

ROAR III

Ricerca Operativa Applicazioni Reali

Alessandro Gobbi Alice Raffaele Gabriella Colajanni Eugenia Taranto

IIS Antonietti, Iseo (BS)

03 dicembre 2022

Introduzione

Chi siamo e i nostri contatti



Alessandro Gobbi (UniBS)
alessandro.gobbi@unibs.it



Alice Raffaele (Univr)
alice.raffaele@univr.it



Gabriella Colajanni (Unict)
gabriella.colajanni@unict.it



Eugenia Taranto (Unict)
eugenia.taranto@unict.it

Abbiamo trovato il file *Problema_1.py* che contiene l'implementazione in Python+Pulp di un modello matematico.

Purtroppo il codice non è nemmeno commentato e quindi non abbiamo alcun indizio sulla formulazione matematica del problema.

Potreste aiutarci, partendo dal codice, a scrivere sia la formulazione matematica sia un plausibile testo che si adatti bene al problema?

Correzione compiti – Dal codice al problema...

```
from pulp import *

model = LpProblem("Problema?", LpMaximize)

indici = [1,2,3]
Parametri={1:150, 2: 320, 3:60}
Valori_1 = {1: 9500, 2: 123}
valori_2 = {
    1: {1: 30, 2: 120, 3: 12},
    2: {1: 0.7, 2: 1, 3: 0.5}}
valori_3 = {1: 0, 2: -1, 3:5}

var_x = LpVariable.dicts("x", indici, 0, None, LpInteger)

model += lpSum(var_x[i] * Parametri[i] for i in indici)

for j in valori_2.keys():
    model += lpSum(var_x[i]*valori_2[j][i] for i in indici) <= Valori_1[j]

model += lpSum(valori_3[k]*var_x[k] for k in indici) >= 0

status = model.solve(PULP_CBC_CMD(msg = 0))
if status == 1 :
    print("Soluzione ottima:")
    for v in model.variables():
        print(v.name, " = ", v.varValue)

    print("\nIl valore della funzione obiettivo è {}".format(round(value(model.objective),2)))
else:
    print("Problema non ammissibile")
```

1. Leggere bene lo script *Problema_1.py* e comprendere ogni riga del codice.
2. Scrivere la formulazione del modello matematico descritto dal codice.
3. Ideare un plausibile testo del problema, coerente con il modello.

Correzione – Formulazione del modello

Il problema è costituito da:

- tre variabili intere (x_1, x_2, x_3);
- due vincoli di capacità (nella stessa forma);
- un ulteriore vincolo.

La formulazione del modello è la seguente:

$$\max 150 \cdot x_1 + 320 \cdot x_2 + 60 \cdot x_3$$

funzione obiettivo

$$30 \cdot x_1 + 120 \cdot x_2 + 12 \cdot x_3 \leq 9500$$

$$0.7 \cdot x_1 + x_2 + 0.5 \cdot x_3 \leq 123$$

$$0 \cdot x_1 - x_2 + 5 \cdot x_3 \geq 0 \quad \Longleftrightarrow \quad x_3 \geq \frac{1}{5}x_2$$

$$x_1, x_2, x_3 \in \mathbb{N}$$

variabili intere

In generale, il problema è il seguente:

$$\max \sum_{i \in I} p_i \cdot x_i$$

funzione obiettivo

$$\sum_{i \in I} c_{i,j} \cdot x_i \leq \bar{C}_j$$

$\forall j \in V$

$$\sum_{i \in I} r_i \cdot x_i \geq 0$$

$$x_i \in \mathbb{N}$$

variabili intere

Parte 1: Obiettivo 1 –
Formulazione del modello
matematico del problema
Filtrec-M1

Filtrec-M1: il problema dei milkrun semplificato su un giorno

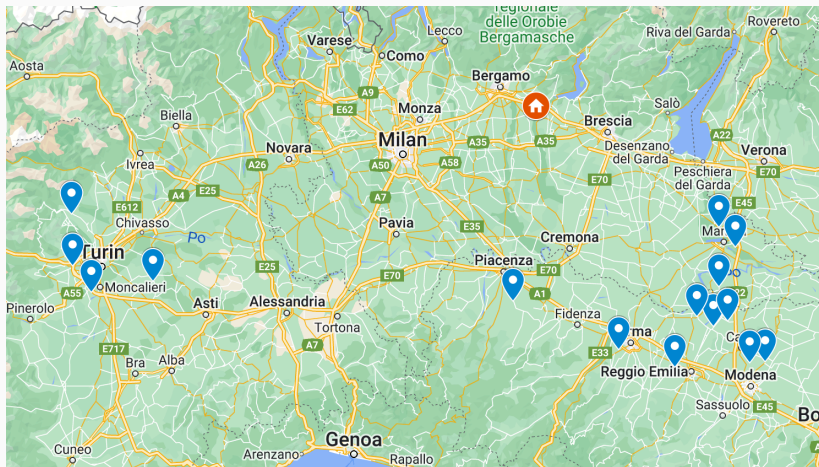
Consideriamo:

- un orizzonte temporale di un giorno;
- un solo deposito (la sede di Telgate), punto di partenza/ritorno dei corrieri;
- un sottoinsieme di terzisti di Filtrec, ognuno con una quantità media di pallet consegnata dal corriere durante una visita (ipotizziamo che non ci siano pallet da ritirare e che i tempi di servizio siano trascurabili);
- un insieme di veicoli omogenei, aventi caratteristiche uguali in termini di capacità, tempo massimo di circolazione, punto di partenza/ritorno e costo fisso giornaliero.

Obiettivo:

- dato il costo fisso giornaliero di ogni veicolo, si vuole cercare di spendere meno possibile, usando meno veicoli possibili per servire tutti i terzisti in quel determinato giorno.

Mappa dei terzisti considerati



Mappa: <https://shorturl.at/adjlW>

Filtrec-M2: il problema dei milkrun semplificato su una settimana lavorativa

Estendiamo il problema *Filtrec-M1* su tutta la settimana lavorativa.

Consideriamo:

- un orizzonte temporale di cinque giorni (dal lunedì al venerdì);
- le altre ipotesi del problema *Filtrec-M1*;
- una *frequenza minima di visita settimanale* per ogni terzista.

Analogie con il problema VRP Challenge

- Uso della teoria dei grafi per modellizzare il problema, di natura temporale.
- Esempio di Vehicle Routing Problem, un problema di instradamento che prevede la progettazione delle rotte che un gruppo di veicoli deve seguire per rispondere a determinate richieste.
- Scelta delle strade più opportune e del numero di veicoli, allo scopo di efficientare il servizio di consegna delle borse/pallet spendendo il meno possibile.
- Anche in questo problema i veicoli non possono sfiorare un tempo massimo di circolazione (che va minimizzato quanto più possibile), hanno un deposito di partenza e rientro, e hanno una capacità limitata.
- Costo fisso di uso del veicolo, che non è obbligatorio usare

Differenze con il problema VRP Challenge

- Nel problema reale, l'orizzonte temporale è di più giorni, si può sia consegnare sia ritirare merce, e la presenza dell'area di carico/scarico merci può influire sui veicoli scelti.
- Estensione geografica più ampia
- I terzisti non sono sempre pronti per il ritiro e la consegna, mentre i clienti a ricevere la spesa sì.
- Si tiene conto dei dati storici per quantificare le consegne.
- I veicoli sono omogenei e non eterogenei per capacità/dimensioni.
- Esiste solo un deposito, che funge sia da punto di arrivo sia da punto di partenza.
- La funzione obiettivo è diversa, perché nei costi è già incluso quello della benzina (da non valutare quindi separatamente).

Insiemi base per gli Obiettivi 1 e 2:

- *depositi*, ognuno dotato di un *nome*;
- *terzisti*, ognuno dotato di un *nome* e di una certa quantità di *consegne*;
- *veicoli*, ognuno dotato di un *nome*, un *deposito*, una *capacità*, una *distanza massima* e un *tempo massimo* di percorrenza, e un *costo giornaliero*.

Insiemi per l'Obiettivo 3:

- *vicini*, descritti come un insieme di *terzisti*;
- *precedenze*, descritte come *coppie ordinate di terzisti*;
- *diversi*, descritti come *coppie* di *terzisti*.

Altri parametri del problema `Filtrec-M1`

- `MAX_TIME_PULP` = 300 secondi, il tempo massimo di esecuzione dello script;
- `M` = `1000`, un valore molto alto utile in alcuni vincoli;
- `distanze`, un dizionario per ottenere la distanza (in Km) tra qualsiasi coppia di nodi i e j ;
- `tempi`, un dizionario per ottenere il tempo (in minuti) tra qualsiasi coppia di nodi i e j .

Variabili decisionali del problema Filtrac-M1

Domande:

- *uso o non uso il veicolo v ?*
- *assegno o non assegno il terzista i al veicolo v ?*
- *il tratto (i, j) è percorso o no dal veicolo v ?*
- *in quale istante giunge il veicolo v nel nodo j partendo dal nodo i ?*

Variabili:

- per ogni veicolo $v \in V$, si introduce una variabile binaria u_v , che vale 1 se v è usato;
- per ogni coppia (i, v) , dove $i \in T$ e $v \in V$, si introduce una variabile binaria y_{iv} , che vale 1 se i è visitato da v ;
- per ogni arco (i, j) con $i, j \in D + T$, e per ogni veicolo $v \in V$, si introduce una variabile binaria x_{ijv} , che vale 1 se v percorre (i, j) ;
- per ogni arco (i, j) con $i, j \in D + T$, e per ogni veicolo $v \in V$, si introduce una variabile continua q_{ijv} , il cui valore corrisponde all'istante in cui v giunge in j da i .

Variabili decisionali del problema *Filter-M1* in Python+PuLP

uso o non uso il veicolo v?

```
u = LpVariable.dicts("u", V, 0, 1, LpBinary)
```

assegno o non assegno il terzista i al veicolo v?

```
y = LpVariable.dicts("y", (T,V), 0, 1, LpBinary)
```

il tratto (i,j) è percorso o no dal veicolo v?

```
x = LpVariable.dicts("x", (D+T, D+T, V), 0, 1, LpBinary)
```

in quale istante giunge il veicolo v nel nodo j partendo da i?

(servono per i vincoli di eliminazione dei cicli)

```
q = LpVariable.dicts("q", (D+T, D+T, V), 0, None, LpContinuous)
```

Lavoro di gruppo #1 – Formulazione del modello matematico di Filtrec-M1 (45 minuti)

1. Scrivere **su carta** la formulazione del modello matematico usando le variabili decisionali definite poco fa.
2. Fare una foto al modello formulato e caricarla nella cartella Drive del vostro gruppo.

Vedasi file *Filtrec-M1.pdf*

Parte 2:

Obiettivo 2 – Implementazione
del modello matematico del
problema **Filtrec-M1** in
Python+PuLP

Lavoro di gruppo #2 – Implementazione in Python+PuLP (75 minuti)

1. Implementare in Python+PuLP la formulazione corretta del modello matematico.
2. Caricare lo script sviluppato nella cartella Drive del vostro gruppo.

Vedasi script *Filtrec-M1.py*

Conclusione



www.menti.com – Codice: 5403 2055