Agenzia nazionale per le nuove tecnologie, l'energia e lo sviluppo economico sostenibile

MINISTERO DELLA TRANSIZIONE ECOLOGICA

Simulazione ed analisi di strategie di ride-sharing e valutazione dei risultati

Stefano Chiesa, Vincenzo Nanni, Sergio Taraglio

Report RdS/PTR(2021)/201

SIMULAZIONE ED ANALISI DI STRATEGIE DI RIDE-SHARING E VALUTAZIONE DEI RISULTATI (LA 2.21)

Stefano Chiesa, Vincenzo Nanni, Sergio Taraglio (ENEA, TERIN-SEN-RIA)

Dicembre 2021

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero della Transizione Ecologica - ENEA
Piano Triennale di Realizzazione 2019-2021 - III annualità 2021
Obiettivo: *Tecnologie*
Progetto: Tecnologie per la penetrazione efficiente del vettore elettrico negli usi finali
Work package: Mobilità
Linea di attività: *LA 2.21* Integrazione delle nuove funzionalità nella piattaforma e testing
Responsabile del Progetto: Claudia Meloni, ENEA
Responsabile del Work package: Maria Pia Valentini ENEA

Indice

Sommario

In questo documento sono descritte le attività compiute nell'ambito della linea di attività LA 2.21. Esse si sono concentrate in due diversi ambiti di lavoro, lo studio e lo sviluppo di un simulatore di ride sharing e l'analisi di alcuni scenari tramite di esso; ed il disegno e lo sviluppo di una interfaccia tra il modello di ride sharing e quello di generazione di traffico, sviluppato nella precedente annualità con l'interfaccia grafica realizzata in un'altra linea di attività.

Utilizzando i viaggi generati con il modello di generazione del traffico, è stato sviluppato un modello di ride sharing che può essere schematizzato in due fasi. La prima prevede la costruzione di matrici origine destinazione orarie, all'interno delle quali si estraggono i viaggi che potenzialmente potrebbero dare origine a ride sharing, ovvero quei viaggi che arrivando in una medesima zona entro un dato intervallo orario siano originati da posizioni ricadenti in una medesima zona in altre zone della città. La seconda fase del sistema prevede la simulazione dei viaggi di ride sharing in modalità dinamica, ovvero le auto di car sharing sono seguite nella loro evoluzione temporale che le può portare ad essere riutilizzate nella zona di arrivo in tempi successivi.

Si sono poi studiate le prestazioni di un tale scenario di ride sharing al variare di alcuni dei parametri che lo governano, tra questi la dimensione delle zone di arrivo e/o partenza, l'intervallo temporale, il numero di veicoli utilizzati e, soprattutto, la distribuzione iniziale dei veicoli nella città.

Come detto, nel secondo ambito di lavoro si è curata l'interfaccia tra i moduli di simulazione del traffico e di ride sharing con l'interfaccia grafica del sistema EMU. Tramite questa interfaccia è possibile inizializzare i parametri delle due simulazioni e permetterne l'avvio. Al termine delle simulazioni è possibile lo scambio dei risultati, tramite file, per la corretta visualizzazione sull'interfaccia stessa.

1 Introduzione

Nel corso del secondo anno del Piano Triennale 2019-2021, la parte principale delle attività è stata dedicata allo studio e alla realizzazione di un modello di simulazione di richiesta di traffico per la Città Metropolitana di Roma.

Tale modello è stato realizzato utilizzando un approccio basato sull'utilizzo di algoritmi di Machine Learning, in particolare si è utilizzato un approccio basato sulla realizzazione di un'architettura neurale impiegata come auto encoder variazionale (VAE). Questo approccio è in grado di catturare la struttura statistica dei dati sui quali viene addestrato e quindi essere impiegato per produrre traiettorie di traffico sintetico che ricalchino le caratteristiche statistiche del dataset originale. Ciò permette da un lato di salvaguardare la privacy, essendo le traiettorie simili ma non uguali a quelle registrate e, dall'altro, di poter produrre quante traiettorie siano necessarie alla simulazione da compiere.

Questo modello è stato utilizzato nelle attività della presente annualità per generare la richiesta di traffico alla quale dare una parziale risposta tramite il cosiddetto ride sharing, con l'obiettivo di ridurre il traffico complessivo.

Una seconda parte delle attività della presente annualità sono state rivolte alla realizzazione di un'interfaccia tra i modelli di generazione di traffico e di ride sharing ed una interfaccia grafica di front end sviluppata in una differente linea di attività. L'interfaccia permette sia di comandare la generazione di traiettorie sintetiche che di permettere la condivisione dei dati di output per l'opportuno display.

1.1 Definizione di ride sharing

La mobilità urbana in tempi recenti ha subito profonde trasformazioni creando diversi nuovi concetti quali, ad esempio, il car sharing, il car pooling ed il ride sharing. Le definizioni di questi concetti sono variabili a seconda della fonte utilizzata, ma possono comunque essere schematizzate come segue.

- Car sharing: autonoleggio a tempo di un'automobile di proprietà di terze parti (in Italia solo le aziende possono dare a noleggio un'auto, in altri Paesi anche i privati). Sono le auto che possono essere noleggiate a breve termine tramite l'utilizzo di app su telefono cellulare.
- Car pooling: uso condiviso di veicoli privati tra due o più persone che devono percorrere uno stesso itinerario, o parte di esso, senza finalità di lucro (ad esempio BlaBlaCar).
- Ride sharing: attività di trasporto di terzi da parte di un privato con un'automobile di proprietà, con o senza finalità di lucro. Se con finalità di lucro viene talvolta chiamato Ride sharing on-demand.

Di queste tre modalità di mobilità condivisa, di seguito verrà considerata la terza, ovvero il ride sharing rilassando però il vincolo di proprietà. Nel prosieguo si assume che gli autoveicoli a disposizione per il ride sharing possano essere sia veicoli elettrici autonomi, ovvero privi di conducente, che veicoli non autonomi, in ogni caso messi a disposizione della collettività. Il singolo veicolo viene utilizzato per il ride sharing, ma non è di proprietà di alcuno dei passeggeri e rimane disponibile per successivi itinerari una volta che l'itinerario corrente sia concluso.

Il ride sharing è dunque una soluzione di mobilità condivisa che prevede la possibilità di utilizzare uno stesso veicolo da parte di più persone che desiderino recarsi nella medesima zona della città. Più in generale la Sharing Mobility consiste in una trasformazione del comportamento degli individui che tendono progressivamente a preferire l'accesso temporaneo ai servizi di mobilità in modo condiviso, piuttosto che utilizzare il proprio mezzo di trasporto.

Un servizio di mobilità può essere condiviso tra più utenti in due modi:

- contemporaneamente, quando si è per esempio all'interno di un vagone della metropolitana ma anche quando si fa parte di un equipaggio che si è formato con BlaBlaCar;

- in successione, come accade quando si preleva un'automobile di un qualunque servizio di car sharing ma anche salendo su un taxi o su un'auto di Uber.

Le recenti innovazioni che hanno investito questo settore hanno permesso che la condivisione di veicoli possa avvenire con costi di transazione notevolmente più bassi rispetto al passato e questo ha permesso che ad essere condivisi oggi siano veicoli normalmente concepiti per un uso personale.

Esistono chiaramente tutta una serie di problematiche sia di carattere sociale che di carattere logistico-organizzativo. Omettendo gli aspetti prettamente sociali, quali il problema di interazioni piacevoli o spiacevoli con i co-passeggeri o quello della sicurezza personale [1], il focus dell'attività qui presentata è una metodologia per la distribuzione dei veicoli per ride sharing nel tessuto urbano e la successiva analisi dei risultati per misurare l'eventuale riduzione dei flussi di traffico.

1.2 Il modello di richiesta di traffico

Il modello di richiesta di traffico è stato sviluppato nel corso della precedente annualità e descritto nel relativo Rapporto Tecnico [2]. Esso è un modello generativo realizzato con un auto encoder variazionale [3]. Un auto encoder variazionale è un'architettura di rete neurale che durante il suo addestramento impara la distribuzione statistica dei dati di un dataset e, una volta addestrata, è in grado di replicare, producendo nuovi dati distribuiti statisticamente nel medesimo modo [4, 5].

I dati di partenza sono quelli relativi ad un dataset della OCTO Telematics [6] organizzato per tracce con informazioni relative a posizione, tempo e stato del motore.

Il dataset originale è stato poi organizzato per tragitti giornalieri e per soste intermedie, con un massimo di otto soste; questo numero rappresenta un buon compromesso tra un elevato numero di traiettorie ed uno limitato di punti sperimentali. Ogni sosta è descritta da sei quantità: latitudine, longitudine, ora di inizio della sosta, durata della sosta, giorno della settimana, lunghezza del tragitto compiuto per arrivare alla sosta.

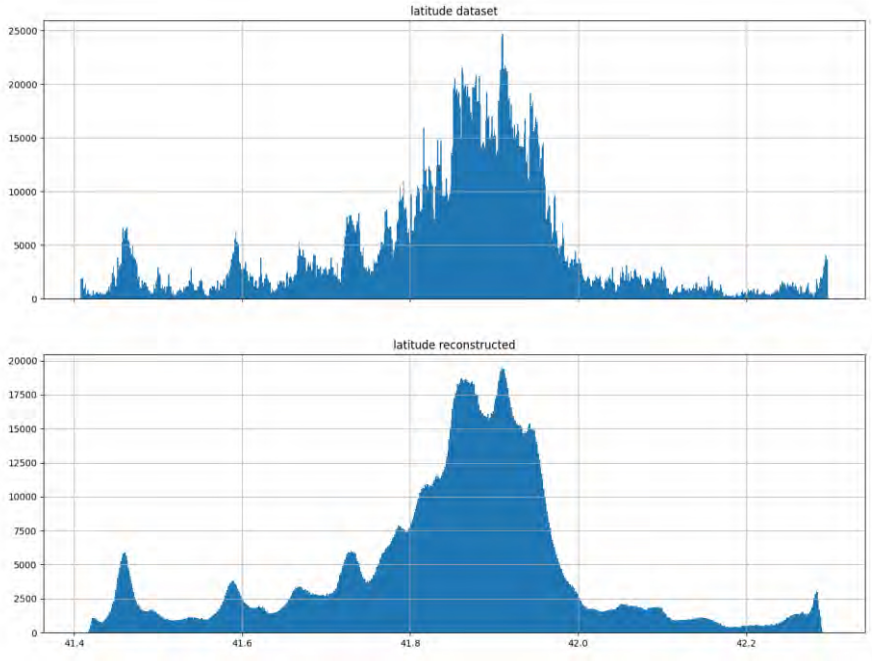


Figura 1. Distribuzione della latitudine delle soste. In alto nel dataset, in basso come ricostruita dal modello

Allo scopo di mostrare cosa si intenda per produzione di dati statisticamente distribuiti nel medesimo modo del dataset originale, in Figura 1 viene riportata la distribuzione delle latitudini delle soste come presenti nel dataset di addestramento, in figura in alto, e come ricostruite dal modello generativo, in basso.

E' chiaro come il sistema sia in grado di modellare i dati, producendone di nuovi con una distribuzione più *liscia*, ma ben riproducente la distribuzione iniziale.

Il modello di richiesta di traffico sviluppato nella seconda annualità è utilizzato nel corso delle attività di quella presente come generatore della domanda di traffico che debba essere parzialmente soddisfatta con una qualche strategia di ride sharing.

2 Descrizione delle attività svolte e risultati

Le attività della presente annualità si sono articolate nella realizzazione di una strategia per l'allocazione di veicoli in ride sharing e la conseguente analisi delle ricadute di una tale strategia, introducendo una metodologia di analisi basata su alcune quantità misurabili. Altra attività è quella relativa all'interfacciamento dei modelli di richiesta di traffico e di ride sharing con il modulo di display realizzato in altra linea di attività.

Le attività si sono articolate in 6 sotto attività:

- il processing dei dati del modello: ovvero i dati di uscita del modello generativo sono stati riaggregati in termini di singoli spostamenti;
- disegno di una strategia per l'allocazione dei veicoli di ride sharing;
- analisi delle caratteristiche di algoritmi di clustering:
- realizzazione di un semplice algoritmo di tassellatura dello spazio: analisi spaziale e temporale;
- realizzazione di un algoritmo per la simulazione del ride sharing urbano: analisi dei flussi e dei vantaggi connessi.
- Interfacciamento dei moduli realizzati con l'interfaccia grafica.

2.1 Il processing dei dati del modello

Come precedentemente esposto, il modello di richiesta di traffico produce sequenze di spostamenti giornalieri espressi tramite le soste intercorrenti tra i singoli spostamenti delle sequenze. In particolare sono stati generati 782213 giorni-veicolo, dove per giorno-veicolo si intende il numero di soste che compie un veicolo in un giorno (pari a 8). Questo numero è uguale al numero di dati utilizzati nell'addestramento del modello.

Allo scopo di studiare la richiesta di trasporto urbano è utile cambiare il sistema di riferimento ed esaminare i singoli spostamenti, spostando l'attenzione dal singolo veicolo seguito nell'arco della giornata ai singoli tragitti. A questo scopo si sono trasformati i dati di uscita del modello in singoli viaggi.

Ogni veicolo compie al più in un giorno 7 viaggi in quanto la sua stop-trajectory è composta di 8 stop, si definisce un viaggio quello che parte da uno stop e arriva al successivo. Le 8 soste giornaliere vengono quindi accoppiate in modo sequenziale a formare 7 viaggi, vedi Figura 2.

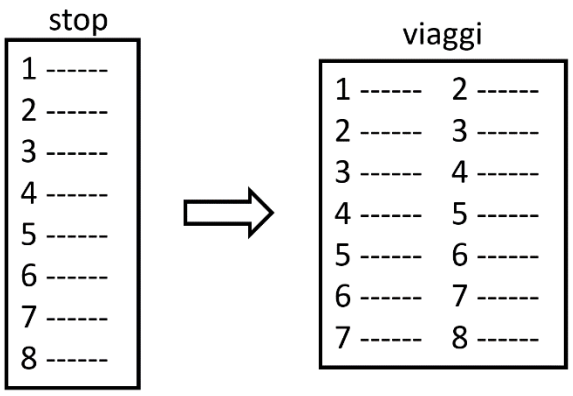


Figura 2. La trasformazione da 8 stop a 7 viaggi

Ogni viaggio è descritto da 10 parametri (latitudine e longitudine del punto di partenza; ora di partenza; giorno di partenza; latitudine e longitudine del punto di arrivo; durata del viaggio; ora di arrivo; giorno di arrivo; lunghezza del percorso)

Le dimensioni dei viaggi generati sono quindi (782213, 7, 10). Questi 782213 "giorni" contengono potenzialmente 782213 * 7 viaggi, ma, come precedentemente detto, una parte degli stop potrebbe non esistere in quanto l'utente potrebbe aver compiuto meno di 8 stop giornalieri. Il numero di viaggi effettivo, ossia con una partenza e una destinazione valide, sono 2.458.484. Questo numero è coerente con il numero medio di fermate in un giorno nel dataset originale di addestramento, pari a 4.99.

2.2 Disegno di una strategia per il ride sharing

Il ride sharing è qui inteso come l'utilizzo da parte di più persone di un medesimo veicolo, elettrico ed autonomo, da una posizione di partenza ad una di arrivo entrambe comuni. In questo lavoro non è contemplata l'opzione di 'raccolta' o discesa di passeggeri lungo il percorso.

Nell'intervallo di tempo in cui più persone necessitano di uno spostamento dallo stesso punto di partenza allo stesso punto di arrivo, entro un dato raggio, viene utilizzato un veicolo di ride sharing con un certo numero di posti disponibili. Si assume che il veicolo possa trasportare fino ad un massimo di tre passeggeri. L'idea di fondo è la seguente.

Dato un orario ed un intervallo di tempo, ad esempio una finestra di 15 minuti, si vanno a selezionare tutte le destinazioni appartenenti all'intervallo di tempo. Tra tutte le destinazioni selezionate, si vanno poi a estrarre quelle posizionate in modo da essere a mutue distanze spaziali inferiori ad un dato raggio di soglia. Tra tutte le aggregazioni così selezionate, ovvero tutti gli spostamenti che abbiano destinazioni molto vicine nel tempo e nello spazio, si verifica la contemporanea esistenza di aggregazioni spaziali entro lo stesso raggio nell'insieme dei punti di partenza. Qualora queste seconde aggregazioni siano non nulle, sarà possibile creare una tratta di ride sharing allocandovi gli utenti a multipli dei posti disponibili sui veicoli utilizzati, dalla zona dove giacciono i punti di partenza a quella dove sono quelli di arrivo.

Nelle aggregazioni sulle partenze il vincolo sull'orario è rilassato in quanto si suppone che sia di interesse arrivare in un dato luogo ad un certo orario e che si sia disposti a cambiare il proprio orario di partenza in modo opportuno per soddisfare tale vincolo sull'arrivo.

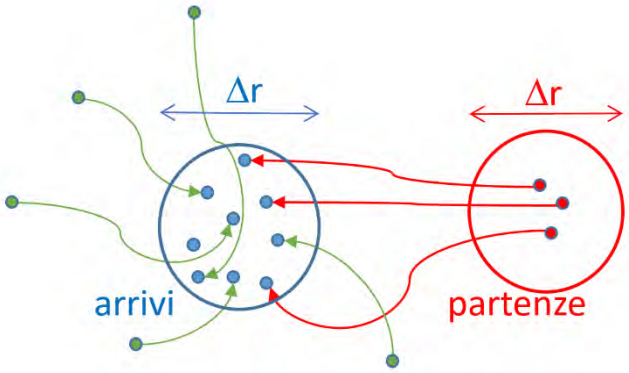


Figura 3. Data una finestra temporale, tra tutti gli arrivi entro un dato raggio, si aggregano quelli partenti da posizioni vicine

In Figura 3 è mostrato graficamente il concetto: data una finestra temporale, tra tutti gli arrivi aggregati entro un diametro Δr, si scelgono quelli che abbiano le posizioni di partenza entro un dato diametro Δr. Le due dimensioni possono, naturalmente, essere anche diverse. Facendo riferimento alla Figura 3, sarà possibile sostituire alle tre partenze un veicolo di ride sharing con almeno tre posti a bordo, nell'ipotesi di un solo passeggero per veicolo nelle tratte originali esaminate.

2.3 Analisi delle caratteristiche di algoritmi di clustering

Come esposto nel precedente paragrafo, una volta scelto l'orario e la finestra temporale, è necessario clusterizzare spazialmente i punti di arrivo prima e quelli di partenza dopo. Esistono numerose tecniche di clustering. Sono stati qui indagati due algoritmi molto noti in letteratura e, soprattutto, dotati di efficienti implementazioni in librerie di calcolo: DBSCAN e K-means.

Il DBSCAN (Density-Based Spatial Clustering of Applications with Noise) è un metodo di clustering proposto nel 1996 da Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu [7]. È basato sulla densità in quanto connette regioni di punti con densità sufficientemente alta. DBSCAN è un algoritmo tra i più usati ed è anche il più citato nella letteratura scientifica.

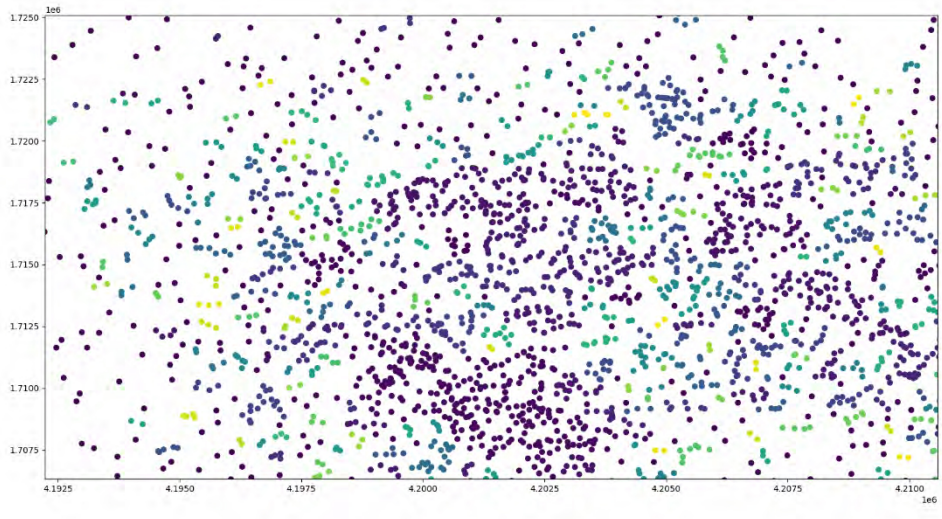


Figura 4. Esempio di risultati con DBSCAN

In nuce DBSCAN stima la densità attorno a ciascun punto contando il numero di punti in un intorno specificato dall'utente, ed applica delle soglie numeriche per identificare i punti "core", "border" e "noise".

In un secondo passaggio, i punti core sono riuniti in un cluster, se sono “raggiungibili per densità”, cioè se esiste una catena di punti core in cui ogni punto ricade all’interno dell’intorno del successivo. Infine i punti di bordo sono assegnati ai cluster. L’algoritmo richiede solo due parametri: le dimensioni dell’intorno ed il numero di punti di tipo core.

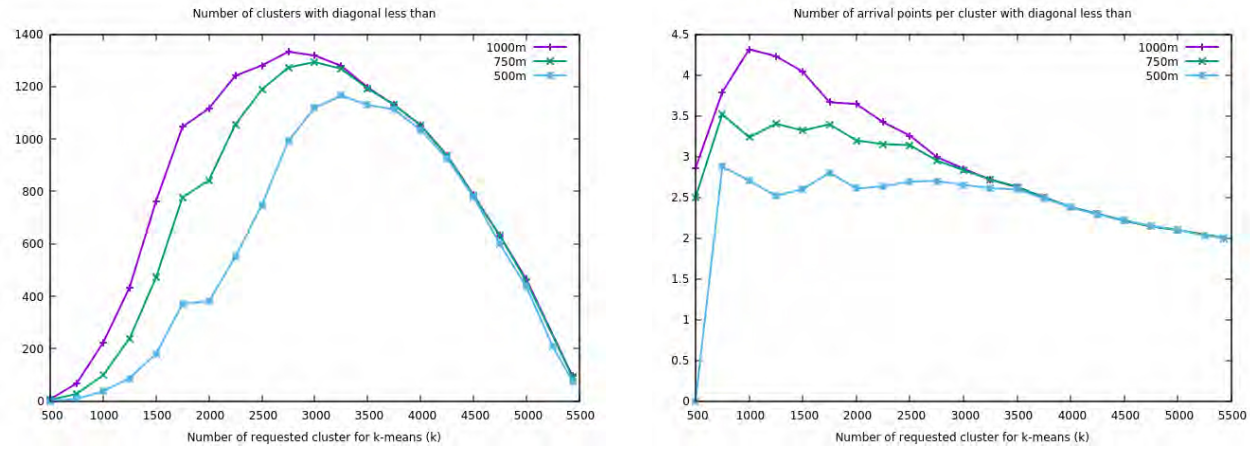
In Figura 4 è mostrato un esempio di clustering sulle posizioni di arrivo utilizzando DBSCAN. Appare abbastanza evidente che l’algoritmo clusterizza i dati per densità, ma, per gli interessi della presente applicazione, mostra una certa facilità a mettere nello stesso cluster punti che risultano essere con posizione relativa più grande delle dimensioni cercate. Questo perché esiste un terzo punto che è a distanza minore da entrambi i punti considerati e che li ‘mette in connessione’, creando delle collane di punti.

E’ stato dunque esaminato l’algoritmo K-means [8]. Esso è un algoritmo di analisi dei gruppi partizionale che permette di suddividere un insieme di oggetti in k gruppi sulla base dei loro attributi. L'obiettivo che l'algoritmo si propone è di minimizzare la varianza totale intra-gruppo; ogni gruppo viene identificato mediante un centroide. L'algoritmo segue una procedura iterativa: inizialmente crea k partizioni e assegna casualmente i punti d’input a ogni partizione; quindi calcola il centroide di ogni gruppo; costruisce in seguito una nuova partizione associando ogni punto di input al gruppo il cui centroide è più vicino ad esso; infine vengono ricalcolati i centroidi per i nuovi gruppi e così via, minimizzando la varianza, finché l'algoritmo non converga.

Il principale problema è legato alla necessità di conoscere a priori il numero k dei cluster da ottenere. E’ possibile utilizzare delle euristiche per calcolare questo numero k, che però prevedono la ripetizione dell’algoritmo per tutti i possibili k allo scopo di poter scegliere graficamente il valore più idoneo.

In Figura 5 a titolo di esempio sono mostrati alcuni risultati ottenuti per i dati relativi alle posizioni di arrivo in una finestra di 15 minuti intorno alle ore 10:00. A sinistra (Figura 5a) è mostrato il numero di cluster minori del raggio desiderato in funzione del numero k di cluster input dell’algoritmo. È evidente come all’aumentare del raggio considerato, aumenti anche il numero di cluster trovati; allo stesso tempo il grafico ha una configurazione a campana: se sono chiesti pochi cluster all’algoritmo (k piccolo) si troveranno pochi cluster mentre per k grandi si avrà una enorme frammentarietà e quindi meno cluster di interesse.

Nella Figura 5b, a destra, è invece mostrato il numero medio di punti per cluster. Anche in questo caso più è ampio il diametro del cluster, maggiore è il numero di punti di arrivo, ma in funzione di k il numero di punti per cluster tende a diminuire molto lentamente.



(a) (b)

Figura 5. Risultati con il clustering K-means

Il principale difetto dell'algoritmo k-means è la necessità di conoscere a priori il numero di cluster in cui si desidera raggruppare i dati. Ciò rende poco naturale una strategia quale quella che si vuole implementare, infatti, per gli scopi della presente attività, l'interesse è incentrato sulla vicinanza geografica tra punti. Si è quindi approcciato il problema dal punto di vista geometrico.

2.4 La tassellatura dello spazio

Lo scopo di questa attività è quella di raggruppare i dati geografici di arrivo prima e di partenza poi, secondo una certa dimensione spaziale. Oltre ad un approccio di clustering, come è stato descritto nei paragrafi precedenti è anche possibile utilizzare una tassellatura direttamente fornita da una discretizzazione dello spazio in quadrati, si veda la Figura 6.

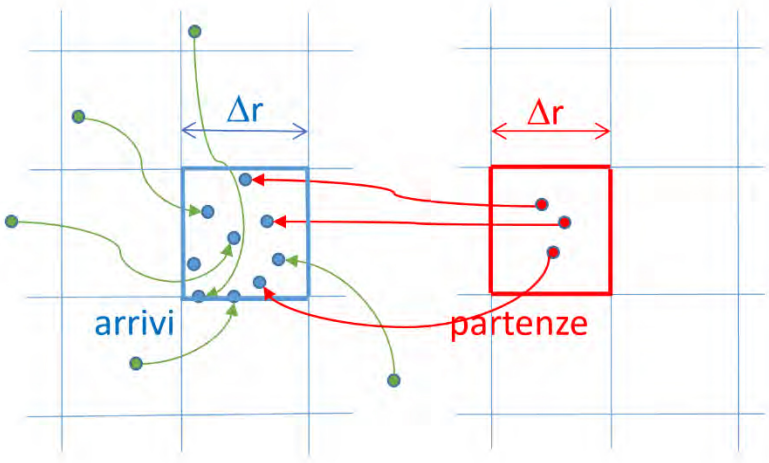


Figura 6. Aggregazione per tiling

A differenza degli algoritmi di clustering, in questo caso si compie una semplice discretizzazione dello spazio in quadrati di un dato lato, che misurerà la distanza che un utente è disposto a coprire per poter usufruire del servizio di ride sharing. La misura di questo lato è, quindi, uno dei parametri del sistema complessivo. Nel prosieguo questi quadrati saranno indicati con il nome di celle.

Naturalmente è possibile che accada che qualche punto di arrivo (o di partenza) venga ignorato perché, pur vicino, ricade nella cella adiacente, ma è stato deciso di trascurare questo problema a fronte della possibilità di ottenere un'aggregazione computazionalmente molto efficiente.

2.5 Simulazioni

A valle di queste analisi propedeutiche si è proceduto alla simulazione del ride sharing urbano. Queste simulazioni possono essere logicamente divise in due approcci: uno statico ed uno dinamico. Con il termine statico si intende l'analisi del numero di viaggi che possono essere effettuati in ride sharing per singole finestre temporali dovuti alle richieste di traffico giornaliere. Ovvero: data una certa matrice origine destinazione (OD) ad una data ora del giorno, quanti viaggi che arrivano in una certa area sono originati in posizioni vicine tra loro e che quindi possano essere condivisi?

L'approccio dinamico invece prevede una simulazione dinamica delle auto in ride sharing, che vengono effettivamente mosse nella città e che possono essere quindi riutilizzate quando nuovamente libere nelle celle di arrivo precedentemente raggiunte nella finestra temporale di arrivo.

2.5.1 Approccio statico

Come detto, l'approccio statico prevede l'analisi della condivisibilità di viaggi in funzione dell'ora della giornata, in altre parole è il numero di viaggi di ride sharing che si potrebbero effettuare nell'ipotesi che fossero disponibili un numero sufficiente di veicoli in tutte le celle in cui è suddivisa la città.

La procedura prevede la scansione di tutte le celle alla ricerca di punti di arrivo delle traiettorie. Nell'insieme delle traiettorie che terminano in una data cella si ricerca poi l'esistenza di viaggi che originino in una medesima cella. Qualora questo secondo insieme fosse non vuoto si procede ad 'accorpare' un dato numero *n* di traiettorie in una sola, ipotizzando di riempire completamente un veicolo di ride sharing con *n* passeggeri. L'accorpamento viene ripetuto fino a che siano presenti passeggeri.

I parametri di ingresso del codice sono rappresentati dalla dimensione del lato della cella in metri e dall'intervallo temporale entro il quale si considerano condivisibili i veicoli (15 o 30 minuti) e, naturalmente, delle traiettorie generate dal modello di richiesta di traffico come rielaborate secondo quanto esposto nel paragrafo 2.1.

```
finestra temporale: dt
lato della cella: dx
dati: traiettorie simulate

generazione della griglia (dx)
for t nelle 24h a passi dt:
        selezione dei dati rilevanti(t)
        eliminazione traiettorie non corrette
        eliminazione celle vuote

        for i nelle celle di arrivo non vuote:
            cerco le relative celle di partenza consultando le traiettorie
            se esistono celle di partenza uguali (cioè esistono più viaggi)
                aggregazione dei più viaggi

        calcolo passeggeri in sharing e salvataggio log
        aggiunta ad un file complessivo di output
calcolo quantità globali
salvataggio dati (V, O, D, T, K)
```

Figura 7. Lo pseudo codice dell'algoritmo di calcolo dei viaggi condivisibili

I passi logici del codice sono mostrati in Figura 7.

L'algoritmo seleziona le celle di arrivo in una finestra temporale sotto l'ipotesi ragionevole che i viaggiatori che arrivino in una data cella ad una certa ora siano interessati ad arrivare lì esattamente a quell'ora, quindi ricerca le celle origine e se una cella origine è presente più di una volta, considera come condivisibili questi viaggi.

Entrando in maggiore dettaglio, l'algoritmo una volta accettati i parametri di ingresso, crea la griglia di celle. Il loop principale è sul tempo nelle 24 ore a passi della finestra temporale. Per ogni passo si selezionano i dati relativi, ovvero i viaggi con orario di arrivo in quella finestra temporale. Si eliminano i viaggi errati perché con orario di arrivo precedente a quello di partenza, cosa che può avvenire data la natura stocastica della generazione di richiesta di traffico con l'approccio utilizzato; si eliminano le celle vuote, ovvero che non siano né origine né destinazione di viaggi, per alleggerire il calcolo.

Il secondo loop, annidato nel primo, scandisce le celle di arrivo non vuote e cerca le relative celle di partenza che, se presenti più di una volta, sono aggregate. Si contano infine quanti passeggeri possono così essere aggregati e si salvano i risultati dei calcoli. Al termine del loop più interno si ricominciano i calcoli per la finestra temporale successiva. Viene così ottenuta una matrice per ogni passo di simulazione (ad es. 48 matrici se il passo di simulazione è 30 min.). Queste matrici contengono, per ogni coppia di origine destinazione, il numero di viaggi che partono da una zona ad un orario qualsiasi e arrivano nella zona di destinazione in uno stesso intervallo temporale, il tempo di percorrenza e la distanza medi di questi viaggi.

La matrice OD così generata è relativa agli orari di arrivo (es. quante macchine a una data ora dalla cella i arrivano nella cella j). Viene quindi calcolata una nuova matrice OD relativa all'orario di partenza

assumendo che ogni viaggio da una generica zona i a una zona j impieghi un tempo pari al tempo medio necessario per coprire questa distanza calcolato al passo precedente.

Al termine si calcolano delle quantità globali quali ad esempio il numero di viaggi risparmiati e si salvano su disco i risultati.

Questa procedura può essere interpretata come calcolo delle matrici Origine Destinazione (OD) per ogni destinazione e per ogni finestra temporale e contando quanti gruppi da n siano presenti in ogni elemento.

A titolo di esempio In Figura 8 è mostrato il profilo orario dei viaggi di ride sharing ottenibili nell'ipotesi di un traffico composto di veicoli occupati da un solo individuo. I dati sono relativi ad un numero totale di viaggi settimanali prodotto dal modello, pari a poco meno di due milioni e mezzo (2458215) e sono quelli di un giorno feriale, in particolare un mercoledì con 356115 viaggi.

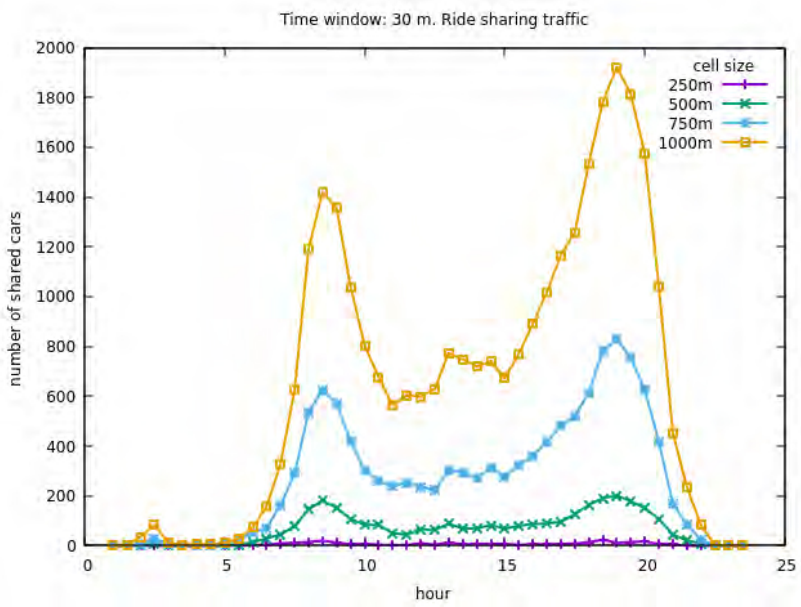


Figura 8. Il traffico shared in funzione dell'orario (finestra temporale di 30 minuti)

Nella Figura 8 i tracciati di diverso colore sono relativi a diverse dimensioni delle celle di 250, 500, 750 e 1000 metri, mentre la finestra temporale scelta è di 30 minuti.

Nel corso di questa analisi e delle successive, non sono stati considerati i viaggi che abbiano origine e termine nella medesima cella, sotto l'assunzione che un viaggio all'interno della medesima cella non sia significativo.

Appare evidente che il numero di viaggi condivisi cresce con la dimensione delle celle e che presenta due picchi relativi ai principali flussi di spostamento urbano: da e per l'attività lavorativa.

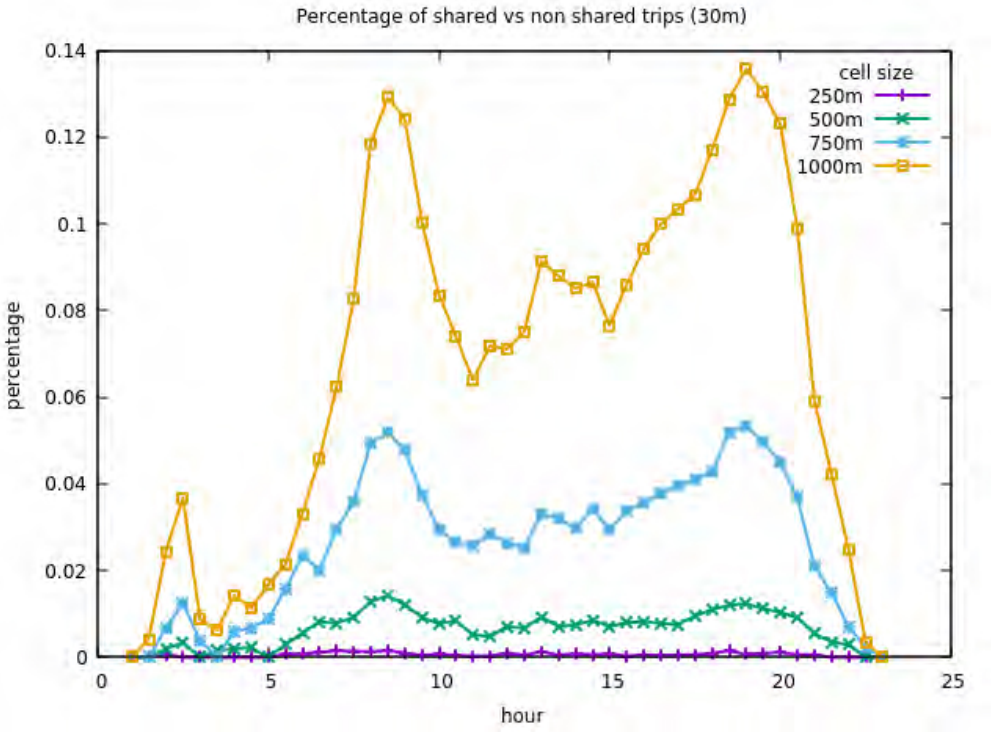


Figura 9. Percentuale di viaggi shared rispetto alla totalità dei viaggi

In Figura 9 è mostrata la stessa informazione della figura precedente, ma in termini di percentuale di viaggi condivisi rispetto alla totalità dei viaggi.

In Figura 10 è invece riportata la percentuale di celle 'attive', ovvero quelle celle che sono interessate dal contenere punti di partenza o di arrivo per i viaggi.

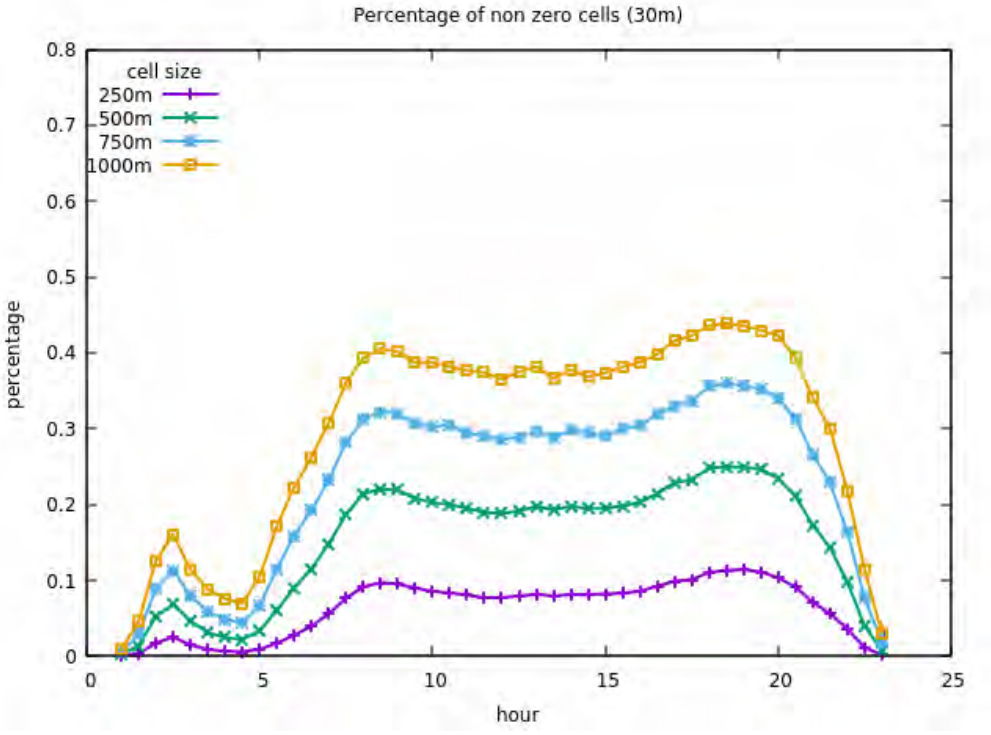


Figura 10. Percentuale di celle con arrivi o partenze, rispetto all'intera tassellizzazione

E' interessante notare come la distribuzione spaziale delle celle attive appaia fondamentalmente costante tra le 8:00 e le 20:00, indicando una situazione di traffico condivisibile abbastanza uniformemente distribuita.

Il prodotto finale di questa attività è rappresentato da una matrice OD sparsa che integra la serie temporale di matrici OD che rappresentano le necessità di ride sharing tra le varie celle della città in funzione dell'ora del giorno e complete di informazioni accessorie quali il tempo di viaggio, la distanza coperta ed il numero di passeggeri trasportabili.

2.5.2 Approccio dinamico

Le analisi presentate nel precedente paragrafo sono di carattere statico e mostrano il modello di richiesta di ride sharing durante la giornata al variare dell'orario in un dato giorno della settimana. E' stato quindi implementato un modello di simulazione dinamico che utilizza i dati calcolati nel modello statico per poter studiare la distribuzione più opportuna dei veicoli di ride sharing nella città. Il modello si focalizza sui soli veicoli di ride sharing, simulandone i percorsi giornalieri e mostra come questi evolvano durante l'operatività quotidiana. In altre parole il modello statico fornisce le istruzioni per far evolvere il modello dinamico.

Il punto di partenza è rappresentato da un popolamento delle celle cittadine con un dato numero di veicoli di ride sharing distribuiti con una qualche funzione di distribuzione. Per fissare le idee si pensi ad una distribuzione uniforme su tutte le celle di un numero N di veicoli. Al passare del tempo, sulla base delle richieste di viaggi condivisi calcolati nel precedente paragrafo, i veicoli di ride sharing sono mossi, aggiornandone la posizione e la disponibilità. Si segue poi l'evoluzione giornaliera, assumendo che i veicoli a fine servizio, possano autonomamente ritornare nella posizione iniziale, per poter affrontare una nuova giornata di servizio.

```
carico griglia celle
carico la matrice OD sparsa (mODs) dei viaggi condivisibili
cerco max x e y in mODs per centrare la mappa
creo le posizioni iniziali dei veicoli con funzione di
   distribuzione data e tempo uguale a 0

NOTA:  Cars(position, time)

for t nelle 24 ore a passi dt:
      for Car in Cars:
            if Car.time <= t:  (l'auto e' disponibile in questo timestep?)
                  seleziono in mODs(t) i viaggi che originano da Car.position al tempo t
                  if esistono viaggi:
                        scelgo opportunamente la destinazione del viaggio shared (orig-dest) [ArgMax]
                        rimuovo da mODs(t) i viaggi appena aggregati [in persone] (orig-dest)
                        calcolo il tempo di arrivo e lo scrivo in Car.time
                        salvo in un log file i dati del viaggio
```

Figura 11. L'algoritmo di evoluzione dinamica dello scenario in pseudo codice

In Figura 11 è mostrato l'algoritmo di evoluzione dinamica dello scenario. L'inizializzazione è rappresentata dal caricamento della griglia di celle, della matrice OD sparsa delle richieste di ride sharing e l'individuazione del massimo per x e y allo scopo di trovare la corretta posizione dei dati. Questo perché le celle in cui è tassellata la mappa non sono tutte attive, ovvero non tutte sono origine e/o destinazione di viaggi, inoltre, al variare dei parametri dell'approccio statico, le celle attive possono cambiare.

Il passo successivo è quello di inizializzare le posizioni dei veicoli di ride sharing. Per poter distribuire i veicoli secondo distribuzioni qualunque si è fatto uso del cosiddetto Rejection Sampling che verrà brevemente illustrato più avanti.

Ogni veicolo è descritto da due campi: la propria posizione ed un tempo, quello in cui sarà considerato libero di essere usato per il ride sharing. L'idea è la seguente: il veicolo *Car(pos_iniz,0)* al tempo zero potrà essere usato e sarà utilizzato per compiere un viaggio di ride sharing da *pos_iniz* a *pos_fin*. Per compiere questo tragitto il veicolo impiegherà ad esempio 7 passi temporali, l'algoritmo allora aggiornerà i dati del veicolo con *Car(pos_fin,7)*. L'algoritmo confronta il proprio tempo con quello del veicolo in questione e non potrà riutilizzarlo prima del passo temporale 7, quando si troverà in *pos_fin*.

L'algoritmo ha un loop esterno sul tempo nelle 24 ore a passi della finestra temporale al cui interno è presente un loop sui veicoli di ride sharing che vengono utilizzati solo se sono liberi, con il meccanismo appena descritto. I veicoli sono utilizzati sulla base della matrice OD di quella finestra temporale aggregando, ove possibile, i viaggi con uguale cella di origine e destinazione. Qualora esistano, si aggregano i viaggiatori fino a un massimo di tre contemporaneamente, viene aggiornata la posizione, la disponibilità del veicolo che ha compiuto il viaggio di ride sharing e gli elementi della matrice sparsa OD nei campi della numerosità dei viaggi da compiere, sottraendo il numero di passeggeri già "serviti". Vengono anche salvati i singoli viaggi compiuti con i dati rilevanti: origine, destinazione, passeggeri trasportati, distanza e tempo di viaggio.

Mano a mano che lo scenario evolve i veicoli di ride sharing si muovono tra le celle e diventano via via disponibili in altre zone.

2.5.3 Il Rejection Sampling

Il Rejection Sampling è una tecnica statistica per estrarre numeri casuali secondo una distribuzione *difficile*. L'idea di base è che sebbene non si possa campionare facilmente la distribuzione di interesse, esiste un'altra densità di probabilità da cui è facile campionare, classico esempio la distribuzione uniforme. Quindi è possibile campionare da questa seconda densità direttamente e poi "rifiutare" i campioni in modo opportuno per far sì che i campioni "accettati" risultanti provengano proprio dalla funzione di interesse.

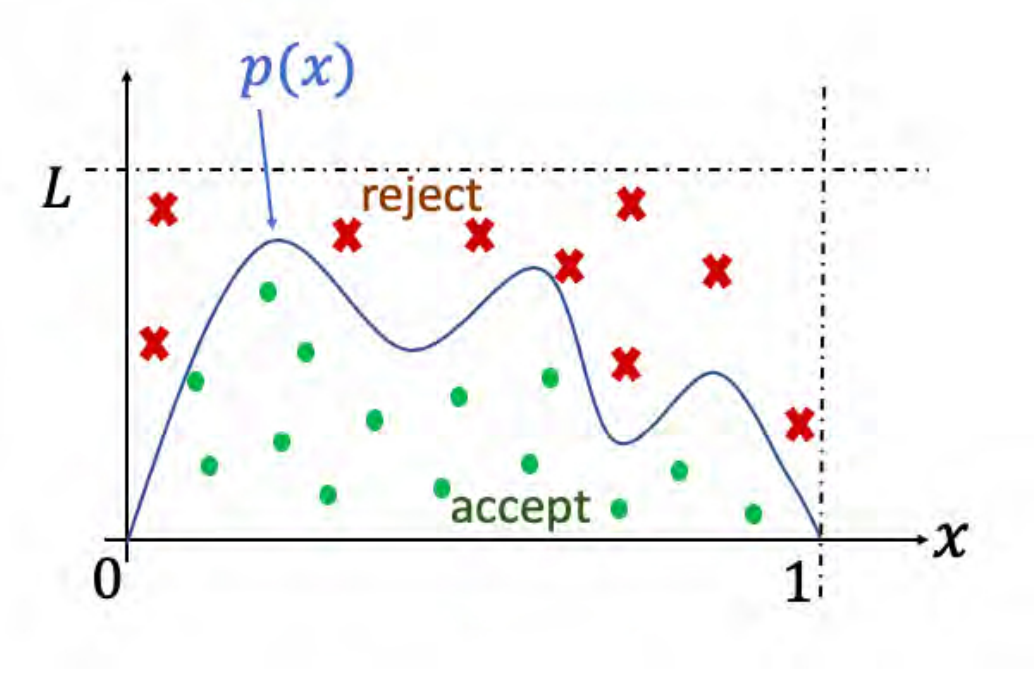


Figura 12. L'idea del Rejection Sampling

In Figura 12 è riassunta l'idea. Si campionano due numeri casuali x_1 tra 0 e 1 e y_1 tra 0 e L, se il punto (x_1, y_1) risulta sotto la distribuzione di probabilità p(x) desiderata, esso viene accettato, altrimenti viene rifiutato. Ripetendo questo campionamento moltissime volte i punti accettati saranno via via distribuiti come la funzione p(x) di interesse.

Nel presente caso, in cui si voglia ad esempio scegliere una distribuzione per i veicoli di ride sharing uguale alla richiesta di viaggi delle ore 8:00, è sufficiente estrarre un primo numero tra 0 ed il numero totale delle celle come x e un numero uniformemente distribuito tra 0 ed il numero massimo di viaggi in partenza da una qualunque cella come y ed operare il rejection sampling.

2.5.4 Risultati della evoluzione dinamica dello scenario

I parametri delle simulazioni sono rappresentati da quattro diverse grandezze:

- la dimensione della cella;
- la dimensione dello step temporale;
- il numero dei veicoli;
- la distribuzione iniziale dei veicoli.

Sono stati compiuti diversi esperimenti con diversi valori e diverse distribuzioni iniziali di veicoli nelle celle. In particolare sono stati studiati i casi di distribuzione uniforme e di distribuzione simile alle richieste di ride sharing, come calcolate nel paragrafo 2.5.1, delle 8:00 e delle 9:00, si veda la Figura 13. I dati qui presentati rappresentano le analisi relative ad un mercoledì non festivo.

Qui di seguito sono esposti i risultati ottenuti utilizzando come distribuzione iniziale quella delle richieste delle ore 8:00, dimensione della cella pari a 1000 m e dimensioni del time step 30 minuti. Non sono qui riportati i risultati ottenuti distribuendo i veicoli in modo uniforme in quanto nettamente peggiori rispetto ad una distribuzione iniziale che tenga conto delle richieste di ride sharing di una data ora. Di seguito si è utilizzata la distribuzione alle ore 8:00, in quanto i risultati utilizzando questa sono simili ma migliori rispetto, ad esempio, ad utilizzare quella delle ore 9:00.

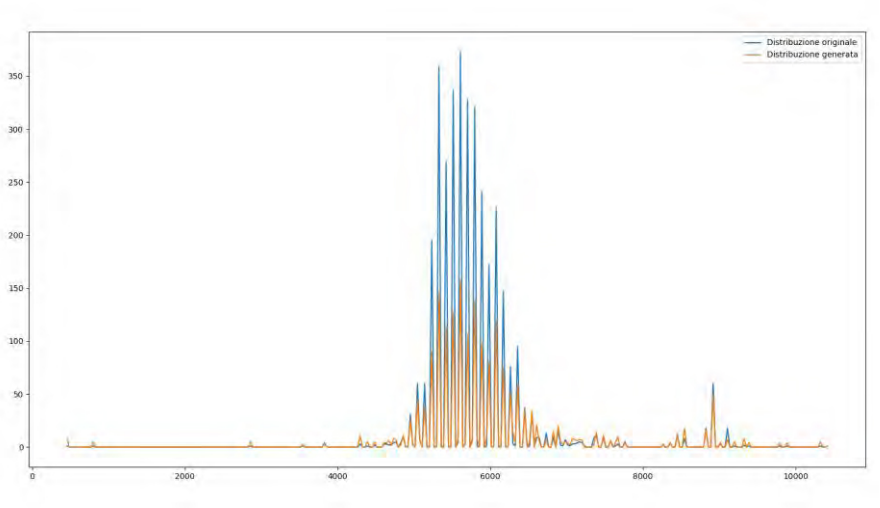


Figura 13. La distribuzione iniziale nelle varie celle secondo la richiesta di ride sharing delle ore 8:00. In blu la distribuzione originale, in arancione la distribuzione di 1000 veicoli generata con il rejection sampling

La prima quantità esaminata è il numero di veicoli di ride sharing utilizzati in funzione dell'ora del giorno, a parità degli altri tre parametri. In Figura 14 è mostrata questa quantità per i tre scenari: 500, 1000 o 2000 veicoli.

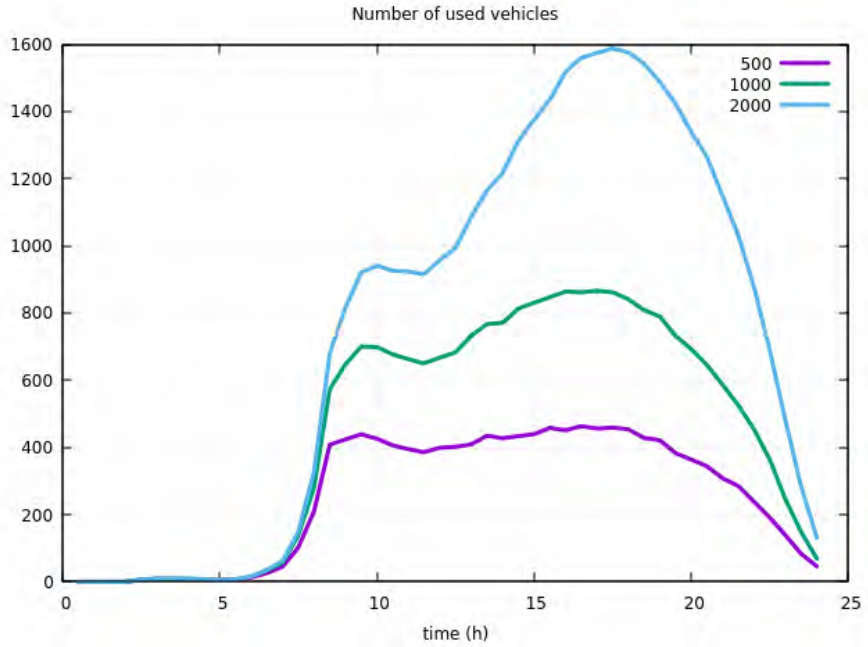


Figura 14. Il numero di veicoli di ride sharing in uso alle varie ore del giorno

E' possibile osservare che la frazione di veicoli utilizzati è sempre abbastanza alta: con 500 veicoli abbiamo un picco di utilizzo al 93%, per 1000 si ha un 87% di veicoli usati e per 2000 l'80%. Con un numero basso si può notare che i veicoli di ride sharing sono utilizzati con un'alta percentuale durante l'intera giornata. Nel caso di distribuzione uniforme su tutte le celle le percentuali di utilizzo crollano a valori intorno al 9-10%.

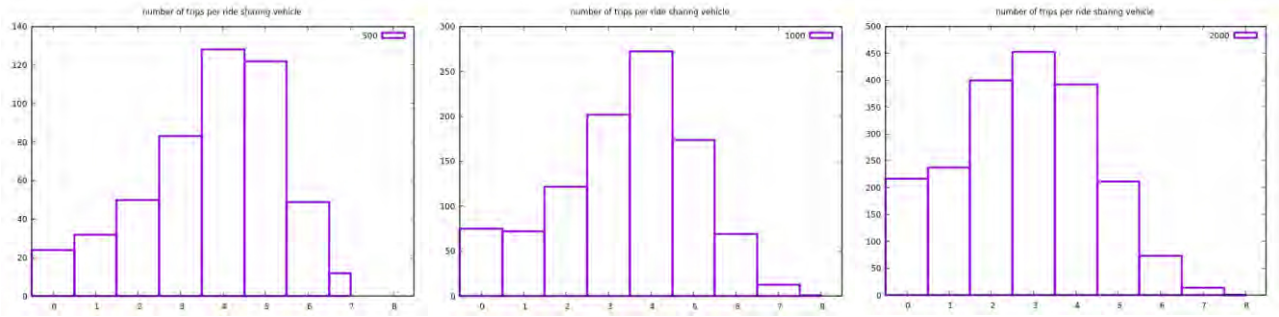


Figura 15. Istogrammi del numero di viaggi per veicolo nei casi di 500, 1000 e 2000 veicoli di ride sharing

In Figura 15 è mostrato invece l'istogramma del numero di viaggi realizzati da ciascun veicolo nei tre casi di numerosità di auto. In tutti e tre gli scenari le auto in ride sharing vengono utilizzate mediamente 4 volte nell'arco della giornata. Il numero di veicoli che non vengono utilizzati affatto rappresenta rispettivamente il 5%, il 7.5% e l'11% del relativo numero di veicoli di ride sharing.

Osservando la Figura 15, appare evidente che con un minor numero di veicoli si ha una distribuzione più piccata verso un numero alto di viaggi e quindi una maggiore efficienza relativa, mentre con parchi auto più grandi si ha una minore efficienza relativa in quanto il picco si sposta verso sinistra, ma il numero assoluto di viaggi effettuati sarà ovviamente maggiore.

Un indicatore importante è rappresentato dai *km x veicolo* risparmiati. Per calcolare questa quantità si sono moltiplicati i km del singolo viaggio per il numero di passeggeri trasportati in quel viaggio meno uno. L'assunzione è che si vengano a sostituire *n* veicoli, uno per passeggero, con un solo veicolo di ride sharing e che quindi si possano risparmiare (*n-1*) *km x veicolo*. In queste simulazioni si è anche assunto un numero massimo di passeggeri per veicolo pari a tre. In Tabella 1Tabella 1 sono riassunti i *km x veicolo* risparmiati nell'arco dell'intera giornata, sia assoluti che relativi alla numerosità del parco veicoli di ride sharing. E'

anche in questo caso evidente come all'aumentare del parco auto aumenti il numero di *km x veicolo* risparmiati, ma con efficienza sempre minore.

Tabella 1. I km x veicolo risparmiati, valore assoluto e relativo

Numero veicoli	Km x veicolo risparmiati	Km x veicolo risparmiati, media sui veicoli
500	16502	33.0
1000	29860	29.9
2000	48534	24.3

Altro descrittore è rappresentato dal numero di viaggi con due o con tre passeggeri, questo è riassunto in Tabella 2. In parentesi è la media riferita alla numerosità di veicoli di ride sharing.

Tabella 2. Il numero di viaggi in due o tre passeggeri

Numero veicoli	Viaggi in 2	Viaggi in 3
500	1470 (2.94)	450 (0.90)
1000	2625 (2.63)	803 (0.80)
2000	4369 (2.18)	1280 (0.64)

In Figura 16 sono mostrati gli istogrammi relativi al numero di passeggeri trasportati da macchine di ride sharing in più viaggi. Questa, insieme alla Figura 15, fornisce una descrizione più puntuale dell'effettiva condivisione dei veicoli.

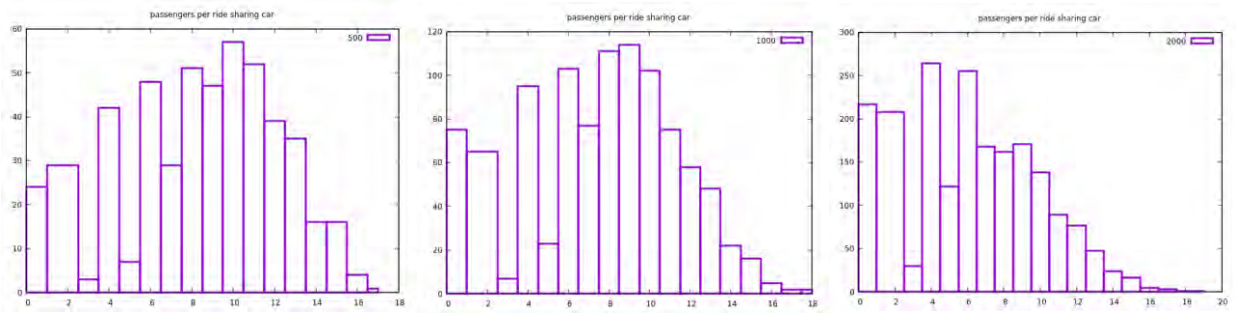


Figura 16. Gli istogrammi del numero di passeggeri trasportati in più viaggi dai veicoli di ride sharing in un giorno

2.5.5 Discussione sulla evoluzione dinamica dello scenario

Quanto mostrato indica la possibilità di simulare scenari di ride sharing per poter poi misurarne a posteriori l'efficacia attraverso una serie di variabili. E' possibile variare diversi parametri quali le dimensioni della cella e lo step temporale, il numero di veicoli, ma, soprattutto, la distribuzione spaziale nella città dei veicoli di ride sharing.

E' chiaro che quest'approccio simulativo rappresenta una prima approssimazione all'analisi di strategie di ride sharing e che scenari più raffinati possono e devono essere messi in campo, quale quello, ad esempio, di prevedere la possibilità di far salire e scendere passeggeri durante il tragitto del veicolo.

Ciò nonostante l'approccio presentato permette comunque di realizzare una analisi in termini di efficienza, giocando con i parametri del modello.

E’ evidente dai risultati che per avere un ride sharing significativo è necessario che gli utenti siano disposti ad utilizzare celle di dimensioni 1 km e siano altresì disposti a sopportare tempi dell’ordine della mezz’ora. Naturalmente questi valori sono anche funzione del numero di viaggi creati con il modello di richiesta di traffico, che, in queste simulazioni, genera un flusso di veicoli pari a quello presente nel dataset originale che rappresenta circa il 7% del parco auto circolante nell’area. E’ intuitivo che se si ripetono le analisi con una richiesta di traffico maggiore, si avranno risultati migliori.

Una distribuzione iniziale opportuna per il parco auto di ride sharing è assolutamente basilare per ottenere una sufficiente efficienza per il sistema. La distribuzione iniziale che ricalchi la richiesta di traffico delle ore 8:00 è quella che permette risultati migliori, se viene usata quella relativa alle 9:00 i risultati sono inferiori, ma simili, mentre con una distribuzione uniforme l’efficienza crolla.

Il numero di veicoli di ride sharing messi in campo è un altro parametro ovviamente basilare. Dalle simulazioni è possibile osservare che l’efficienza relativa del ride sharing diminuisce all’aumentare dei veicoli disponibili, anche se in termini assoluti il numero di viaggi e di *km x veicolo* aumenta con i veicoli. Sta poi al decisore sul territorio scegliere la strategia che meglio si adatta a queste caratteristiche, tenendo conto dei vincoli soprattutto economici, qui non considerati.

2.6 Interazione ed interfacciamento con l’interfaccia grafica

Durante la presente annualità è stata portata avanti un’altra attività concernente l’interfacciamento dei modelli sviluppati con l’interfaccia grafica sviluppata dalla Ilab s.r.l., proseguendo l’attività già iniziata nella scorsa annualità.

I modelli sviluppati che devono essere integrati con il sistema sviluppato da Ilab sono due:

- il modello di richiesta di domanda di traffico, sviluppato nella seconda annualità del Piano Triennale [2], in grado di generare traffico veicolare virtuale ed anonimizzato;
- il modello di ride sharing descritto nei precedenti paragrafi.

Il primo modello genera il traffico virtuale della giornata utilizzando in input alcuni parametri:

- il numero di viaggi da generare;
- i giorni della settimana per i quali generare i viaggi;
- i limiti geografici entro cui generare i viaggi.

Per inserire tali parametri, l’utente si deve collegare tramite un browser all’interfaccia di EMU ed accedere alla maschera per la creazione di un nuovo scenario, vedi Figura 17. Una volta creato lo scenario, i parametri sono salvati nella tabella *scenario_traffic* del database di funzionamento della interfaccia di EMU.

Il modulo di generazione delle traiettorie monitora periodicamente il database e, nel momento in cui trova un nuovo scenario nella citata tabella scenario_traffic, avvia la generazione di nuove traiettorie.

Un record della tabella scenario_traffic è costituito dai seguenti campi:

- idscenario_traffic: l’identificativo dello scenario;
- name: il nome dello scenario;
- day: l’elenco dei giorni per i quali è richiesta la generazione delle traiettorie (Es. “lun-mar-merc-gio-ven-sab-dom”);
- bbox: il riquadro che delimita lo spazio di generazione delle traiettorie (4 punti rappresentati ognuno da latitudine e longitudine);
- do: è un flag che indica se la simulazione è in corso o meno;
- nviaggi: il numero di viaggi da generare.

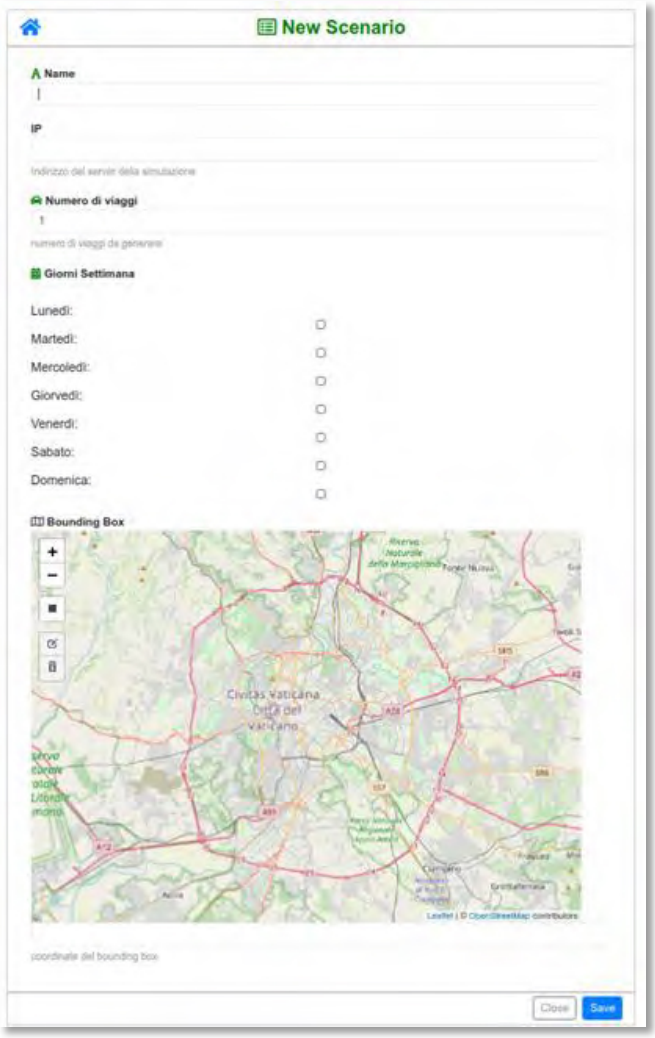


Figura 17. La maschera di inserimento dei parametri per la simulazione

Una volta avviata la generazione delle traiettorie, al suo termine il modulo aggiorna il flag “do” nella tabella scenario_traffic e carica tramite sftp i risultati aggreggati in una cartella sul server.

A valle della sintesi del traffico virtuale relativo ad una giornata con il primo modello, si avranno quale output le seguenti grandezze relative ad ogni ora della giornata e relative alla suddivisione in zone della città tramite esagoni di circa 700 m di diametro:

- count_in: il numero di viaggi che terminano nella zona e nell’ora corrente;
- count_out: il numero di viaggi che originano dalla zona e nell’ora corrente;
- cars_parked: il numero di veicoli in sosta nella zona e nell’ora corrente;
- avg_stop_time: il tempo medio di sosta nella zona e nell’ora corrente.

Questi dati sono contenuti in 24 file, uno per ogni ora della giornata, in formato ‘geojson’ [https://en.wikipedia.org/wiki/GeoJSON]. Il software EMU è poi in grado di visualizzare questi dati, si veda la Figura 18.

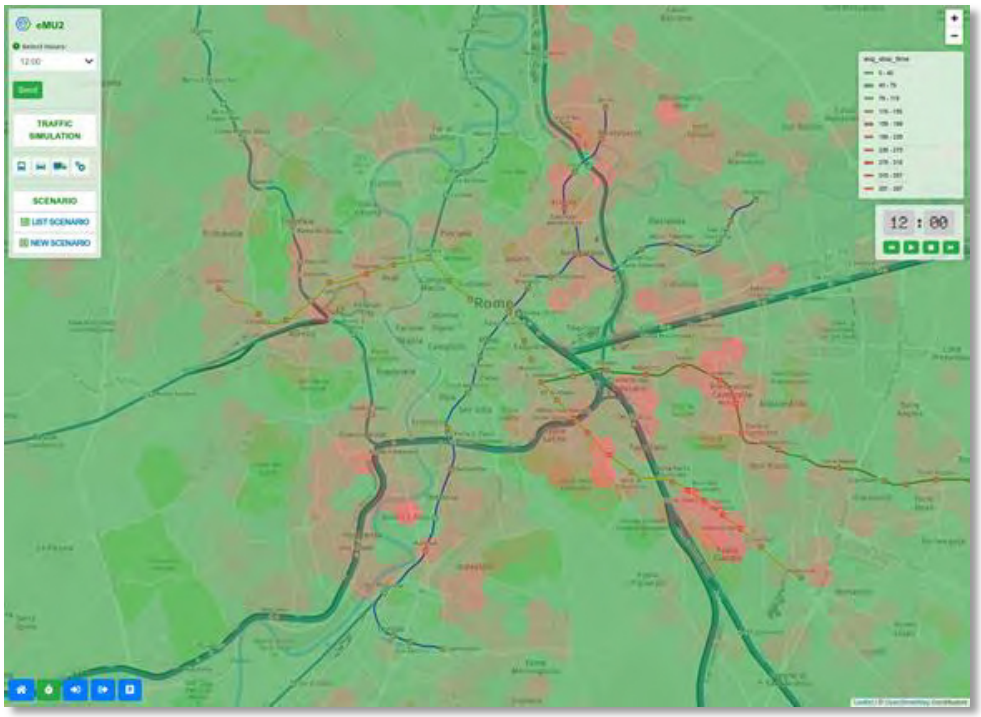


Figura 18. Esempio di visualizzazione dei dati in EMU

Per ciò che attiene al modulo di simulazione del ride sharing, esso ha come parametri di input il numero di auto di ride sharing da utilizzare e la funzione di distribuzione spaziale nella città. Il modulo, sulla base delle matrici OD del traffico generate dal modello di richiesta di traffico, simula gli spostamenti dei veicoli di ride sharing ed analogamente al caso precedente carica sul server i relativi file di output, sempre in formato geojson. Questi file descrivono le caratteristiche di ciascuna cella (esagono) tramite i seguenti campi:

- il numero di macchine di ride sharing disponibili in quella cella;
- la distanza percorsa dalle macchine di ride sharing;
- il numero di passeggeri trasportati;
- il numero di auto di ride sharing inutilizzate.

3 Conclusioni

Nel corso di questa terza annualità le attività si sono concentrate su due filoni di lavoro: da una parte si sono utilizzati i prodotti dei precedenti periodi per generare delle simulazioni di ride sharing, dall'altra parte si sono approntati i mezzi per interfacciare il modulo di generazione di traffico e quello di ride sharing con l'interfaccia grafica realizzata in un'altra linea di attività.

Il principale risultato del primo filone è rappresentato dalla realizzazione di un sistema capace di simulare l'uso di veicoli di ride sharing durante il corso della giornata per poter studiare l'influenza dei vari parametri sulla efficacia del servizio. In particolare è possibile misurare le prestazioni del paradigma di ride sharing rispetto alla distribuzione geografica dei veicoli. E' infatti possibile concludere che una distribuzione iniziale di auto che ricalchi quella della richiesta di traffico delle ore del mattino è molto più efficiente di quella di distribuire i veicoli in modo uniforme.

Altra interessante conclusione è quella che vede una efficienza decrescente con il numero di veicoli; in altre parole aumentando il numero di auto di ride sharing aumenterà, ovviamente, il numero di percorsi

effettuati, ma questo aumento sarà minore rispetto a quello dei veicoli. La metodologia di simulazione qui rappresentata può quindi essere utile al decisore sul territorio per la scelta della migliore strategia che metta d'accordo le prestazioni con i vincoli soprattutto economici, qui non considerati.

Nel secondo filone di attività si è fondamentalmente realizzato il sotto sistema di interscambio tra l'interfaccia grafica EMU e i due moduli di generazione del traffico e di ride sharing. Il generatore di traffico consulta il database dell'interfaccia periodicamente per controllare la richiesta di una nuova generazione e fornisce i dati di output tramite file dati. Anche il modulo di ride sharing riceve dall'interfaccia i parametri per la simulazione e fornisce in uscita i dati della simulazione tramite file.

4 Riferimenti bibliografici

1. Sarriera J.M., Álvarez G.E., Blynn K., Alesbury A., Scully T., Zhao J. “To Share or Not to Share: Investigating the Social Aspects of Dynamic Ridesharing”, Transportation Research Record. 2017;2605(1), pp 109-117. doi:10.3141/2605-11

2. Chiesa S., Nanni V., Taraglio S.. “Implementazione e test del modello per la simulazione della mobilità urbana veicolare”, ENEA Technical Report, Report RdS/PTR2020/059, Apr 2021

3. Chiesa S., Taraglio S.. “Traffic modelling through a LSTM variational auto encoder approach: preliminary results”, Proceedings of ICSSA 2021, Cagliari, 13-16 Sept 2021. In: Gervasi O. et al. (eds) Computational Science and Its Applications – ICCSA 2021. Lecture Notes in Computer Science LNCS 12950, Springer, Cham, pp. 598–606, 2021. https://doi.org/10.1007/978-3-030-86960-143

4. Harshvardhan GM, Mahendra Kumar Gourisaria, Manjusha Pandey, Siddharth Swarup Rautaray: A comprehensive survey and analysis of generative models in machine learning. Computer Science Review, Volume 38, 100285, ISSN 1574-0137, https://doi.org/10.1016/j.cosrev.2020.100285, 2020.

5. D.P. Kingma, M. Welling, “Auto-encoding variational Bayes”, International Conference on Learning Representations, ICLR 2014, Apr 14 - 16, 2014, Banff, Canada.

6. Octo Telematics, https://www.octotelematics.com/it/home-it/, ultimo accesso 28/01/2022.

7. Ester, M., Kriegel H.P., Sander J., and Xu X.. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press, pp. 226-231, 1996.

8. MacQueen J.B.. “Some Methods for classification and Analysis of Multivariate Observations”, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1, pp 281-297, 1967.

5 Indice delle figure