ]

lp.simplex()
lp.integer() # -> per ottenere soluzioni intere

print(f'profitto = {lp.obj.value}') # Retrieve and print obj func value
print('; '.join('%s = %g' % (c.name, c.primal) for c in lp.cols))
```

```
profitto = 315.0
P1 = 0; P2 = 0; P3 = 7; P4 = 0
```

```
[ ]:
```