# A2A Challenge: Ottimizzazione Multi-Obbiettivo Vincolata del Routing dei Veicoli AMSA

Conte Enrico1, De Servi Federico1

1University of Milan-Bicocca, MSc - Data Science

Abstract

La sfida proposta da A2A consiste nell’ottimizzazione dei percorsi seguiti dalla flotta di 20 veicoli AMSA addetti allo svuotamento dei cestini pubblici sparsi per la città di Milano in un’area che va da Porta Venezia fino a Sesto San Giovanni. Il nostro studio si propone di comparare 3 diversi set di soluzioni che si differenziano per la tecnica adottata a monte per associare tra di loro i cestini. Nello specifico, la prima soluzione non adotta alcuna tecnica di clustering, la seconda si basa su un algoritmo di clustering K-Means applicato alle coordinate geografiche dei singoli cestini e il terzo, infine sfrutta una funzione matematica da noi definita per riassumere la dinamica di riempimento dei cestini, in modo da ottenere gruppi con comportamenti sufficientemente omogenei. Questi gruppi verranno poi assegnati a ciascun turno sulla base delle loro caratteristiche, al fine di massimizzare l’efficienza della raccolta. I migliori risultati in termini di Chilometri giornalieri percorsi sono stati ottenuti con il terzo metodo, grazie al quale si andrebbero a risparmiare un considerevole numero di Chilometri percorsi giornalmente senza danneggiare la qualità del servizio offerto. La Sezione 1 di questo report mira a fornire indicazioni circa la natura del problema in questione, la Sezione 2 illustra i processi di Pre-Processing che sono stati necessari, nella Sezione 3 andiamo nel dettaglio circa i metodi di clustering, nella Sezione 4 affrontiamo il problema del routing, nella Sezione 5 esponiamo i risultati ottenuti e, infine, nella Sezione 6 illustriamo le nostre conclusioni.

Keywords: Constrained Vehicle Routing Problem

**Indice**

1. **Introduzione e Modeling**
   1. Inquadramento del Problema
   2. Modeling del Problema
2. **Descrizione Dataset e Pre-Processing**
   1. Descrizione Dataset
   2. Pre-Processing
3. **Metodologie di Clustering**
   1. Clustering Geografico
   2. Clustering per Dinamica di Riempimento
4. **Routing**
   1. Calcolo matrice delle distanze
   2. Algoritmo di Routing Multi-obbiettivo
5. **Risultati**
   1. Confronto risultati ottenuti
   2. Visualizzazione dei percorsi ottenuti
6. **Conclusioni**

**Appendice**

1. **Introduzione e Modeling**
   1. **Inquadramento del Problema**

A2A rappresenta la più grande realtà italiana nel campo delle aziende Multi-Utility. Nello specifico, nel territorio urbano ed extraurbano della città di Milano, si occupa, tra le altre cose, anche del servizio di nettezza urbana tramite AMSA, sua controllata. La challenge lanciata al corso di Laurea Magistrale in Data Science dell’Università di Milano Bicocca consiste nell’individuazione dei percorsi ottimali che i 20 veicoli della flotta AMSA dovranno seguire ogni giorno durante lo svolgimento del loro compito di svuotamento dei cestini pubblici.

A tal fine, A2A ha installato degli “smart-bin” dotati di appositi sensori che ogni 15 minuti riportano il livello di riempimento del cestino in una scala da 1 a 4 (dove 1 è il minimo e 4 il massimo), le coordinate geografiche del cestino e il suo numero seriale.

* 1. **Modeling del Problema**

Una volta analizzate le richieste, è chiaro che ci si trova di fronte ad un problema di ottimizzazione vincolata multi-obbiettivo. In particolare, la prima risorsa da ottimizzare è la distanza totale di percorrenza dei mezzi, che deve essere minimizzata tenendo conto che ognuno dei 20 mezzi inizia e finisce il suo percorso al deposito di Via Olgettina 35 e che il numero massimo di cestini coperti da ogni mezzo è 150, come dichiarato da A2A. Inoltre, il secondo obbiettivo dell’ottimizzazione è la vuotatura dei cestini, che deve essere massimizzata, ovvero è necessario svuotare i cestini solo nel momento in cui presentano un livello di riempimento sufficientemente alto da giustificare la visita.

Sia *G = (V, E)* un grafo unidirezionale dove *V* è il set dei nostri cestini, detti *nodi* ed *E* è il set degli *archi* che connettono tra di loro i nodi. Ad ogni arco sono associati un costo di percorrenza , che rappresenta la distanza da percorrere per andare dal cestino al cestino , e una funzione di domanda che si aggiorna ad ogni visita e serve a limitare a 150 il numero di cestini visitati da ogni mezzo.

L’obbiettivo è determinare un set di percorsi che minimizzino il costo totale di percorrenza consentendo di svuotare ogni cestino almeno una volta al giorno senza eccedere la capacità massima di ogni veicolo fissata a 150 cestini.

In questo studio, ci poniamo l’obbiettivo di produrre 3 diversi scenari risolutivi per questo problema di *routing* e comparare i risultati ottenuti per identificare la soluzione ottima.

Lo scenario A propone dei percorsi calcolati dando in input all’algoritmo i dati non clusterizzati, nello scenario B calcoliamo i percorsi in base a dei clusters geografici e, infine, nello scenario C calcoliamo i percorsi in base ad una nostra proposta di clustering secondo il valore di una funzione *g(x)* che riassume la dinamica di riempimento dei cestini e associa tra di loro elementi che si comportano in maniera simile per fascia oraria, così da poter ottimizzare non solo la distanza percorsa ma anche, come detto in precedenza, la vuotatura dei cestini.

1. **Descrizione Dataset e Pre-Processing**
   1. **Descrizione Dataset**

Come detto in precedenza, il dataset fornitoci da A2A è il risultato di 36 giorni di rilevazioni effettuate ad intervalli di 15 minuti l’una dall’altra su un totale di 3148 cestini. Il dataset conta 814727 records e un totale di 23 attributi.

Tuttavia, gli unici attributi utili alle nostre analisi saranno ‘bin\_serial’, ‘bin\_level, ‘DataUltimaRilevazione’, ‘Longitudine’ e ‘Latitudine’.

* 1. **Pre-Processing**

Il primo ostacolo incontrato durante le fasi iniziali del lavoro ha riguardato la bontà dei dati forniti come input.

Infatti, volendo cercare cluster nei dati in funzione degli orari in cui i cestini risultano pieni, abbiamo subito notato come il timestamp ‘DataUltimaRilevazione’, riportasse sempre lo stesso orario per ogni rilevazione di ogni diverso cestino, come se fossero state fatte tutte allo stesso istante.

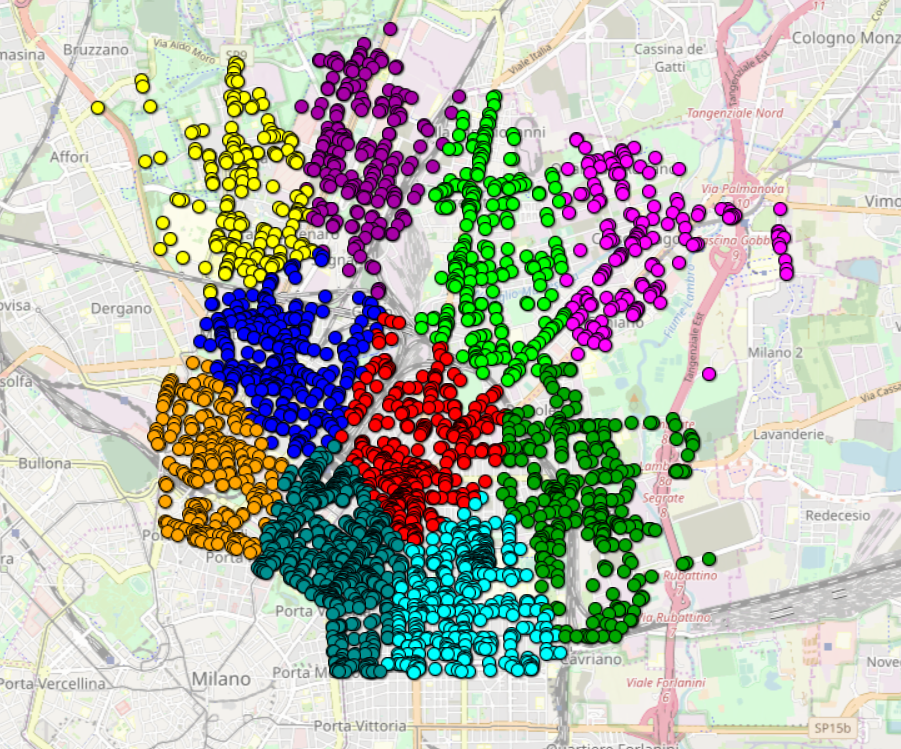
Ovviamente ciò non è logicamente possibile, quindi, abbiamo deciso di modificare i dati corrotti ipotizzando la frequenza di rilevazione che ci è stata comunicata (15 minuti), così da rendere possibile uno studio riguardo la dinamica di riempimento dei cestini.

In Fig.1 (vedi Appendice) è possibile vedere uno screen di quello che è il dataset dopo il pre-processing, che, in ogni caso, si è reso necessario unicamente al fine di rendere possibile il clustering per dinamica di riempimento, che si illustrerà nella sezione successiva.

1. **Metodologie di Clustering**
   1. **Clustering Geografico**

Al fine di associare gli elementi secondo la vicinanza geografica, abbiamo adoperato un algoritmo di clustering K-Means fissando il numero dei centroidi iniziali a 10.

Il numero dei cluster da trovare inizialmente è stato scelto in base al suo valore di Silhouette Score (0,39). In Fig.2 è possibile vedere la mappa dei cestini divisi in cluster geografici.



**Fig.2** Clustering Geografico visualizzato su mappa, link per la visualizzazione interattiva:

<https://drive.google.com/uc?export=download&id=1arWAAUDx2BJtQzL3h4ZQMqqlJKX9JiXG>

* 1. **Clustering per Dinamica di Riempimento**

L’ipotesi che abbiamo voluto mettere alla prova indagando circa le tendenze orarie di riempimento dei cestini è che sia più conveniente svuotarli non procedendo per aree geografiche ma seguendo dei percorsi che raccolgano tutti quei cestini che presentano trend di riempimento simili, in modo da non dover mai svuotare cestini con livello di riempimento basso ma, al contrario, svuotare solo quei cestini che sono pieni nella stessa fascia oraria.

Per permettere questa associazione, abbiamo calcolato, per ogni cestino, in 4 diverse fasce orarie (Mattino, Pomeriggio, Sera e Notte), una funzione *g(x)* definita come la media aritmetica della somma delle differenze positive tra il livello di riempimento i-esimo e il livello precedentemente registrato (escludendo quei casi in cui il livello di riempimento rimane uguale per diverse rilevazioni consecutive in quanto falserebbero i risultati). Ovvero:

*g(x)* *=*  (1)

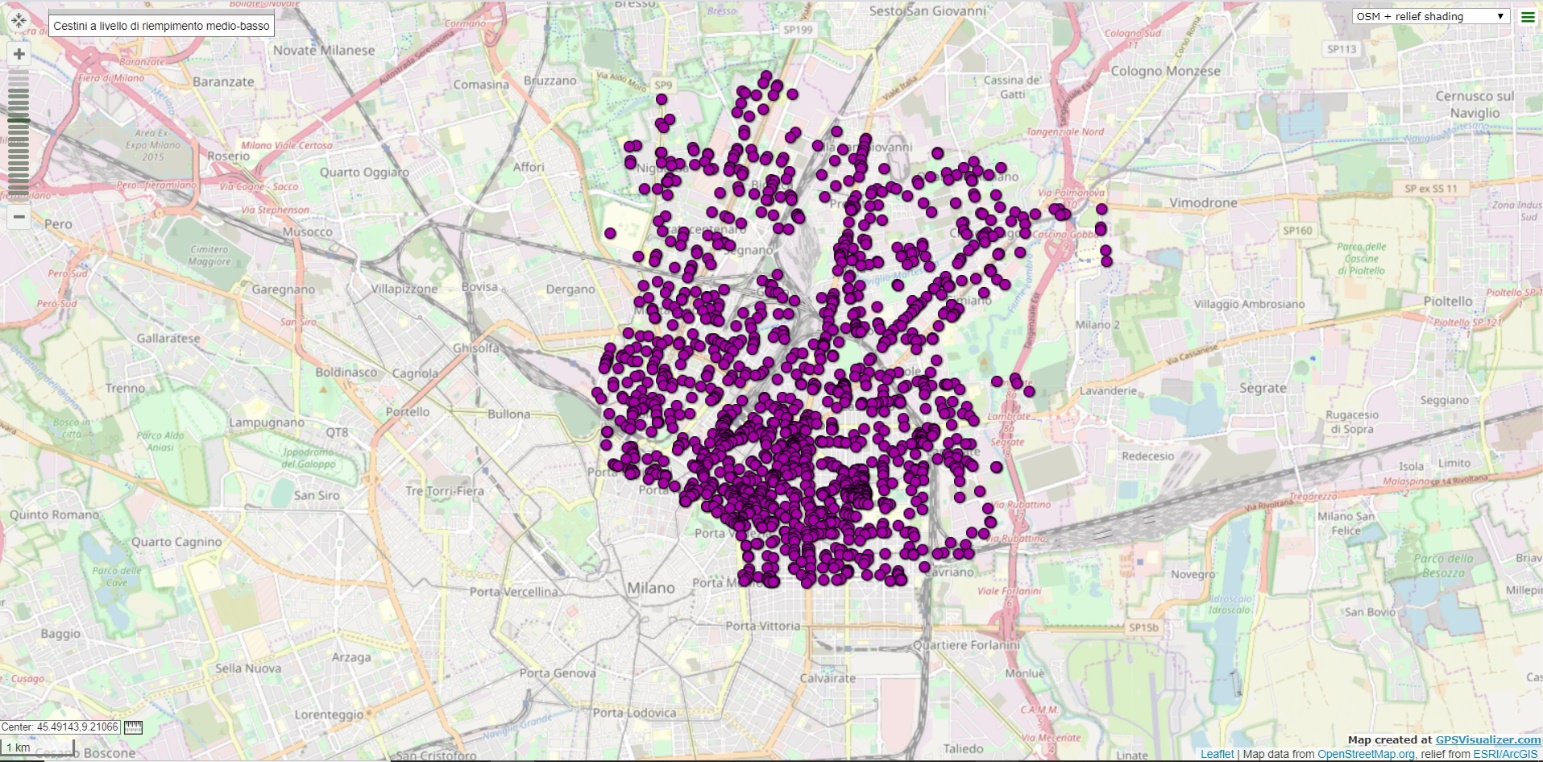
dove con *n* indichiamo il numero totale di rilevazioni per il cestino *x* all’interno della fascia oraria di riferimento e > . In questo modo, ad esempio, sappiamo che più il valore della funzione è alto, più il cestino si riempie velocemente.

Una volta ottenuto un valore medio della funzione *g(x)* per ogni cestino nelle 4 diverse fasce orarie abbiamo applicato nuovamente un algoritmo di clustering K-Means impostando il numero di centroidi inziali a 3 alla luce del valore di Silhouette Score (0,2479).

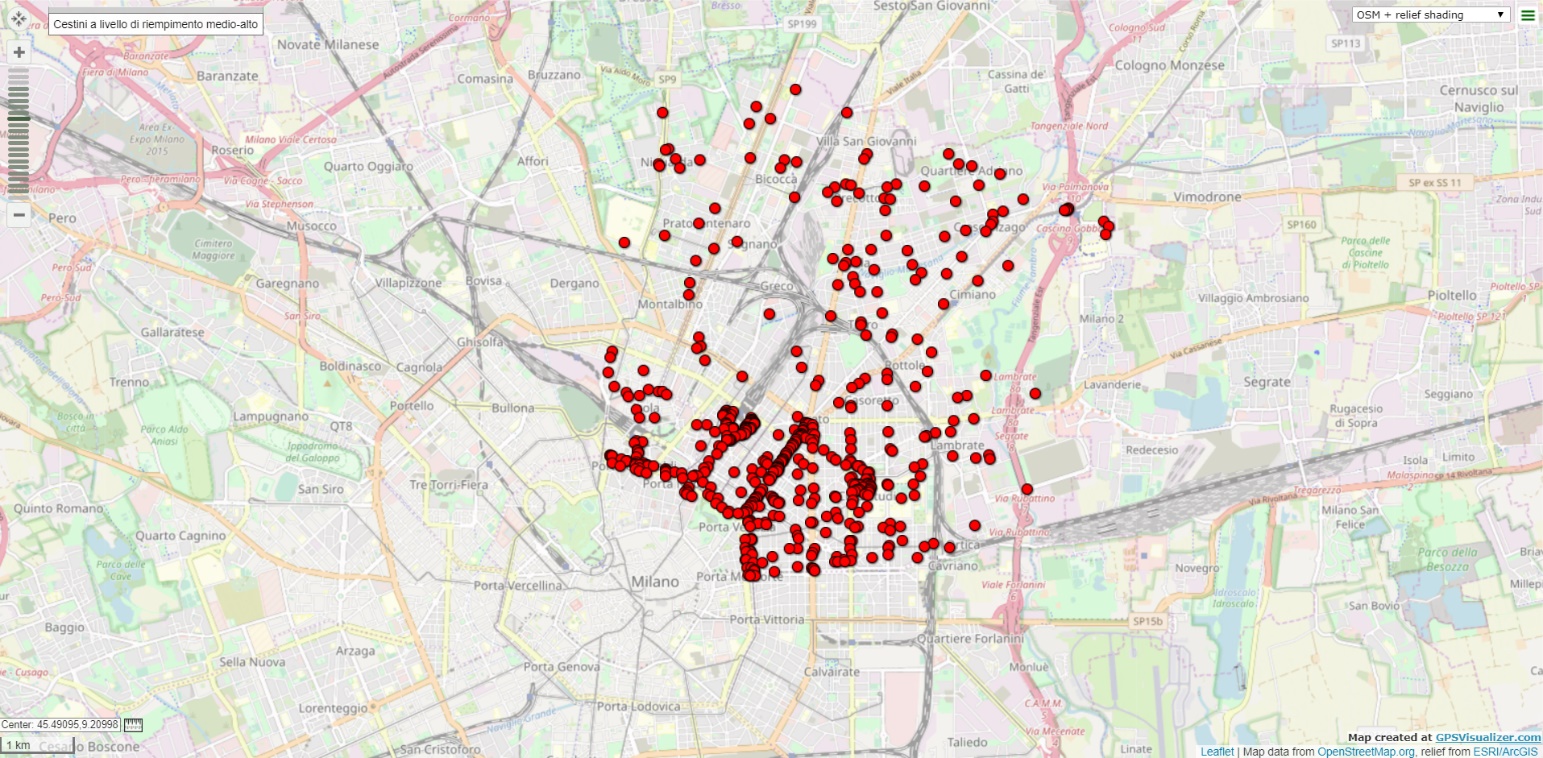
In Figura 3A, 4A e 5A possiamo vedere delle visualizzazioni dei 3 clusters su mappa, mentre le Figure 3B, 4B e 5B(vedi Appendice) mostrano l’aggregazione degli elementi secondo la funzione *g(x)* nelle diverse fasce orarie.

**Fig.3A**. Cluster dei cestini con velocità di riempimento bassa, link per la visualizzazione interattiva:

<https://drive.google.com/uc?export=download&id=1Zph2ZGBNVX_rv8Gqw-A9lLl5P8YgipiC>



**Fig.4A.** Cluster dei cestini a velocità di riempimento medio-bassa, link per la visualizzazione interattiva: <https://drive.google.com/uc?export=download&id=1zhc2QtGiObSKRO5inMxAJ7_o9mqqB8BL>

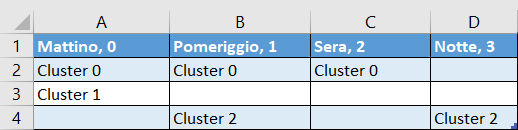


**Fig.5A.** Cluster dei cestini con velocità di riempimento medio-alta, link per la visualizzazione interattiva:

<https://drive.google.com/uc?export=download&id=1u-m1jj7_r8L-j8BPynaJX0iqd9fiLSWS>

Come è possibile vedere, i 3 clusters trovati si caratterizzano per una frequenza di riempimento rispettivamente bassa, medio-bassa e medio-alta.

Di conseguenza, in base ai risultati ottenuti, abbiamo assegnato ognuno dei 3 cluster ad almeno uno dei 4 turni che, come comunicato dall’azienda, vengono svolti giornalmente. Abbiamo poi deciso, in base ai centroidi di ciascuno dei 3 clusters, di svuotare i cestini con livello di riempimento medio.alto 3 volte al giorno, quelli con livello medio-basso 2 volte al giorno, ed infine quelli a livello basso 1 volta al giorno, in modo da ottimizzare la frequenza di vuotatura per quei cestini che si riempiono più velocemente di altri in certe fasce orarie. In Figura 6 è possibile vedere il nostro scheduling.



**Fig.6-** Scheduling pensato per ottimizzare la frequenza di vuotatura dei cestini

1. **Routing**
   1. **Calcolo Matrice delle Distanze**

L’operazione in assoluto più importante e computazionalmente intensa al fine di calcolare i percorsi ottimali è senza dubbio il calcolo della matrice delle distanze che separano i nostri 3148 cestini.

La misura di distanza adottata è stata calcolata tramite un’istallazione locale del *routing engine* OSRM. Le distanze così ottenute corrispondono alla più breve distanza stradale che separa i diversi cestini tenendo conto dei diversi sensi di marcia attualmente in vigore.

Di conseguenza, abbiamo calcolato una matrice delle distanze che comprende tutti i cestini che ci serve per calcolare i percorsi ottimali in caso di assenza di clustering, 10 matrici delle distanze contenenti ognuna gli elementi di uno dei 10 cluster geografici e, infine, 4 matrici delle distanze che contengono gli elementi dei clusters assegnati ad ognuno dei 4 turni giornalieri (Mattino, Pomeriggio, Sera e Notte).

Si noti come nessuna di queste matrici sia simmetrica in quanto, come accennato precedentemente, stiamo consideriamo i sensi di marcia. Appare quindi evidente che *d(x, y)* non sarà più necessariamente uguale a *d(y, x)* dove con *d(x, y)* intendiamo la distanza tra il nodo *x* e il nodo *y*.

* 1. **Algoritmo di Routing Vincolato Multi-obbiettivo**

Una volta calcolate le matrici delle distanze, al fine di risolvere il problema di ottimizzazione matematica, per ciascuno dei 3 casi proposti (assenza di clusters, clustering geografico e clustering per dinamica di riempimento) abbiamo fatto ricorso alla libreria di funzioni OR-Tools messe a disposizione da Google per risolvere problemi di *vehicle routing.*

Nello specifico, l’algoritmo che abbiamo implementato permette di specificare liberamente il numero massimo di mezzi a disposizione, la lunghezza massima che ogni veicolo può percorrere (compreso il viaggio di ritorno verso il deposito di Via Olgettina 35) e il numero massimo di cestini servibili da ogni veicolo.

La strategia di ricerca prevede che, dal nodo di partenza *i*, si cerchi il nodo *j* che corrisponde all’arco con minore costo associato , e, una volta effettuata la connessione, il processo viene iterato di nodo in nodo.

Nella nostra implementazione, dopo svariati esperimenti, abbiamo deciso di impostare il limite di distanza percorribile da ogni mezzo a 50km, il numero massimo di cestini visitabili da ogni mezzo è stato impostato, come detto in precedenza, a 150, e, infine, abbiamo impostato un timer per la ricerca dell’arco più “economico” a 2 ore.

Nonostante i risultati ottenuti siano soddisfacenti, i parametri sopra citati sono facilmente modificabili.

1. **Risultati**
   1. **Confronto Risultati Ottenuti**

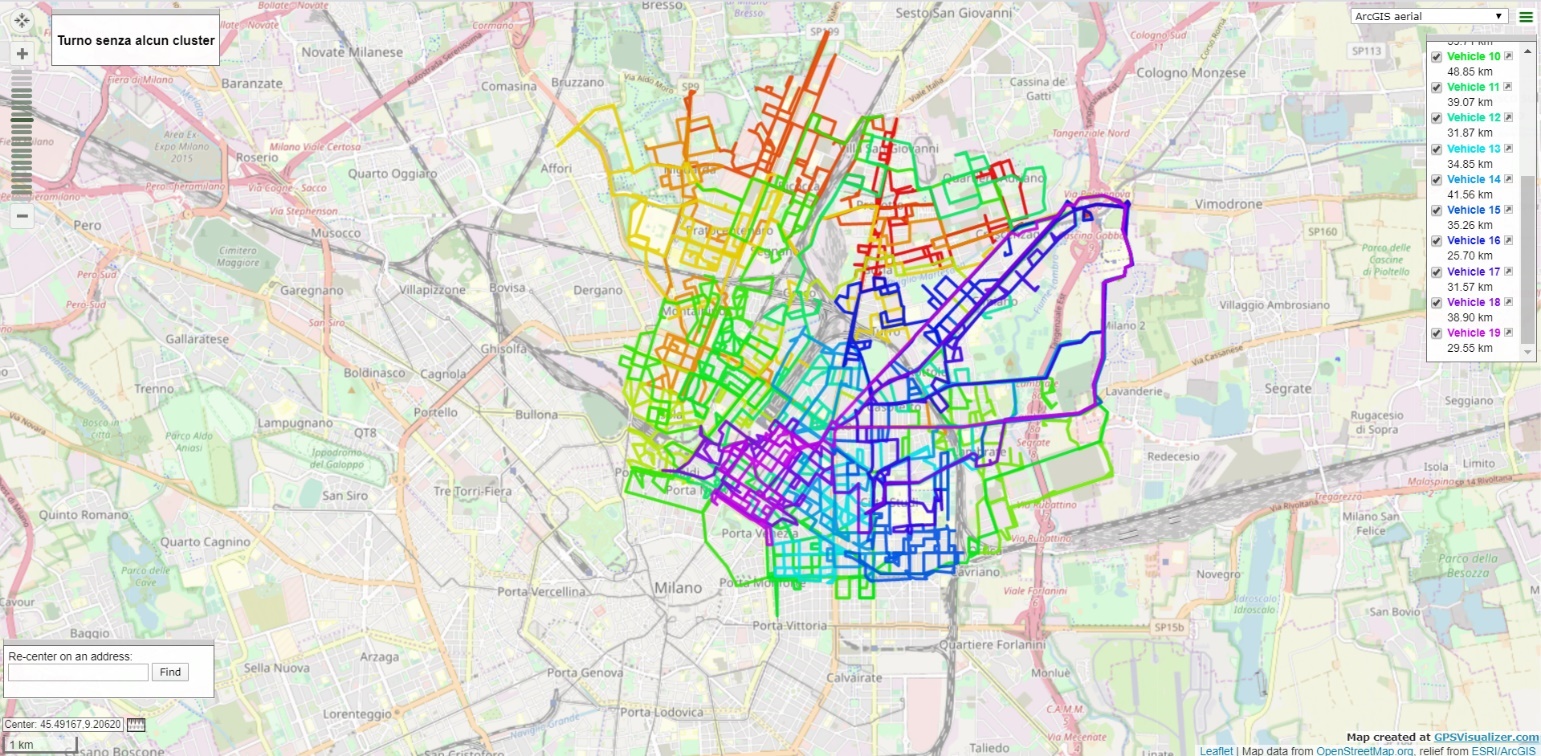
Prima di esporre i risultati ottenuti in termini di chilometraggio complessivo, è utile ricordare le funzioni obbiettivo del problema. Infatti, l’obbiettivo non è solo minimizzare la distanza percorsa, ma anche massimizzare l’efficacia della vuotatura in relazione allo stato di riempimento del singolo cestino, in altre parole, bisogna svuotare i cestini solo quando sono del tutto pieni per evitare di doverli svuotare nuovamente durante uno dei successivi turni giornalieri, che, ricordiamo, sono 4 e si dividono in turno mattutino, pomeridiano, serale e notturno.

Il primo obbiettivo viene raggiungo da ognuna delle 3 soluzioni proposte, mentre, il secondo obbiettivo, la massimizzazione dell’efficacia della vuotatura, può essere raggiunto solo tramite il clustering per dinamica di riempimento e la successiva suddivisione dei cluster ottenuti nelle suddette fasce orarie. Inoltre, nonostante il numero complessivo di mezzi a disposizione sia 20, l’algoritmo è progettato per ridurre al minimo il numero di mezzi necessari a coprire l’area di interesse, e, di conseguenza, vengono utilizzati, effettivamente, meno di 20 mezzi.

Inoltre, insieme alla visualizzazione statica delle mappe includiamo un link di Google Drive al quale è possibile scaricare il file html relativo alla visualizzazione di interesse per permetterne una consultazione interattiva e maggiormente granulare. Fatte queste precisazioni si può procedere con il report dei risultati.

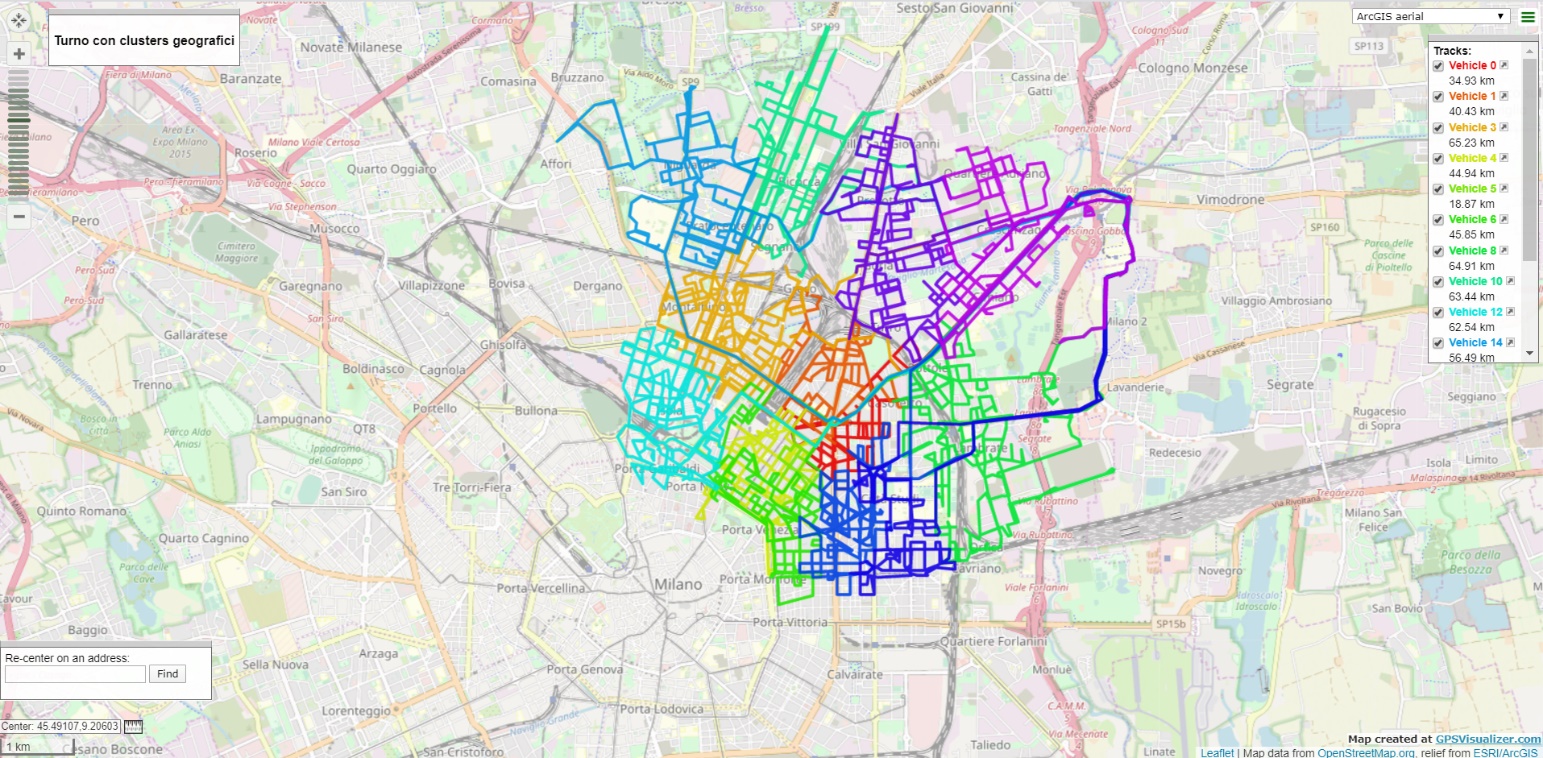
Nello scenario A, ovvero quello in cui abbiamo calcolato i percorsi senza prima effettuare alcun clustering dei dati (Fig.6), ognuno dei 4 turni giornalieri impegna 17 veicoli ed è complessivamente lungo 674.9Km, il che implica che la distanza totale percorsa giornalmente è di 2699.6Km.

https://drive.google.com/open?id=1u-m1jj7\_r8L-j8BPynaJX0iqd9fiLSWS

****

**Fig.6. Scenario A**, nessun Clustering a monte

Nello scenario B, ovvero quello in cui abbiamo calcolato i percorsi ottimi in base ad un clustering geografico in 10 zone (Fig.7), ognuno dei 4 turni giornalieri impegna 14 mezzi ed è complessivamente lungo 683.8Km. La distanza totale percorsa giornalmente adottando questo metodo è di 2735.2Km. La maggiore distanza complessiva si spiega alla luce dei maggiori vincoli ai quali sono sottoposti i camion, che sono confinati nella scelta delle strade da poter percorrere.

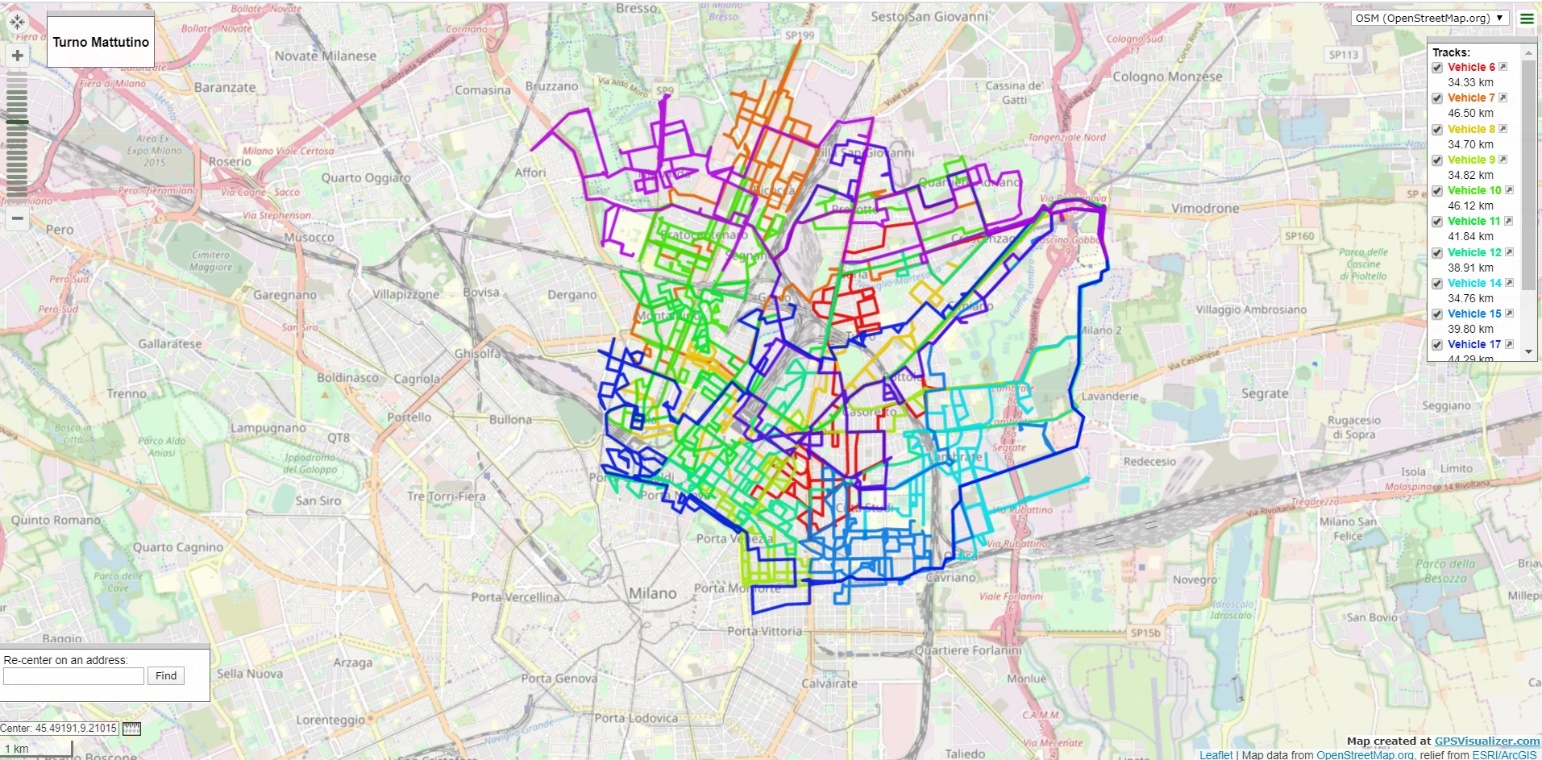


**Fig.7. Scenario B**, Clustering Geografico

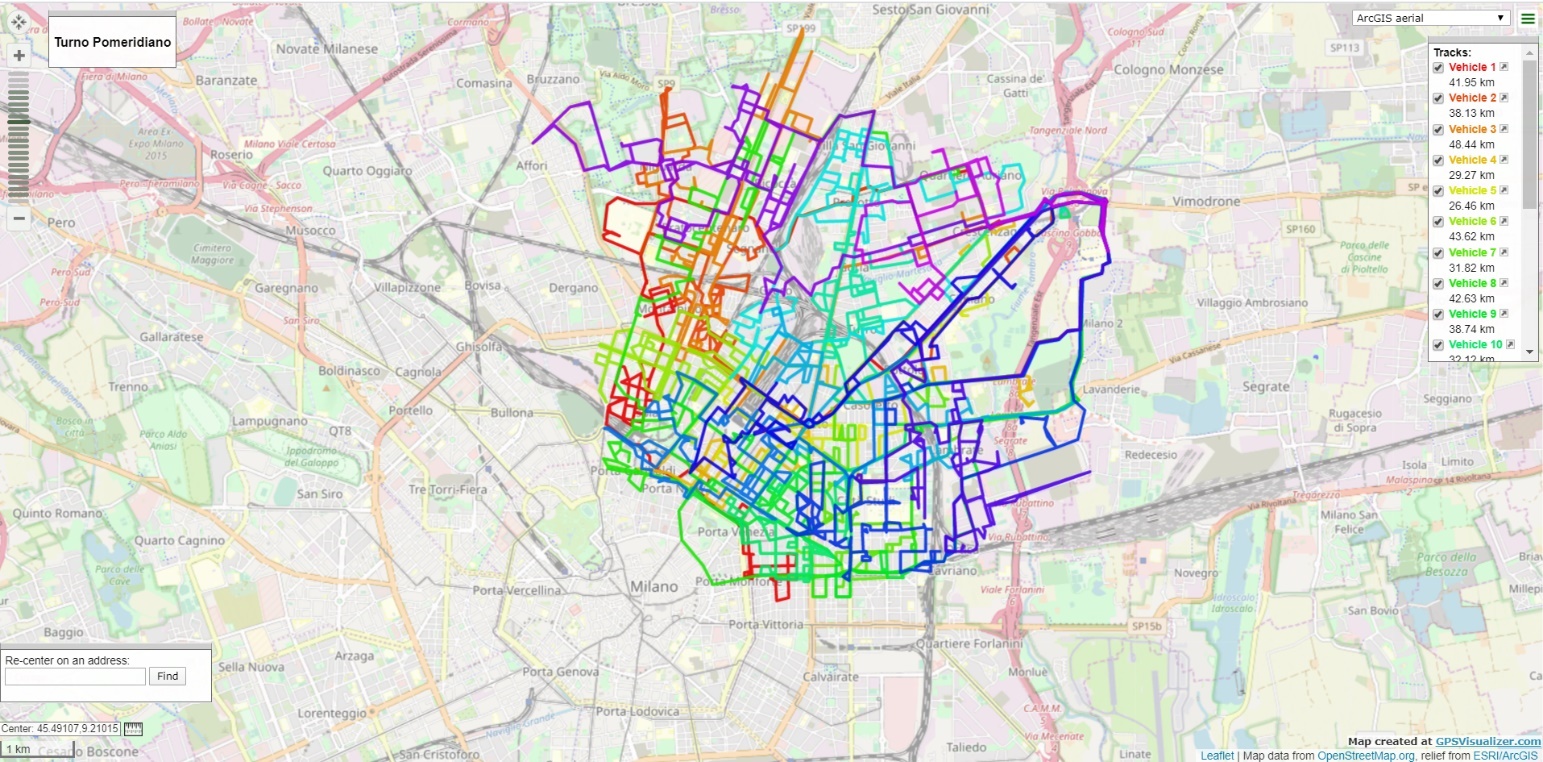
Infine, nello scenario C, ovvero quello in cui abbiamo calcolato i percorsi ottimi in base ad un clustering per dinamica di riempimento che ha prodotto 3 clusters (Fig. 8,9 e 10), i 4 turni giornalieri presentano le seguenti caratteristiche:

* Turno Mattutino: impegna 12 mezzi ed è complessivamente lungo 478.9Km
* Turno Pomeridiano: impegna 19 mezzi ed è complessivamente lungo 666.4Km
* Turno Serale: impegna 9 mezzi ed è complessivamente lungo 394.3Km
* Turno Notturno: impegna 10 mezzi ed è complessivamente lungo 419Km

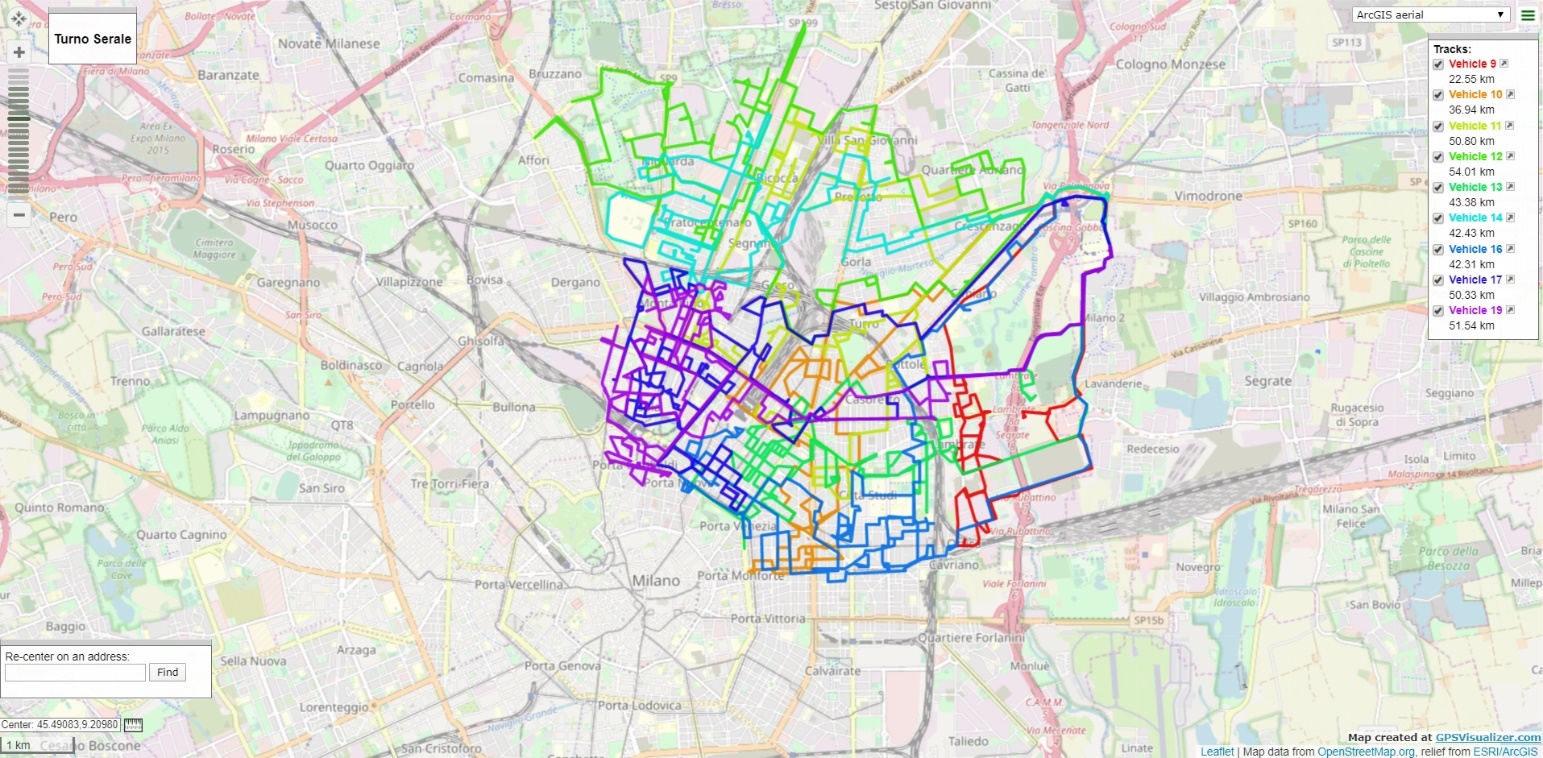
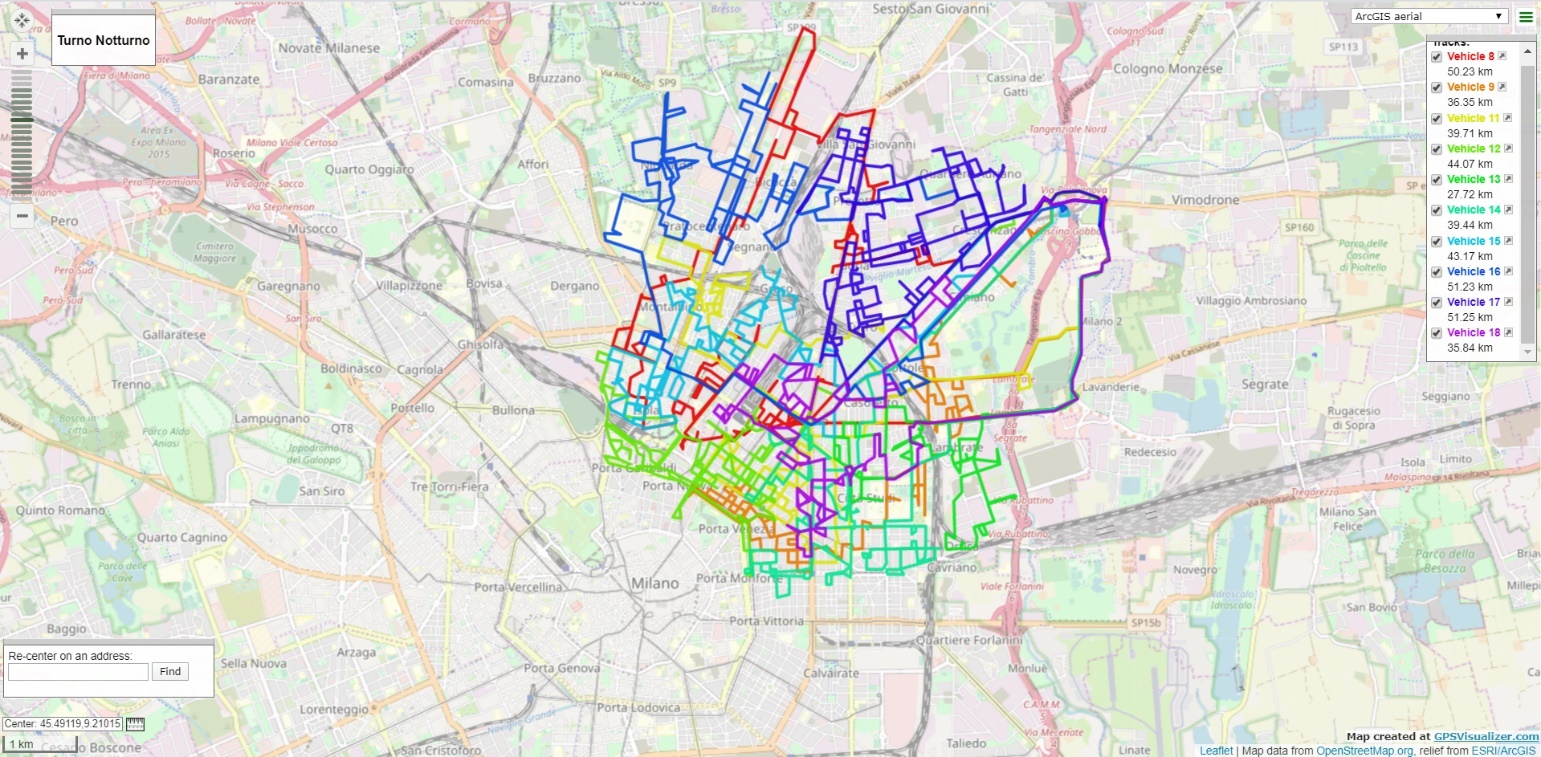
Sommando le lunghezze dei singoli turni si arriva ad un totale giornaliero di 1958.6Km.



**Fig.8. Scenario C,** Turno Mattutino



**Fig.9. Scenario C,** Turno Pomeridiano

**Fig.10. Scenario C,** Turno Serale

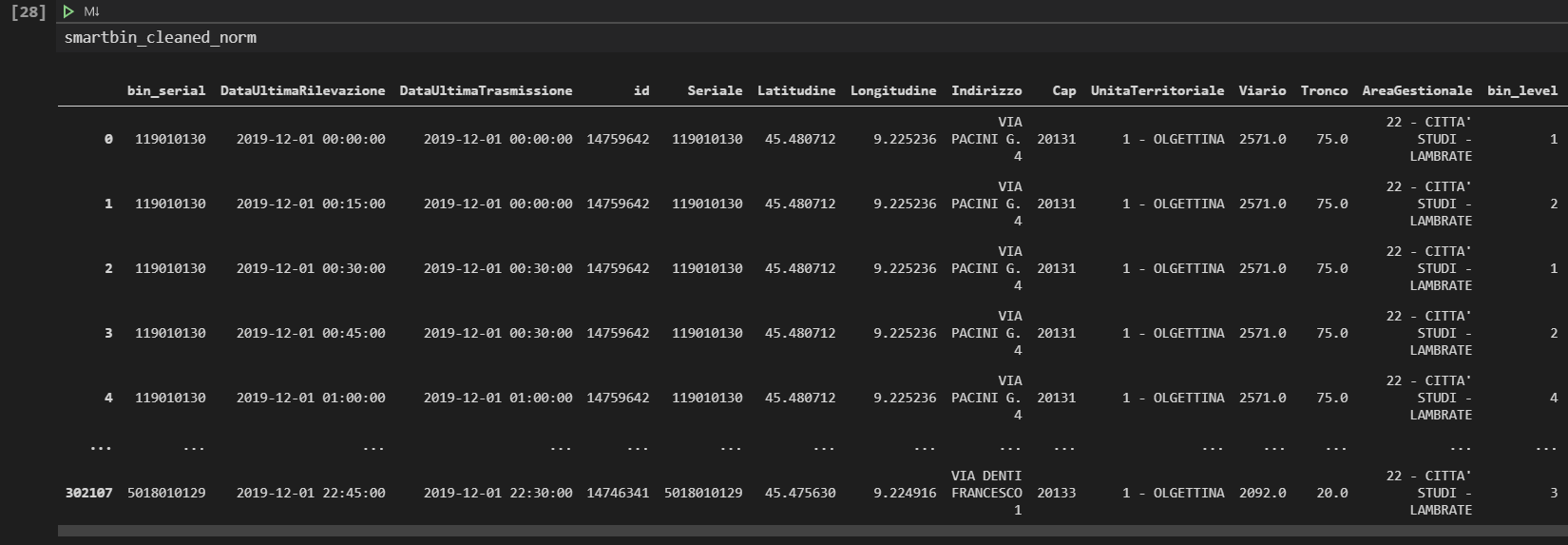
**Fig.11. Scenario C,** Turno Notturno

1. **Conclusioni**

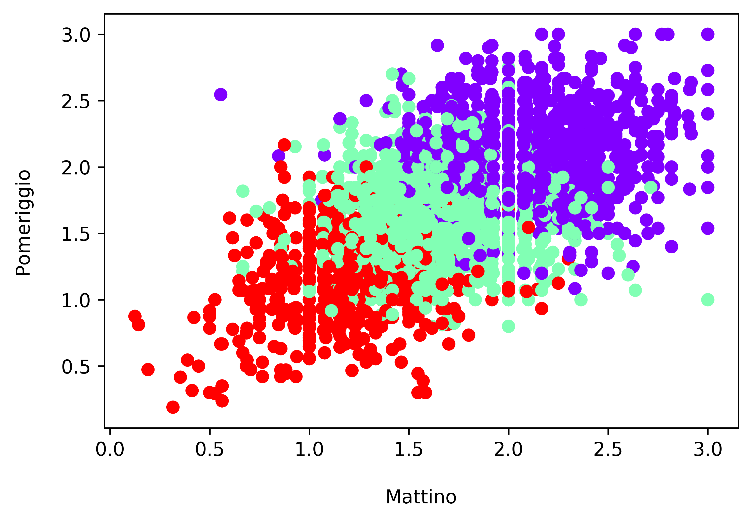
Considerati i risultati ottenuti, lo scenario C è quello che produce i risultati migliori sia in termini di risparmio di veicoli utilizzati sia in termini di distanza complessivamente percorsa giornalmente, portando a risparmiare 741Km al giorno rispetto allo scenario A dove non si opera alcun clustering, e 776.6Km al giorno rispetto allo scenario B, dove si opera un clustering di tipo geografico.

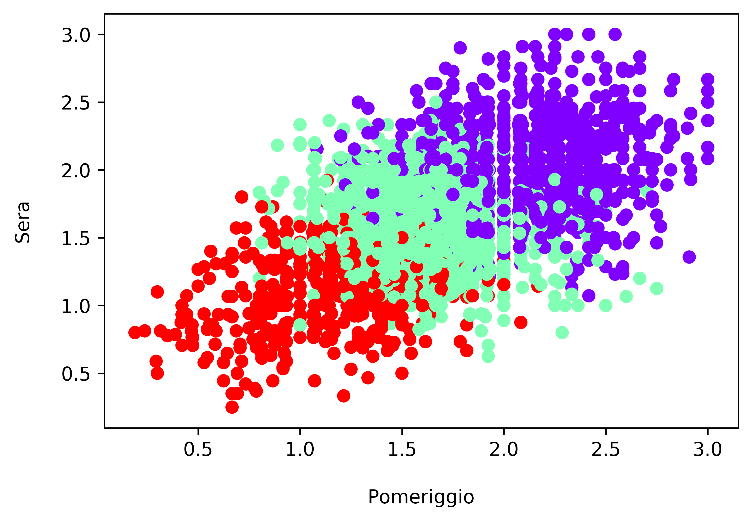
Inoltre, il metodo C è l’unico che permette sia di minimizzare il chilometraggio complessivo sia di massimizzare l’efficienza della vuotatura dei singoli cestini e quindi soddisfa entrambi gli obbiettivi del nostro studio.

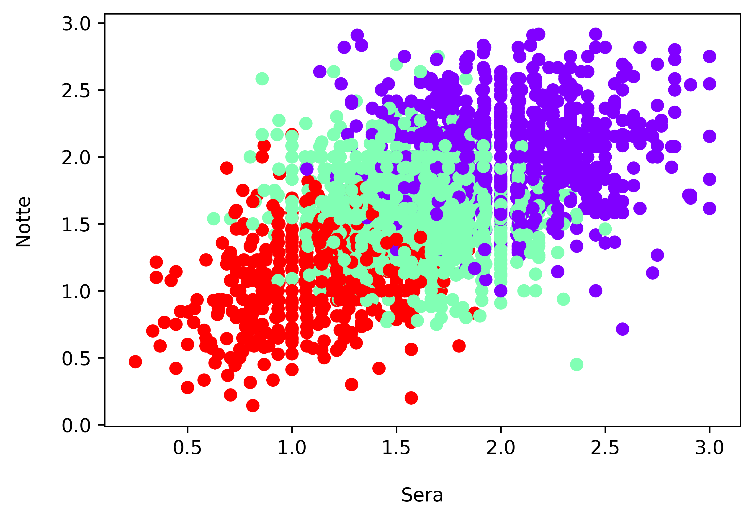
**Appendice**



**Fig.1**- Dataset dopo il pre-processing







**Fig 3B,4B e 5B-** Visualizzazione dei 3 clusters divisi per fascia oraria. Negli assi leggiamo i valori della funzione *g(x)*