



# Ottimizzazione di una Dieta Settimanale Bilanciata

Tramite Mixed Integer Linear Programming



**Studenti:** *Stefano Di Lena*  
*Paolo Santoro*



# Descrizione Qualitativa del Problema

- **Budget:** 90\$.
- **Equilibrio nutrizionale:** 55% di carboidrati; 25% di grassi; 20% di proteine
- **Restrizioni alimentari (settimanali):** carne rossa massimo 1 volta; salmone massimo 2 volte, patate massimo 3 volte; legumi almeno 4 volte; merluzzo almeno 3 volte.
- **Fabbisogno nutrizionale giornaliero:** 2300-3000 kcal/gg.
- **Pasti equilibrati.**



## Valori nutrizionali e prezzo per 100g di alimento

Alimento	Costo [\$]	Kcal	Carboidrati [g]	Grassi [g]	Proteine [g]
<i>Carne Rossa</i>	4,00	127	0	4,6	22
<i>Pollo</i>	1,10	165	0	3,5	30
<i>Salmone</i>	4,50	185	1	12	19
<i>Merluzzo</i>	1,90	82	0	0,5	17,8
<i>Uova</i>	0,10	128	0,7	8,7	12,4
<i>Legumi</i>	0,70	92	16,3	0,4	6,9
<i>Latticini</i>	1,50	275	2	23	20
<i>Patate</i>	0,13	75	16	0,1	2
<i>Pasta</i>	0,25	350	77	1,5	14
<i>Pane</i>	0,21	166	50,6	3,3	7,6



# Funzione Obiettivo

L'obiettivo è minimizzare il costo complessivo della dieta per l'intera settimana.

```
minimize TotalCost : sum {a in ALIMENTI, d in DAY} Costo[a]*Q[a,d];
```

# Variabili decisionali

**$Q[a][d]$** : rappresenta la quantità (in unità di 100g) di un alimento “ $a$ ” che viene consumata nel giorno “ $d$ ” della settimana.

**$y[a][d]$** : È una variabile binaria. Indica se l'alimento “ $a$ ” viene consumato (1) o meno (0) nel giorno “ $d$ ”.

$a$  = {carne rossa, pollo, salmone, merluzzo, uova, legumi, latticini, patate, pasta, pane}  
 $d$  = {lun, mar, mer, gio, ven, sab, dom}





# Vincolo equilibrio nutrizionale

Si è usata la “regola del 4-4-9”, conosciuta come *Atwater general factor system*, adottata nella nutrizione. Abbiamo considerato una tolleranza del 2% sul target calorico..

```
param CarbMin_gg := 304.75; #0,53*2300/4
param CarbMax_gg := 427.5; #0,57*3000/4
param GrassiMin_gg := 58.78; #0,23*2300/9
param GrassiMax_gg := 90; #0,27*3000/9
param ProtMin_gg := 103.5; #0,18*2300/4
param ProtMax_gg := 165; #0,22*3000/4
```

```
s.t. DailyCarbMin {d in DAY}: sum {a in ALIMENTI} Carbo[a]*Q[a,d] >= CarbMin_gg;
s.t. DailyCarbMax {d in DAY}: sum {a in ALIMENTI} Carbo[a]*Q[a,d] <= CarbMax_gg;
s.t. DailyFatMin {d in DAY}: sum {a in ALIMENTI} Grassi[a]*Q[a,d] >= GrassiMin_gg;
s.t. DailyFatMax {d in DAY}: sum {a in ALIMENTI} Grassi[a]*Q[a,d] <= GrassiMax_gg;
s.t. DailyProtMin {d in DAY}: sum {a in ALIMENTI} Prot[a]*Q[a,d] >= ProtMin_gg;
s.t. DailyProtMax {d in DAY}: sum {a in ALIMENTI} Prot[a]*Q[a,d] <= ProtMax_gg;
```



# Vincolo Bilancio Settimanale

```
s.t. BilancioSettimanale:  $\sum \{a \text{ in } \text{ALIMENTI}, d \text{ in } \text{DAY}\} \text{Costo}[a] * Q[a,d] \leq 90;$ 
```

# Vincolo Fabbisogno Calorico Giornaliero

```
s.t. DailyKcalMin {d in DAY} :  $\sum \{a \text{ in } \text{ALIMENTI}\} \text{Kcal}[a] * Q[a,d] \geq \text{KcalMin\_gg};$   
s.t. DailyKcalMax {d in DAY} :  $\sum \{a \text{ in } \text{ALIMENTI}\} \text{Kcal}[a] * Q[a,d] \leq \text{KcalMax\_gg};$ 
```

# Vincolo di Collegamento

Legano la variabile continua a quella binaria:

- se un alimento viene consumato, la sua variabile binaria corrispondente deve essere uguale ad 1;
- se la variabile binaria è 0 allora la quantità deve essere 0.

```
s.t. LinkQy {a in ALIMENTI, d in DAY}:  $Q[a,d] \leq M * y[a,d];$ 
```



## Vincolo Restrizioni Alimentari specifiche

```
s.t. MeatLimit: sum {d in DAY} y["CarneRossa",d] = 1;  
s.t. SalmonLimit: sum {d in DAY} y["Salmone",d] <= 2;  
s.t. PotatoLimit: sum {d in DAY} y["Patate",d] <= 3;  
s.t. LegumiMin: sum {d in DAY} y["Legumi",d] >= 4;  
s.t. MerluzzoMin: sum {d in DAY} y["Merluzzo",d] >= 3;  
s.t. EggSalmon {d in DAY}: y["Uova",d] + y["Salmone",d] <= 1;
```





# Vincoli di dieta varia

È stato necessario introdurre nuovi vincoli, poiché l'alimentazione risultava monotona e squilibrata, con una preferenza esclusiva per uova e pasta.

```
s.t. PolloMin: sum {d in DAY} y["Pollo",d] >= 1;  
s.t. SalmoneMin: sum {d in DAY} y["Salmone",d] >= 1;  
s.t. MozzarellaMin: sum {d in DAY} y["Latticini",d] >= 1;  
s.t. PastaMin: sum {d in DAY} y["Pasta",d] >= 1;  
s.t. PatateMin: sum {d in DAY} y["Patate",d] >= 1;  
s.t. PaneMin: sum {d in DAY} y["Pane",d] >= 1;  
s.t. UovaMin: sum {d in DAY} y["Uova",d] >= 1;
```

Inoltre, è stato aggiunto un nuovo parametro (epsilon) per garantire che, se un alimento è stato scelto in un certo giorno, allora venga effettivamente consumato in una quantità minima (nel nostro caso almeno 100g).

```
s.t. LinkYQ_lower {a in ALIMENTI, d in DAY}: Q[a,d] >= epsilon * y[a,d];
```



# Altri vincoli

Non è possibile consumare due giorni consecutivi uova.

```
s.t. NoUovaLunMar: y["Uova", "lun"] + y["Uova", "mar"] <= 1;  
s.t. NoUovaMarMer: y["Uova", "mar"] + y["Uova", "mer"] <= 1;  
s.t. NoUovaMerGio: y["Uova", "mer"] + y["Uova", "gio"] <= 1;  
s.t. NoUovaGioVen: y["Uova", "gio"] + y["Uova", "ven"] <= 1;  
s.t. NoUovaVenSab: y["Uova", "ven"] + y["Uova", "sab"] <= 1;  
s.t. NoUovaSabDom: y["Uova", "sab"] + y["Uova", "dom"] <= 1;
```



# Analisi dei Risultati

```
Q [*,*]  
:  
      dom      gio      lun      mar      mer      sab      ven  :=  
CarneRossa  0      2.97391  0      0      0      0      0  
Latticini   1.5    1.5      1.5    1.5    1.5    1.5    1.5  
Legumi      0      1      1      0      0      1      1  
Merluzzo    1      0      0      1      0      1      0  
Pane        1.5    1.5      1.5    1.5    1.5    1.5    1.5  
Pasta       3.5    3.5      3.5    3.5    3.5    3.5    3.5  
Patate      0      0      0      1.66273  1.2563  1      0  
Pollo       1.07818  0      1.01758  0      1.00411  0      1.01758  
Salmone     0      0      0      1.11781  0      1.09    0  
Uova        1.2    0      1.2    0      1.2    0      1.2  
;  
  
TotalCost = 59.9295
```



Risultati Risolutore

✕

Il problema è troppo complesso perché il Risolutore possa gestirlo.

☒ Mantieni soluzione del Risolutore

☐ Ripristina valori originali

☐ Torna alla finestra di dialogo parametri Risolutore


☐ Rapporti struttura

Rapporti

OK

Annulla

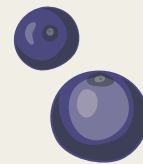
Salva scenario...



**Il problema è troppo complesso perché il Risolutore possa gestirlo.**

Per il Risolutore il numero massimo di celle variabili è 200 e quello dei vincoli è 100, oltre ai limiti per le celle variabili.

# Ottimizzazione Robusta



```
ampl: Q [*,*]
:      dom      gio      lun      mar      mer      sab      :=
CarneRossa  0      0      0      2.97391  0      0
Latticini   1.5    1.5    1.5    1.5      1.5    1.5
Legumi      1      0      1      1        0      0
Merluzzo    0      1      0      0        1      1
Pane        1.5    1.5    1.5    1.5      1.5    1.5
Pasta       3.5    3.5    3.5    3.5      3.5    3.5
Patate      0      1.66273  0      0        0      1.66273
Pollo       1.01758  0      1.01758  0      1.07818  0
Salmone     0      1.11781  0      0        0      1.11781
Uova        1.2    0      1.2      0      1.2      0

:      ven      :=
CarneRossa  0
Latticini   1.5
Legumi      1
Merluzzo    0
Pane        1.5
Pasta       3.5
Patate      0
Pollo       1.01758
Salmone     0
Uova        1.2
;

TotalCost = 72.9341
```



# **Grazie per l'attenzione!**

Fine.