Meal forecasting with capacitated vehicle routing

Esame di Algoritmi di Ottimizzazione

Lorenzo Rossi matr. 183590

2022-2023

Descrizione del problema

Una compagnia di Meal Delivery con diversi centri sparsi in diverse regioni deve preparare il cibo per le settimane a venire. Dati gli ordini precedenti aggregati per settimana è necessario prevedere gli ordini della prossima settimana e creare un piano di consegna da un Depot centrale a tutti i centri di consegna, minimizzando la distanza percorsa.

Dati forniti gentilmente da Vidhya* tramite Keggle, aggregati e anonimizzati, dato che non contiene ingredienti o posizioni è necessario generare i dati verosimilmente.

*Analytics Vidhya. Food demand forecasting, 2020.

Meal forecasting

Due strategie:

Average

Media delle N precedenti settimane, proiettate nelle settimane future.

Il parametro N è stato ottimizzato manualmente a N=3

Prophet

Utilizzo della famosa libreria Prophet per l'estrazione delle stagionalità e dei trend.

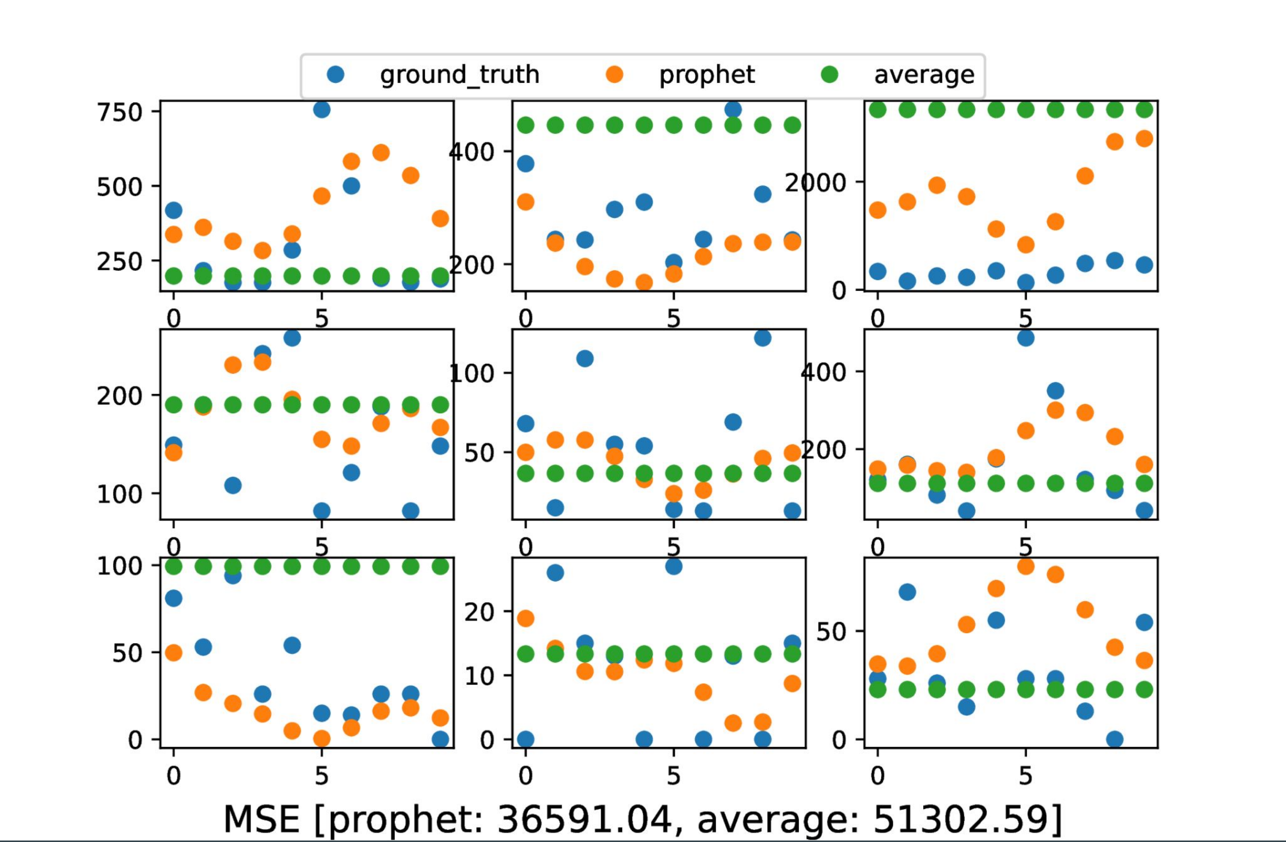
Per semplificare il problema non si sono usati i dati aggiuntivi del dataset (ex. costo, promozioni, ecc. ecc.)

Per comparare i risultati: Mean Squared Error tra i dati predetti e i dati reali

Meal forecasting - Results

Per predizioni più lunghe Prophet ha l'errore minimo dato che riesce ad estrarre i trend dei dati.

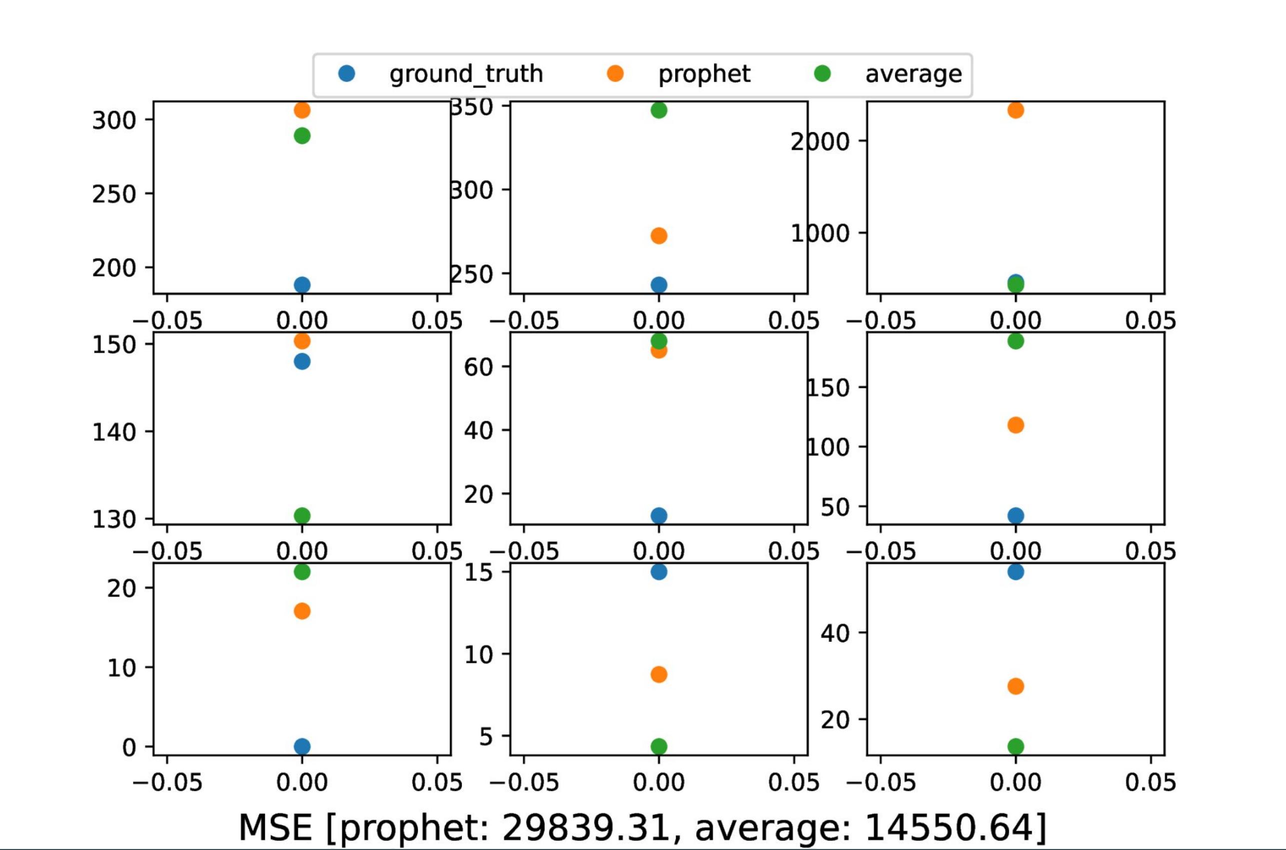
Abbastanza stranamente Average vince sulle predizioni più corte (solo 1 settimana).



Meal forecasting - Results

Per predizioni più lunghe Prophet ha l'errore minimo dato che riesce ad estrarre i trend dei dati.

Abbastanza stranamente Average vince sulle predizioni più corte (solo 1 settimana).



Generazione delle posizioni

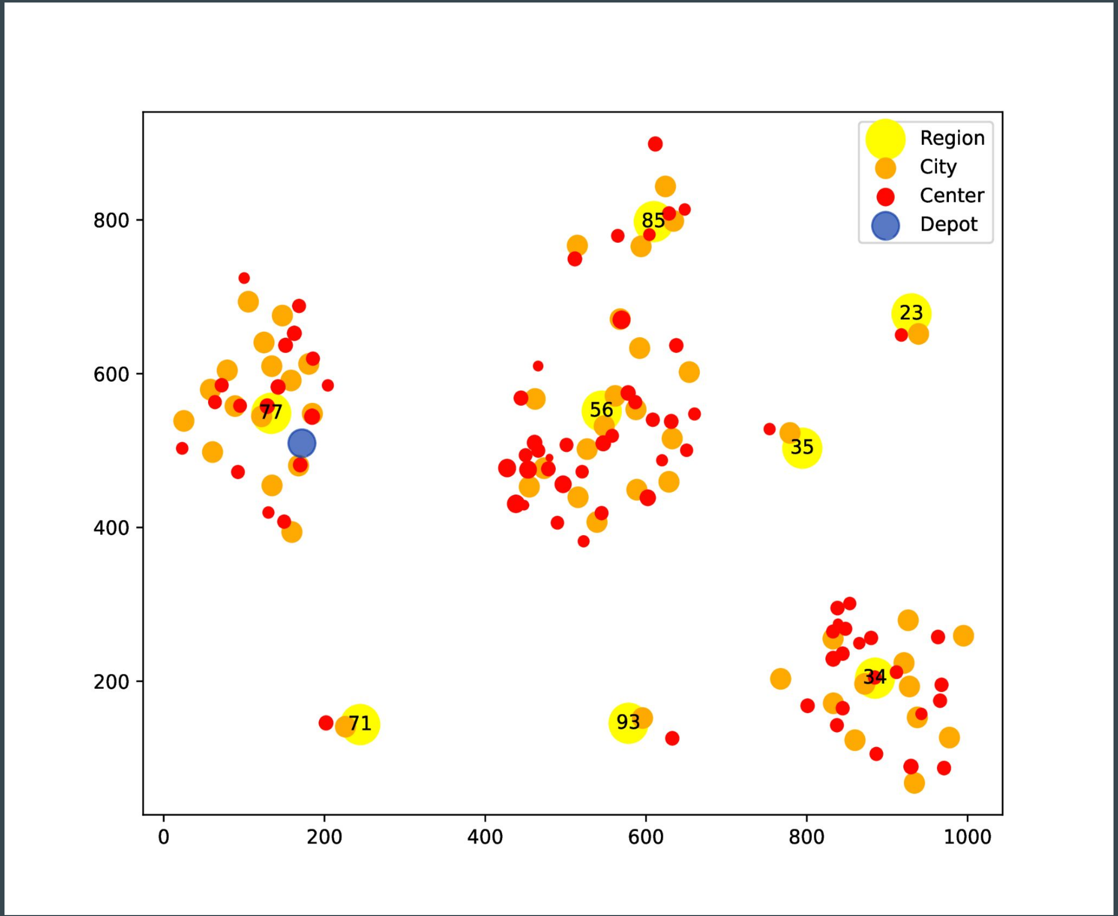
Distribuzioni normali relative

Regioni -> Città -> Centri

Usata anche una distanza minima per evitare overlap

Depot generato come una città, selezionando randomicamente in quale città è situato

Regioni e città per ogni centro erano fornite dai dati.



VRPC polinomiale

Numero polinomiale di variabili e di constraint.

La variabile c tiene traccia della capacità del veicolo e rimuove i cicli.

Versione leggermente modificata del modello visto a lezione per il Traveling Salesman Problem, usando la capacità del veicolo al posto dell'ordine di visita.

$$
\begin{aligned}
\min \quad & \sum_{i,j} d_{i,j} x_{i,j} \\
\text{s.t.} \quad & \sum_{i \neq j} x_{i,j} = 1 \quad \forall j \in V \\
& \sum_{j \neq i} x_{i,j} = 1 \quad \forall i \in V \\
& \sum_{i \neq 0} x_{0,i} - x_{i,0} = 0 \\
& c_0 = 0 \\
& c_j - c_i \geq w_j - M(1 - x_{i,j}) \quad \forall (i,j) \in A | i \neq j \wedge j \neq 0 \\
& c_i \leq Q \quad \forall i \in V \\
& x_{i,j} \in \{0, 1\} \\
& c_i \in \mathbb{N}
\end{aligned}
$$

Flow constraints

Capacity constraint and cycle removal

VRPC subtour elimination

Numero polinomiale di variabili, ma non di constraint.

Si aggiungono constraint aggiuntivi mentre si esplora lo spazio di soluzioni (usando lazy constraints).

Depot: nodo 0 e n+1

C = clienti

V = nodi

K = veicoli

Versione modificata di quella vista a lezione

$$\min \sum_{i,j \in A} d_{i,j} \sum_{k \in K} x_{i,j,k}$$

$$\text{s.t.} \quad x_{i,i,k} = 0 \quad \forall i \in V, k \in K$$

Sink and Source c.

$$x_{i,0,k} = 0 \quad \forall i \in V, k \in K$$

$$x_{n+1,i,k} = 0 \quad \forall i \in V, k \in K$$

$$\sum_{j \in V, k \in K} x_{i,j,k} = 1 \quad \forall i \in C$$

Capacity c.

$$\sum_{i \in C} d_i \sum_{j \in V} x_{i,j,k} <= Q \quad \forall k \in K$$

Flow constraints

$$\sum_{j \in V} x_{0,j,k} = 1 \quad \forall k \in K$$

$$\sum_{i \in V} x_{i,n+1,k} = 1 \quad \forall k \in K$$

$$\sum_{i \in V} x_{i,h,k} = \sum_{k \in V} x_{h,j,k} \quad \forall h \in C, k \in K$$

$$\sum_{(i,j) \in A} x_{i,j,k} \leq |S| - 1 \quad \forall S \subset V, k \in K$$

^ Exponential Constraint

$$x_{i,j,k} \in \{0, 1\}$$

VRPC column generation

Simile a quello descritto da Desrochers, Desrosiers and Solomon*, semplificato per togliere i constraint di tempo..

R = Tutti i possibili itinerari (route).

1. Ottimizzare il problema ridotto
2. Trovare variabili con costi ridotti negativi (sottoproblema ottimizzato con Cython).
3. Aggiungere le variabili e ripetere

Branch and Price algorithm custom.
(leggere il report per una spiegazione migliore)

$$
\begin{aligned}
\min \quad & \sum_{r \in R} c_r x_r \\
\text{s.t.} \quad & \sum_{r \in R} a_{i,r} x_r \geq 1 \quad \forall i \in C && \text{Customer satisfaction} \\
& \sum_{r \in R} x_r = X_d && \text{Vehicle n.} \\
& \sum_{r \in R} c_r x_r = X_c && \text{Distance} \\
& x_r \in \{0, 1\} \\
& X_d, X_c \geq 0, integer.
\end{aligned}
$$

*Martin Desrochers, Jacques Desrosiers, and Marius Solomon. A new optimization algorithm for the vehicle routing problem with time windows. Operations research, 1992

Risultati

Il modello subtour elimination è poco competitivo e non produce soluzioni intermedie.

Il modello polinomiale invece è inaspettatamente veloce e riesce ad arrivare a soluzioni migliori in tempo minore del modello column generation.

Dato che l’istanza del problema è molto grande, con 77 nodi, non è stato possibile arrivare a una soluzione ottimale, ma i risultati sono comunque molto interessanti.

Gurobi ha un risolutore di MIP estremamente ottimizzato anche per casi simili, le soluzioni a cui arriva tramite il problema polinomiale sono visibilmente migliori.

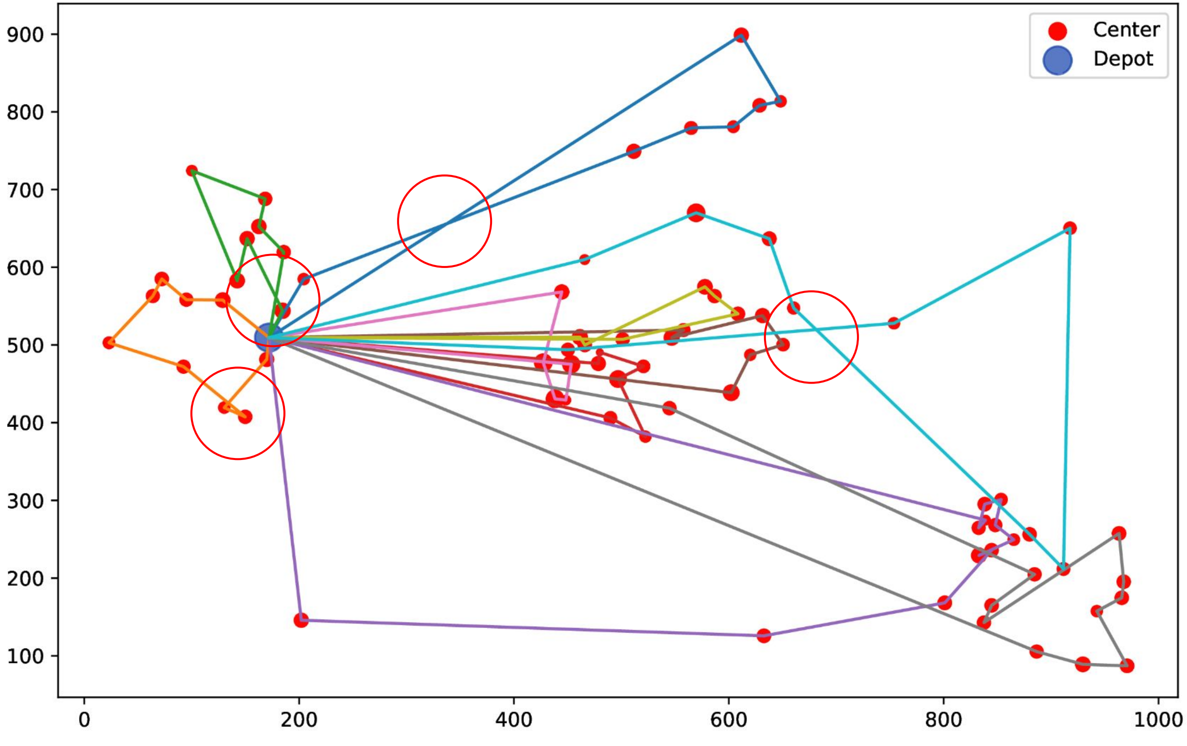
Risultati

Model	Data	Time	Upper Bound	Lower Bound	Initial Lower Bound	Explored nodes	Generated Columns /Rows
Col. Gen.	Forecast	11h	12275.91	10653.04	10624.35	192295	302000
Col. Gen.	Real	22h	11456.74	10573.27	10549.30	363886	766420
Poly.	Forecast	8h†	11223.83	4794.40	4034.81	4977643	-
Poly.	Real	5h†	11354.86	4761.03	4033.18	4670016	-
Sub. Elim.	Forecast	4h†	40474.66	4146.30	3931.12	141979	19452
Sub. Elim.	Real	4h†	40474.66	4204.66	3931.12	138154	19473

†: l’algoritmo è stato terminato perchè ha finito le risorse del computer

Risultati

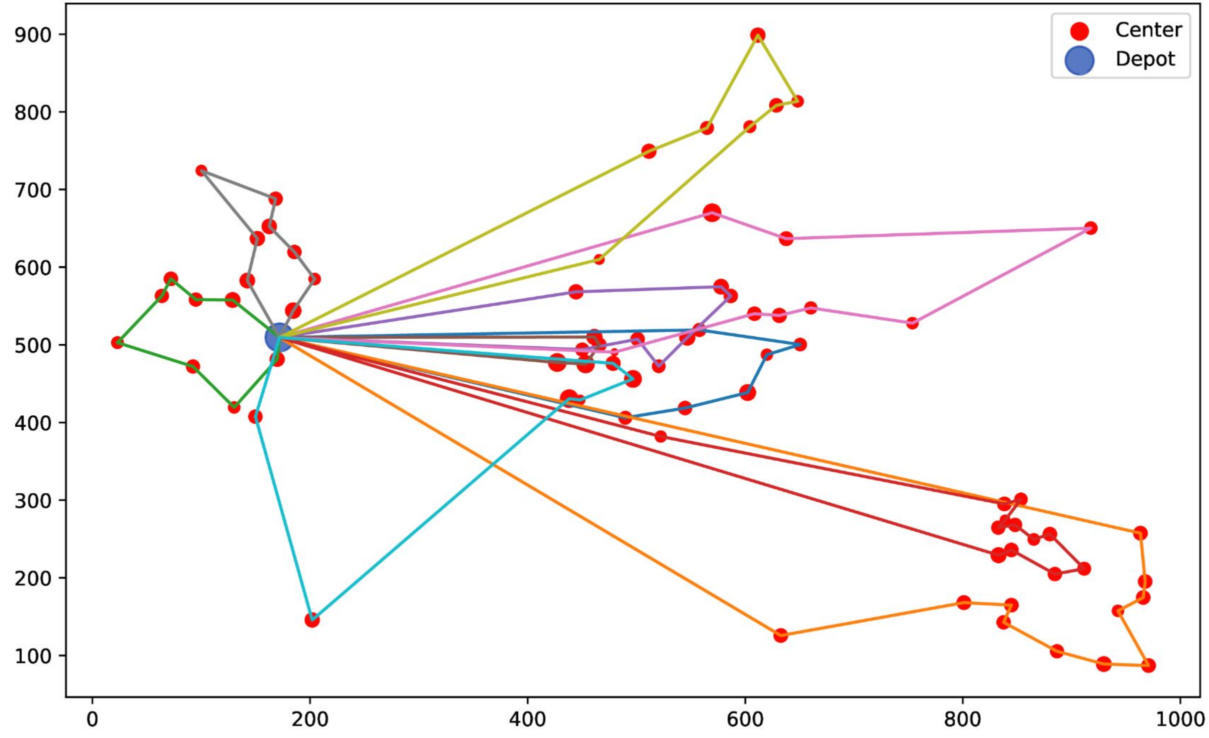
Modello: Column Generation   Runtime: 11h   Data: Average Forecasting



Cost: 12275.46

Risultati

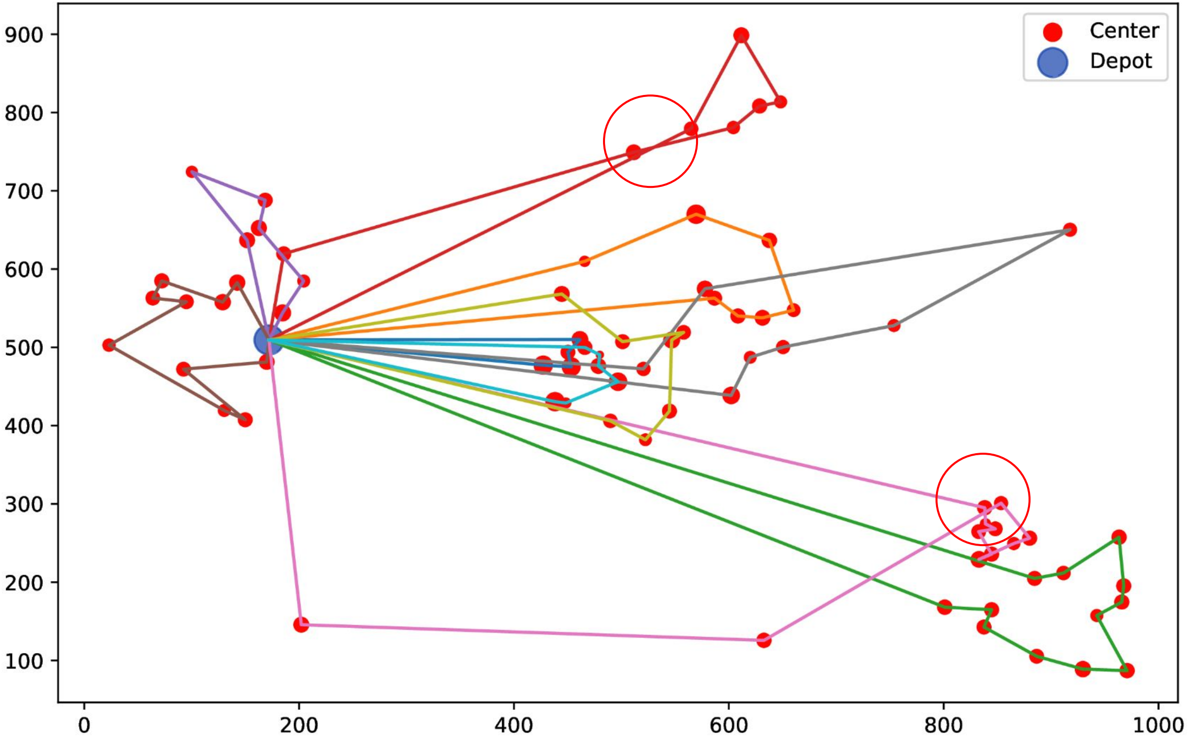
Modello: Polinomiale Runtime: 7h Data: Average Forecasting



Cost: 11223.39

Risultati

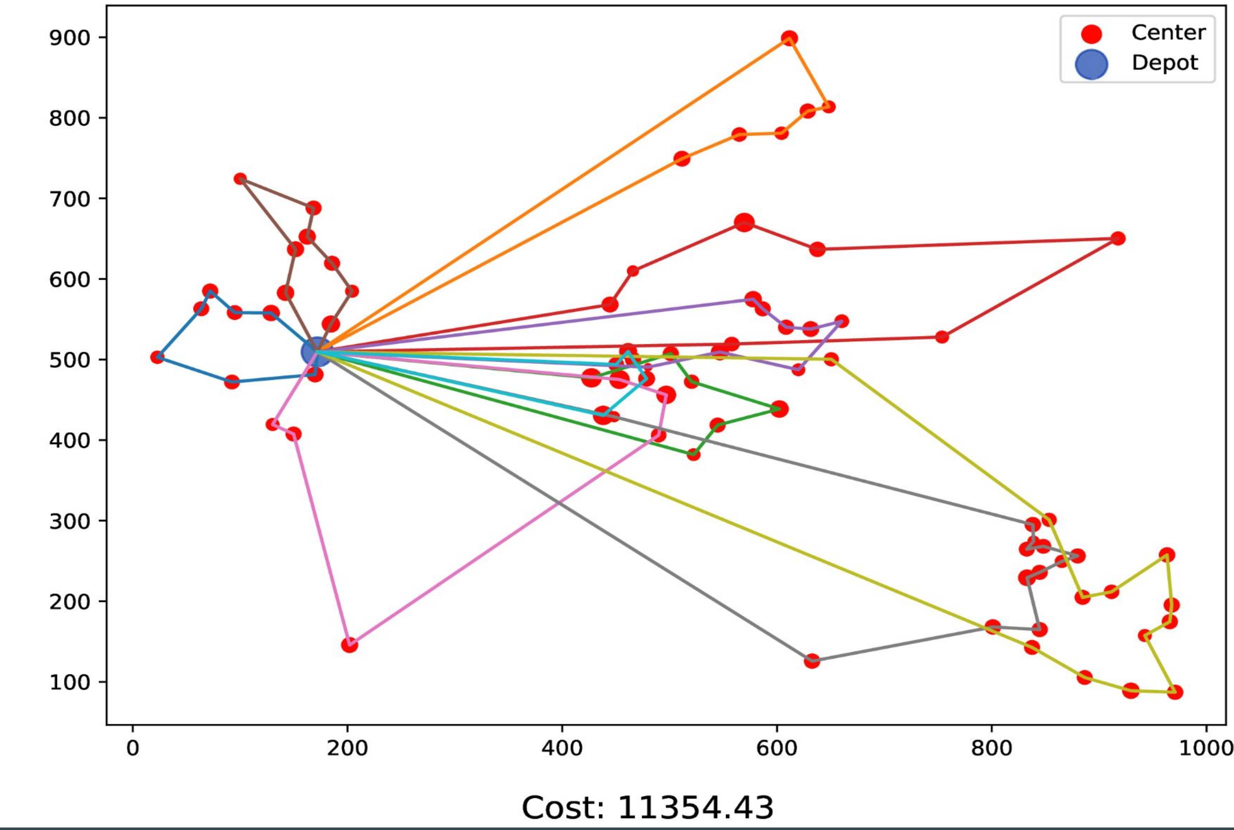
Modello: Column Generation Runtime: 22h Data: Real



Cost: 11456.30

Risultati

Modello: Polinomiale Runtime: 5h Data: Average Forecasting



Grazie per l’attenzione

Informazioni più dettagliate, soprattutto sul sottoproblema e sull’algoritmo di Branch and Price sono scritte nel report tecnico.

Domande?

Lorenzo Rossi matr. 183590