Alma Mater Studiorum · Università di Bologna

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea in Fisica

Modelli di traffico per la formazione della congestione su una rete stradale

Relatore:
Prof. Armando Bazzani

Presentata da:
Gregorio Berselli

Anno Accademico 2021/2022

Abstract

Indice

Elenco delle figure

Introduzione

Le congestioni nel traffico sono oggi uno dei maggiori problemi per lo sviluppo delle città. Queste dipendono strettamente dalla logistica della stessa. Considerando una città fittizia, la cui superficie sia approssimabile a quella di una circonferenza di raggio R, è facile osservare come il numero M di nodi della sua rete stradale debba cresecere proporzionalmente alla superficie, ossia $M \propto R^2$. Ci si aspetta inoltre che il costo C della rete sia legato sia al numero di nodi, sia alla dimensione della rete, ottenendo $C \propto ML$. Unendo le relazioni precedenti si dimostra come il costo di una rete stradale sia $C \propto M^{\frac{3}{2}}$, quindi destinato a crescere all'aumentare del numero di nodi. Nonostante la considerazione precedente le città sono comunque riuscite a espandersi nel tempo grazie a cambiamenti nella mobilità, come ad esempio l'introduzione di mezzi pubblici quali autobus, tram e metropolitane.
Tuttavia, gran parte della mobiltà cittadina si presenta ancora come automobili private le quali, soprattutto negli orari di punta, tendono a saturare la rete e comportano enormi sprechi di tempo e denaro, oltre a causare inquinamento.

0.1 Random Walk su network

In generale, un network è descritto da una matrice di adiacenza $\mathcal{A}_{ij} = \{0, 1\}$ in cui la cella (i, j) assume il valore 1 se il nodo i è connesso al nodo j, 0 altrimenti. Nei casi considerati in questo studio si assume sempre che $\mathcal{A}_{ij} = \mathcal{A}_{ji}$ (proprietà di simmetria).
Dalla matrice di adiacenza si può definire il grado del nodo i-esimo come

$$d_i = \sum_j \mathcal{A}_{ij}$$

che indica il numero delle connessioni per ogni nodo.
Una volta note le matrici descritte in precedenza è possibile definire la matrice Laplaciana del network come

$$\mathcal{L}_{ij} = d_i \delta_{ij} - \mathcal{A}_{ij} \tag{1}$$

che ha le seguenti proprietà:

- è semi-definita positiva;

- $\mathcal{L}_{ij} > 0 \iff i = j$;
- $\sum_j \mathcal{L}_{ij} = \sum_i \mathcal{L}_{ij} = 0$, quindi esiste un autovalore nullo λ_0 con corrispondente autovettore $\vec{v}_0 = (1, \ldots, 1)$;
- $\sum_i \mathcal{L}_{ii} = 2m$, dove m è il numero totale dei link.

Si assuma ora che la rete abbia in totale M nodi e che ognuno di essi possa scambiare particelle coi suoi vicini. Sia π_{ij} la matrice stocastica che definisce la probabilità che una particella effettui il viaggio tra nodi $j \to i$. Questa possiede le seguenti proprietà:

- $\mathcal{A}_{ij} = 0 \implies \pi_{ij} = 0$;
- $\sum_j \pi_{ij} = 1$.

Assumendo inoltre di avere N particelle nella rete, è possibile definire la funzione $\delta_\alpha(i,t)$ che vale 1 se la particella α si trova nel nodo i al tempo t, 0 altrimenti.
Ogni particella segue quindi la dinamica

$$\delta_\alpha(i, t + \Delta t) = \sum_j \xi_{ij}^\alpha \delta_\alpha(j,t) \tag{2}$$

dove ξ_{ij}^α è una matrice random che prende valori della base standard $\widehat{e}_i \in \mathbb{R}^M$ con probabilità π_{ij}. Il numero di particelle nel nodo i al tempo t è dato da

$$n_i(t) = \sum_\alpha \delta_\alpha(i,t) \tag{3}$$

ed è possibile dimostrare [1] che la seguente equazione è un integrale del moto

$$\sum_i n_i(t) = N \tag{4}$$

Capitolo 1

Modelli di traffico

1.1 Osservabili macroscopici

Prima di iniziare a modellizzare il problema è necessario domandarsi quali siano gli osservabili macroscopici principali e come siano legati tra di loro. La descrizione di un sistema a livello macroscopico risulta critica ai fini della descrizione della dinamica.

1.1.1 Densità

La densità è una variabile tipicamente fisica adotatta nella teoria del traffico. La densità ρ rappresenta il numero di veicoli per unità di lunghezza della strada. Sia ora Δx la lunghezza di una strada in cui sono presenti n veicoli, allora ad un tempo generico t si ha che

$$\rho(n, t, \Delta x) = \frac{n(t)}{\Delta x}$$

La densità si esprime in veicoli al kilometro (veh/km). Tipicamente per ogni corsia di una strada si ha una $\rho_{max} \sim 10^2$ veh/km. Si osservi ora come moltiplicando e dividendo per un infinitesimo temporale dt il denominatore divenga l'area dell'intervallo di misura S. In particolare

$$\rho(t, \Delta x, S) = \frac{n(t)dt}{\Delta x dt} = \frac{\text{tempo totale trascorso in } S}{S} \tag{1.1}$$

1.1.2 Flusso

Il flusso Φ rappresenta il numero m di veicoli che attraversano un certo località x in un intervallo di tempo Δt

$$\Phi(m, x, \Delta t) = \frac{m}{\Delta t} \tag{1.2}$$

Il flusso è espresso in veicoli all'ora (veh/h). Considerando ora un intorno infinitesimo dx di x è possibile ricavare la dipendenza più generale

$$\Phi(x, \Delta t, S) = \frac{mdx}{\Delta t dx} = \frac{\text{distanza totale percorsa dai veicoli in } S}{S} \tag{1.3}$$

1.1.3 Velocità media

La velocità media è definita come il rapporto tra il flusso e la densità: si nota immediatamente come questa non dipenda dall'area dell'intervallo di misura. Unendo le Eq. (1.3) e (1.1):

$$\bar{v}(x, t, S) = \frac{\Phi(x, \Delta t, S)}{\rho(t, \Delta x, S)} \tag{1.4}$$

La relazione fondamentale della Traffic Flow Theory [3] è riassumibile nella:

$$\Phi = \rho \bar{v} \tag{1.5}$$

1.2 Diagrammi fondamentali

A causa della relazione fondamentale del traffico riportata in Eq. (1.5) risulta chiaro come su tre osservabili analizzati si abbiano solamente due variabili indipendenti.

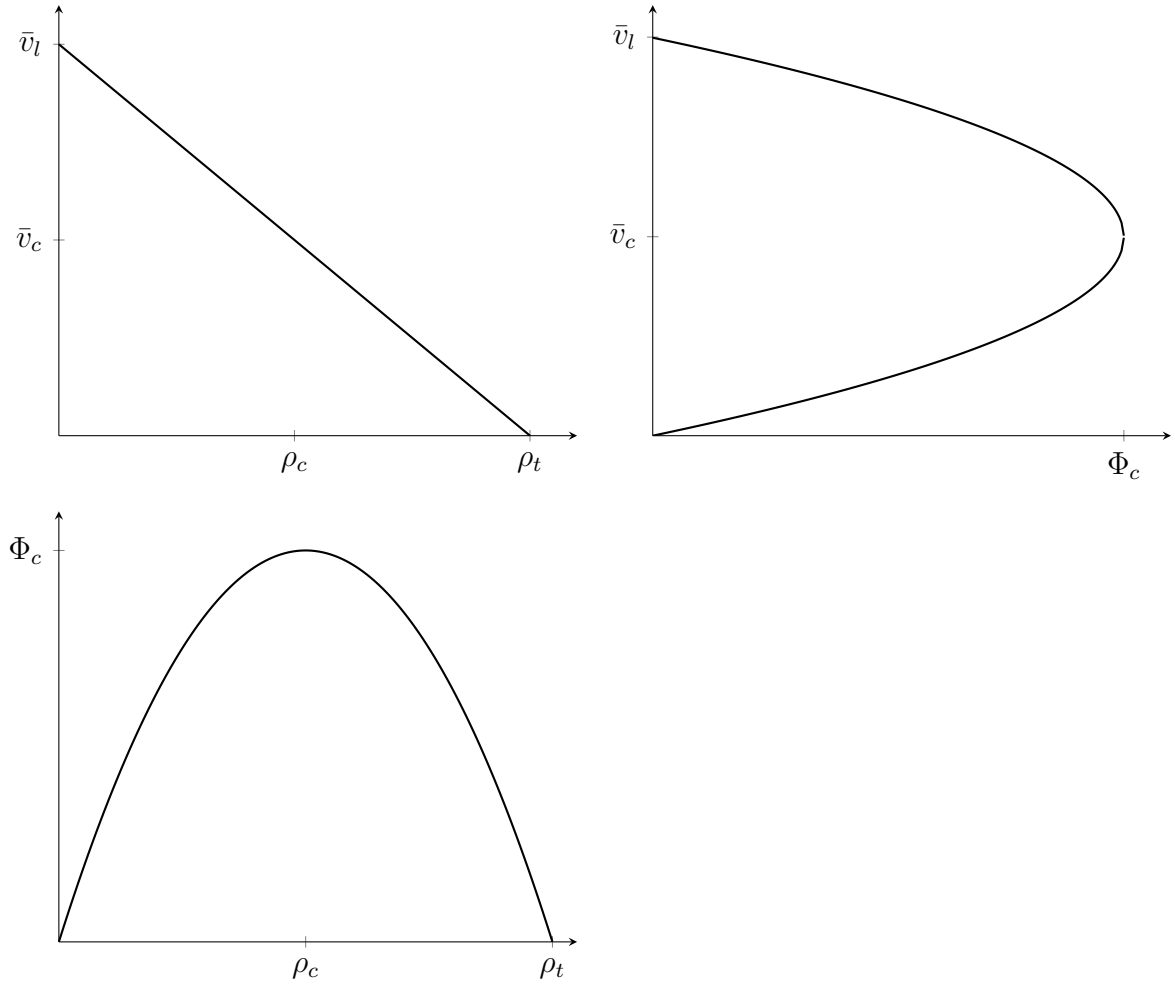


Figura 1.1: *Diagrammi fondamentali d Greenshield.*

In una situazione stazionaria (rete in equilibrio) è possibile descrivere il sistema graficamente con tre diagrammi: $\bar{v}$-Φ, Φ-ρ e $\bar{v}$-ρ. La prima formulazione di questi, riportata come esempio in Fig. 1.1, è stata effettuata da Greenshield sulla base di alcune misurazioni da lui eseguite. Assumendo lineare la relazione tra ρ e $\bar{v}$ le relazioni negli altri due diagrammi risultano paraboliche. In particolare, si ottengono un massimo di flusso sia a $\rho_c = \frac{\rho_t}{2}$ sia per $\bar{v}_c = \frac{\bar{v}_l}{2}$, con ρ_j e $\bar{v}_l$ capacità e velocità massime, rispettivamente.
Studiando i diagrammi fondamentali è possibile suddividere le condizioni di traffico in tre regimi:

- *Completamente Libero*
  Quando i veicoli non sono condizionati dal traffico ed è per loro possibile viaggiare alla velocità massima $\bar{v}_l$, ossia la velocità *libera*. La velocità libera dipende solo

dalla geometria e dalle restrizioni applicate ad una strada. Si osservi come per questo valore di velocità si abbiano un flusso e una densità prossimi allo 0.

- *Saturo*
  Nelle strade sature il flusso e la velocità tendono a 0 e i veicoli si accodano ad una densità massima ρ_t (densità di *traffico*).

- *in Capacità*
  La capacità della strada eguaglia il flusso massimo Φ_c, il quale ha associate una densità ρ_c e una velocità $\bar{v}_c$. Si ha sempre $\bar{v}_c < \bar{v}_l$.

Capitolo 2

Costruzione del modello

2.1 Costruzione del modello

Si consideri una matrice di adiacenza pesata $\mathcal{W}_{ij}$ che differisce dalla matrice di adiacenza $\mathcal{A}_{ij}$ in quanto non possiede solamente i valori $\{0, 1\}$. In particolare, questi pesi rappresentino la lunghezza delle strade (collegamenti) tra i vari nodi della rete, ovvero $\mathcal{W}_{ij} \geq 0$. Una volta definito il network stradale è possibile definire le classi di agenti $c(s, d)$ che vi circoleranno sopra, le quali saranno definite da un nodo s sorgente e da un nodo d destinazione. Ogni individuo dovrà quindi muovere dalla sua sorgente alla destinazione prefissate, seguendo il percorso più breve (a costo minore) tra tutti i percorsi disponibili.

Capitolo 3

Risultati

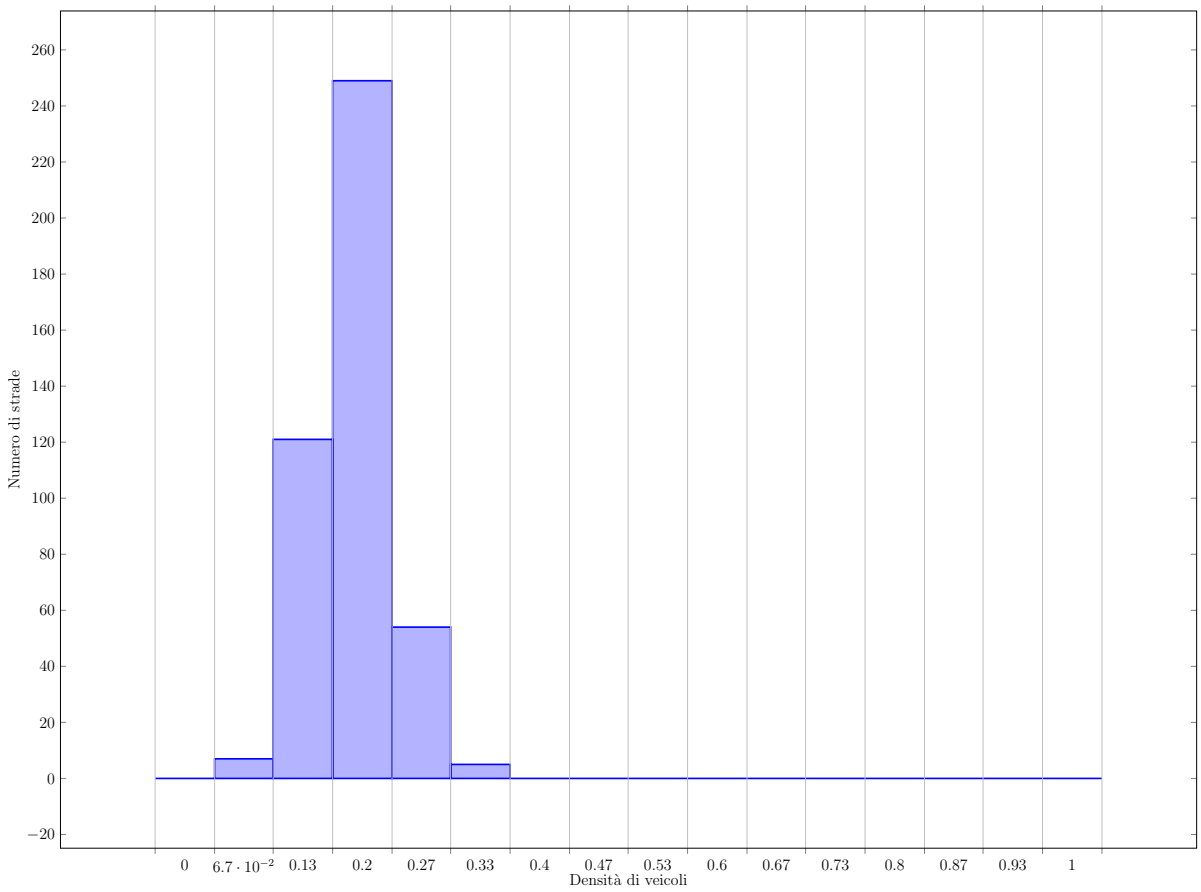


Figura 3.1: *Prova.*

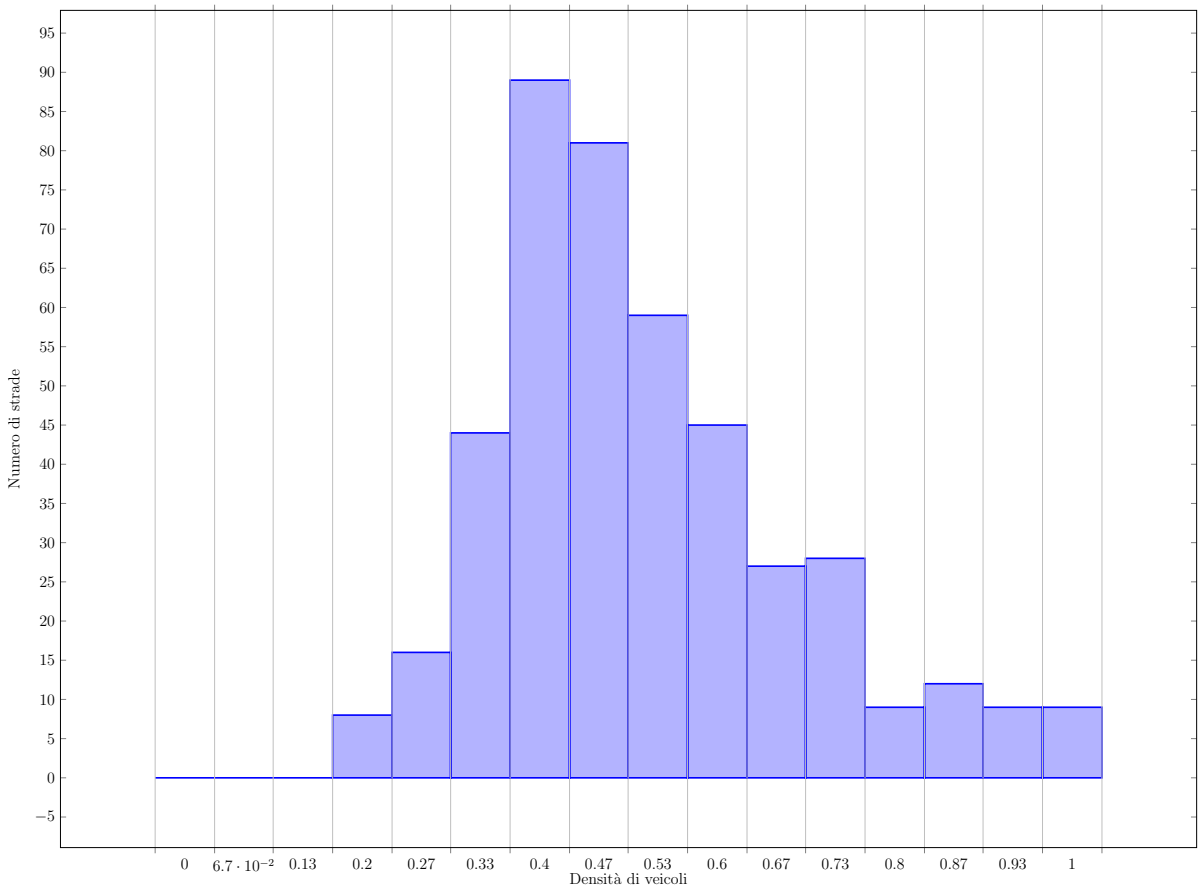


Figura 3.2: *Prova.*

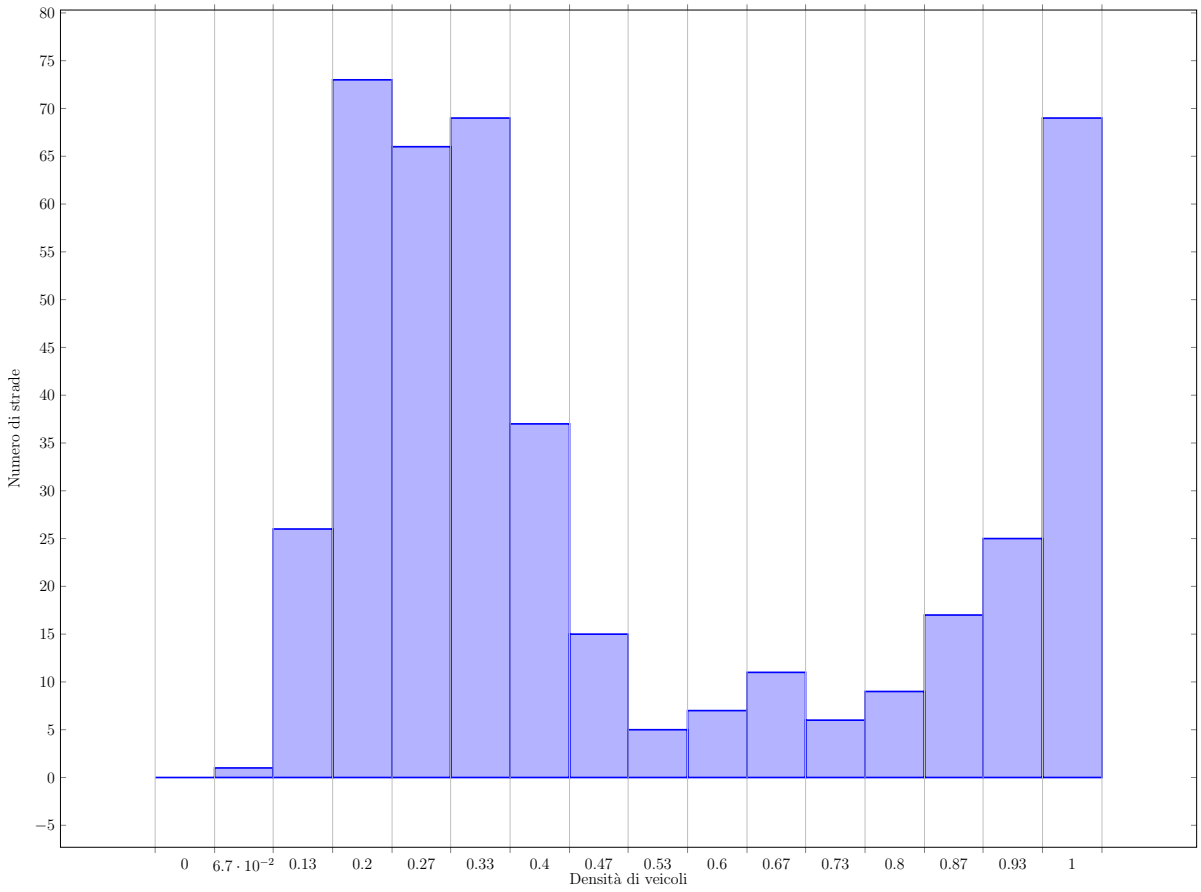
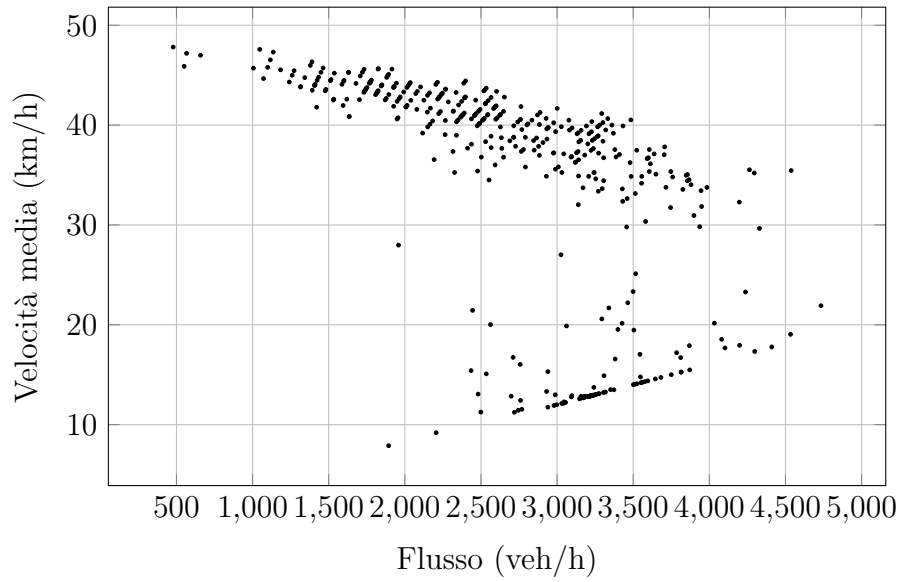
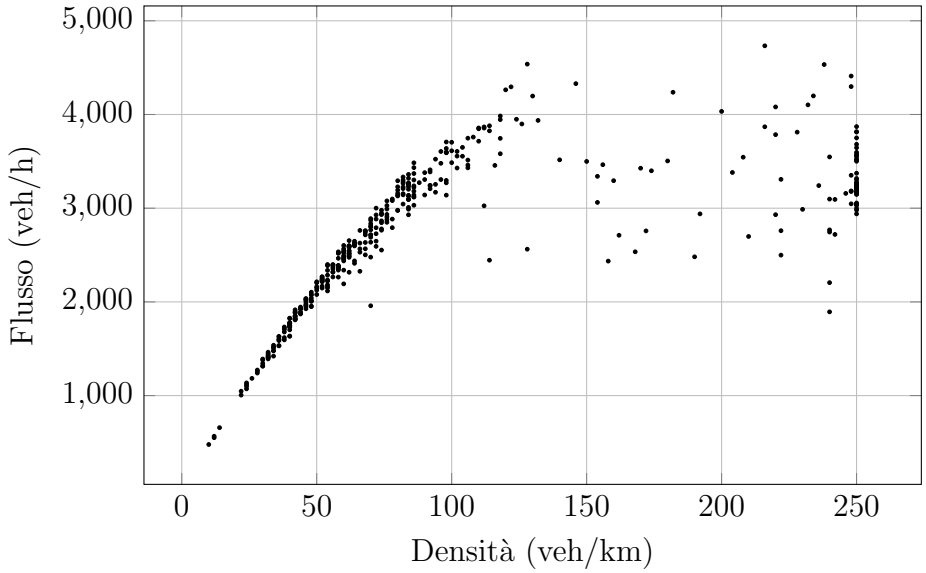
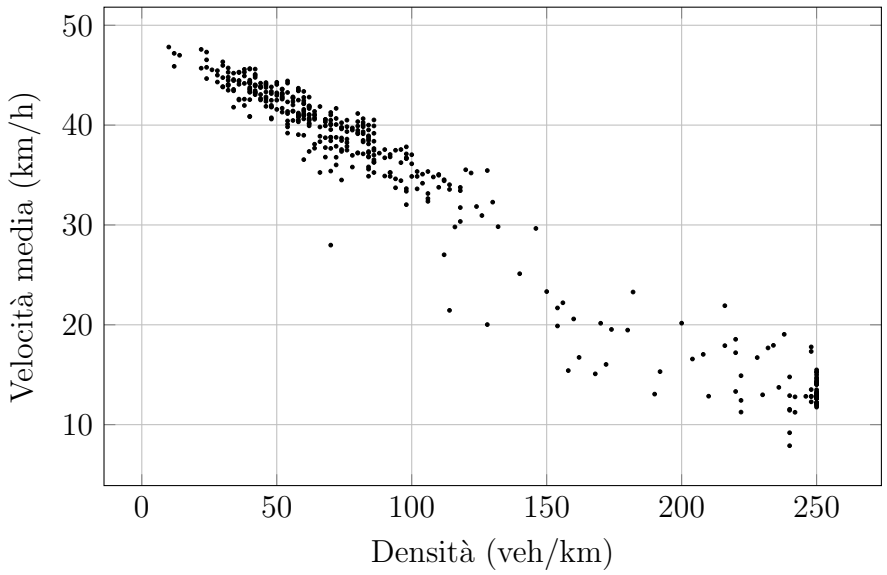
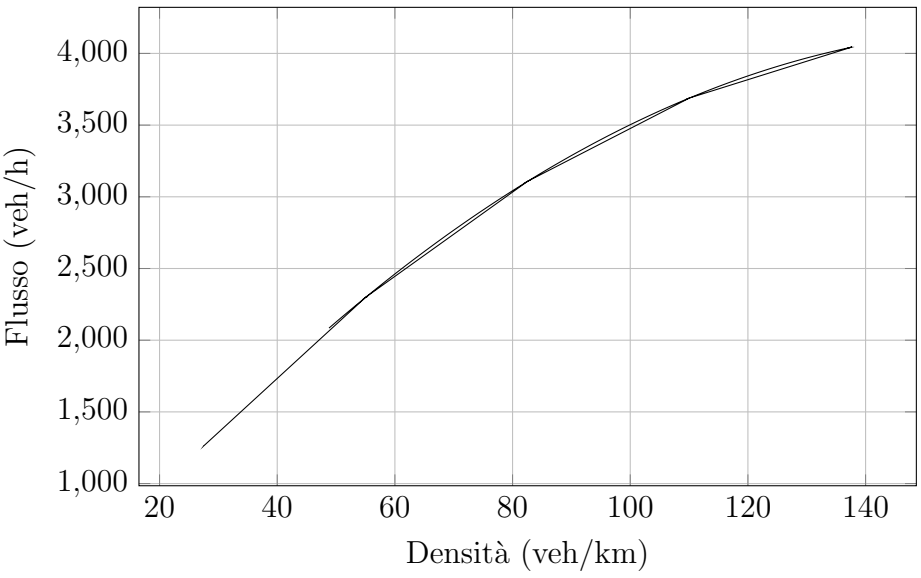


Figura 3.3: *Prova.*


Velocità media (km/h)
50
40
30
20
10
500
1,000
1,500
2,000
2,500
3,000
3,500
4,000
4,500
5,000
Flusso (veh/h)


Flusso (veh/h)
5,000
4,000
3,000
2,000
1,000
0
50
100
150
200
250
Densità (veh/km)


Velocità media (km/h)
50
40
30
20
10
0
50
100
150
200
250
Densità (veh/km)


Flusso (veh/h)
4,000
3,500
3,000
2,500
2,000
1,500
1,000
20
40
60
80
100
120
140
Densità (veh/km)

Appendice A

Implementazione

Il modello descritto in precedenza é stato implementato tramite un software scritto in C++.

A.1 Classi

Sfruttando la programmazione a oggetti su cui é basato il linguaggio C++ si é diviso il modello in varie classi.

A.1.1 VehicleType

La classe a livello inferiore é *VehicleType* che, come suggerisce il nome, definisce una tipologia di veicolo. In questo modello ogni veicolo é caratterizzato dai parametri di input nodo sorgente e nodo destinazione. Tuttavia, per muoversi sul network, ogni tipologia di veicolo necessita anche di una matrice di transizione contenente le probabilità di effettuare o meno un passo in una specifica dimensione. Questa matrice viene solo dichiarata come parametro della classe e viene impostata dopo la definizione del network.

A.1.2 Vehicle

Dopo aver definito le tipologie di veicoli é necessario definire anche i veicoli stessi. La classe *Vehicle* rappresenta gli agenti che andranno a muoversi sul network stradale. Un vettore statico di *VehicleType* permette ad ogni veicolo di avere una tipologia definita, tramite un parametro indiciale che determina la posizione nel vettore. Ogni agente ha inoltre due coordinate che ne definiscono la posizione: nodo attuale e strada attuale. Per permetterne il movimento nel tempo sono presenti altri due parametri, non necessari in input, rappresentanti la velocità del veicolo, dettata dalla strada sulla quale si trova, e la penalità di tempo che questo deve scontare, dipendente sia dalla velocità del veicolo stesso sia alla densità di veicoli presente sulla strada in cui si trova.

A.1.3 Street

Una volta definiti i veicoli é necessario definire le proprietà dei collegamenti tra i vari nodi (incroci) della rete. Ogni istanza della classe *Street* rappresenta un collegamento tra due nodi. I parametri da fornire come input per distinguere una strada in maniera univoca sono l'indice del nodo sorgente e l'indice del nodo destinazione. Si presti ora attenzione al fatto che ogni strada abbia una direzione: considerando due nodi generici i e j, la strada che connette $i \to j$ sarà differente dalla strada che connette $j \to i$. Questa distinzione permette sia una gestione delle densità di veicoli più efficiente e coerente con la realtà, non avendo interferenza tra le corsie, sia l'inserimento di strade a senso unico nella rete.
Nella classe *Street* viene poi definito un parametro di controllo del modello, ossia la lunghezza media dei veicoli. Altri parametri della strada sono la sua lunghezza, il numero n di veicoli su di essa, la velocità massima consentita, il numero di corsie (direzionate come la strada stessa) e la capacità massima n_{max} di veicoli presenti contemporaneamente. Si noti come mentre un veicolo conosce esattamente la strada in cui si trova ciò non sia vero per la strada in quanto quest'ultima possiede informazione solamente sul numero totale di veicoli presenti su di essa. La velocità effettiva mantenibile su una strada, essendo un valore altamente dinamico, non viene considerata come parametro (quindi immagazzinato in memoria) ma viene calcolato tramite una funzione quando necessario. In particolare, l'andamento della velocità su una strada segue la funzione

$$v(n) = v_{max}\left(1 - 0.75\frac{n}{n_{max}}\right) \tag{A.1}$$

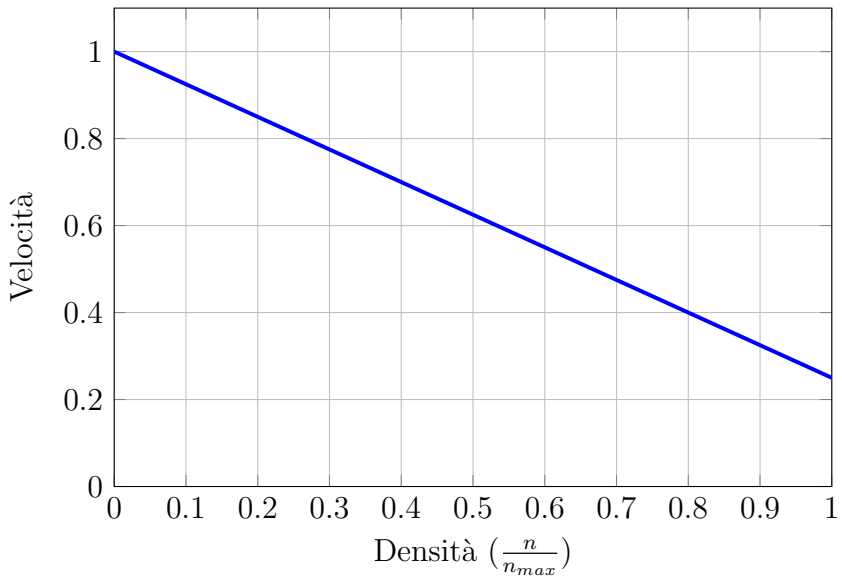


Figura A.1: *Ipotesi dell'andamento della velocità in funzione della densità.*

Come visibile in Fig. A.1 anche una volta raggiunta la densità massima i veicoli non si fermano ma si immettono sulla strada, quando si libera sufficiente spazio, con una velocità minima pari al 25% della velocità massima.

A.1.4 Graph

Ultima classe definita, che comprende tutte le precedenti, é la classe *Graph*, la quale costruisce effettivamente il network stradale. Parametro di input necessario per creare un'istanza é la matrice di adiacenza, che definisce le connessioni tra i nodi. Tramite essa viene poi generato un vettore di puntatori a *Street* che genera le connessioni tra i nodi come strade.
Il movimento degli agenti sul network é determinato dalla matrice di transizione assegnata alle varie tipologie di veicoli. Questa viene generata tramite l'utilizzo del famoso algoritmo Dijkstra [2] e si basa sulla ricerca del *best path*, il percorso a costo (lunghezza) inferiore, dalla posizione dell'agente alla destinazione definita dal suo *VehicleType*. Si é poi deciso di introdurre un parametro di temperatura statistica al network per poter permettere ai veicoli di seguire un percorso differente rispetto al best path fornito dall'algoritmo Dijkstra. L'algoritmo di evoluzione, infatti, assegna peso 1 ai best path e un peso π variabile tra 0 e 1 ai percorsi più lunghi. Quest'ultimo peso varia in base alla temperatura del sistema secondo la funzione

$$\pi(T) = \tanh(kT) \tag{A.2}$$

dove k é un parametro di controllo del modello.

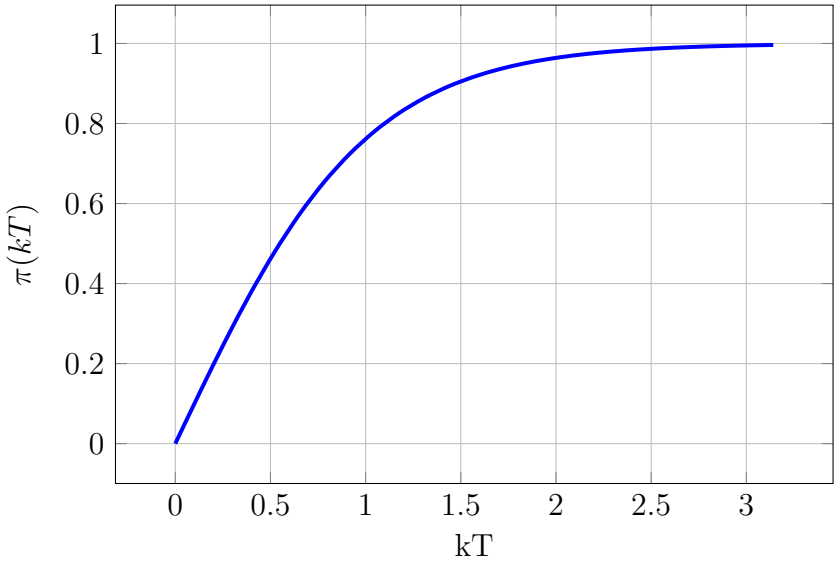


Figura A.2: *Probabilità di errore in funzione della temperatura.*

Si procede poi alla normalizzazione a 1 di ogni vettore riga della matrice in modo tale da poter ottenere la probabilità di transizione. Una volta ottenute le probabilità di transizione il sistema può evolvere tenendo presente che:

- se un agente prova a muovere su una strada piena questo movimento viene impedito e di fatto si perde uno step temporale;
- la velocità di ogni veicolo viene impostata all'ingresso in una strada e non più modificata fino all'ingresso nella strada successiva.

A.2 Esecuzione

A.3 Performance

Il programma é stato compilato utilizzando il compilatore gcc-9 su Ubuntu 20.04 nel Windows Subsystem for Linux. La compilazione é stata effettuata utilizzando le flag

```
-O3 -Wall -Wextra -fsanitize=address
```

in cui:

- *O3* indica il livello di ottimizzazione massima volto a ridurre il tempo di esecuzione del programma;

- *Wall* e *Wextra* consentono di correggere ogni tipo di warning che sorge in compilazione;
- *fsanitize=address* consente di ottenere un'ottima gestione della memoria.

Bibliografia

[1] A. Bazzani. “Random walks on Graphs, Master Equation and Maximal Entropy Principle”. In: 2015.

[2] Thomas Cormen et al. *Introduction to Algorithms, Second Edition*. Gen. 2001, pp. 504–508. ISBN: 0-262-03293-7.

[3] S. Logghe Prof. L.H. Immers. “Traffic Flow Theory”. In: 2002.