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 ad oggi uno dei maggiori problemi per lo sviluppo delle città. Nel solo stato della Florida, ad esempio, è stato stimato che dal 2003 al 2007 queste abbiano causato perdite dai 4.5 ai 7 miliardi di dollari annui [1]. Oltre a danni economici, le congestioni nel traffico causano anche problemi sociali, come l'aumento del livello di stress negli automobilisti [2] e ambientali, in quanto le emissioni da esse causate sono altamente inquinanti e si ripercuotono sulla salute dei cittadini [3]. Nell'ultimo decennio, con l'obiettivo di attenuare questa problematica, nelle grandi città hanno cominciato a proliferare diverse compagnie di ride-hailing, come Uber. Nonostante il beneficio economico portato da esse e dalla competitività del mercato la presenza in gran numero di questi fornitori di servizi va a peggiorare la qualità della mobilità urbana. Questo effetto diventa abbastanza rilevante in zone nelle quali vi è poca domanda o la velocità stradale media è sufficientemente bassa. Ogni operatore aggiunto di questo settore causa, ad esempio, un aumento del numero totale di veicoli su strada del 2.5% a Manhattan e del 37% a San Francisco [4]. Inoltre, la presenza di molteplici compagnie di ride-hailing aggiunge ogni anno 9.17 miliardi di km di strade nelle aree metropolitane di Boston, Chicago, Los Angeles, Miami, New York, Philadelphia, San Francisco, Seattle e Washington DC [5].

Un ruolo fondamentale nella gestione delle congestioni nelle città lo gioca la logistica delle stesse. Si consideri ora la rete stradale di una città generica di superfice approssimabile a quella di una circonferenza di raggio R. È lecito ipotizzare che il numero M di nodi (incroci) della rete cresca proporzionalmente alla superficie della città, quindi al quadrato del raggio $M \propto R^2$. Si definisca ora la variabile costo C della rete, la quale dipenderà sia dal numero di nodi che dalla lunghezza di scala della stessa, ossia $C \propto MR^2$. Unendo le due relazioni precedenti si ottiene $C \propto M^{\frac{3}{2}}$ ed è possibile constatare che per una città ideale come quella considerata il costo della logistica stradale cresca all'aumentare del numero di nodi. La nascita e lo sviluppo delle metropoli, avvenuta nell'ultimo secolo, ha quindi cause da ricercarsi non nel miglioramento della logistica quanto a cambiamenti nella mobilità, come l'introduzione di mezzi pubblici quali autobus, tram e metropolitane.

Per studiare questo fenomeno si sono evoluti negli anni diversi modelli, basati sia su approcci di tipo microscopico che macroscopico.

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 [6] 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.

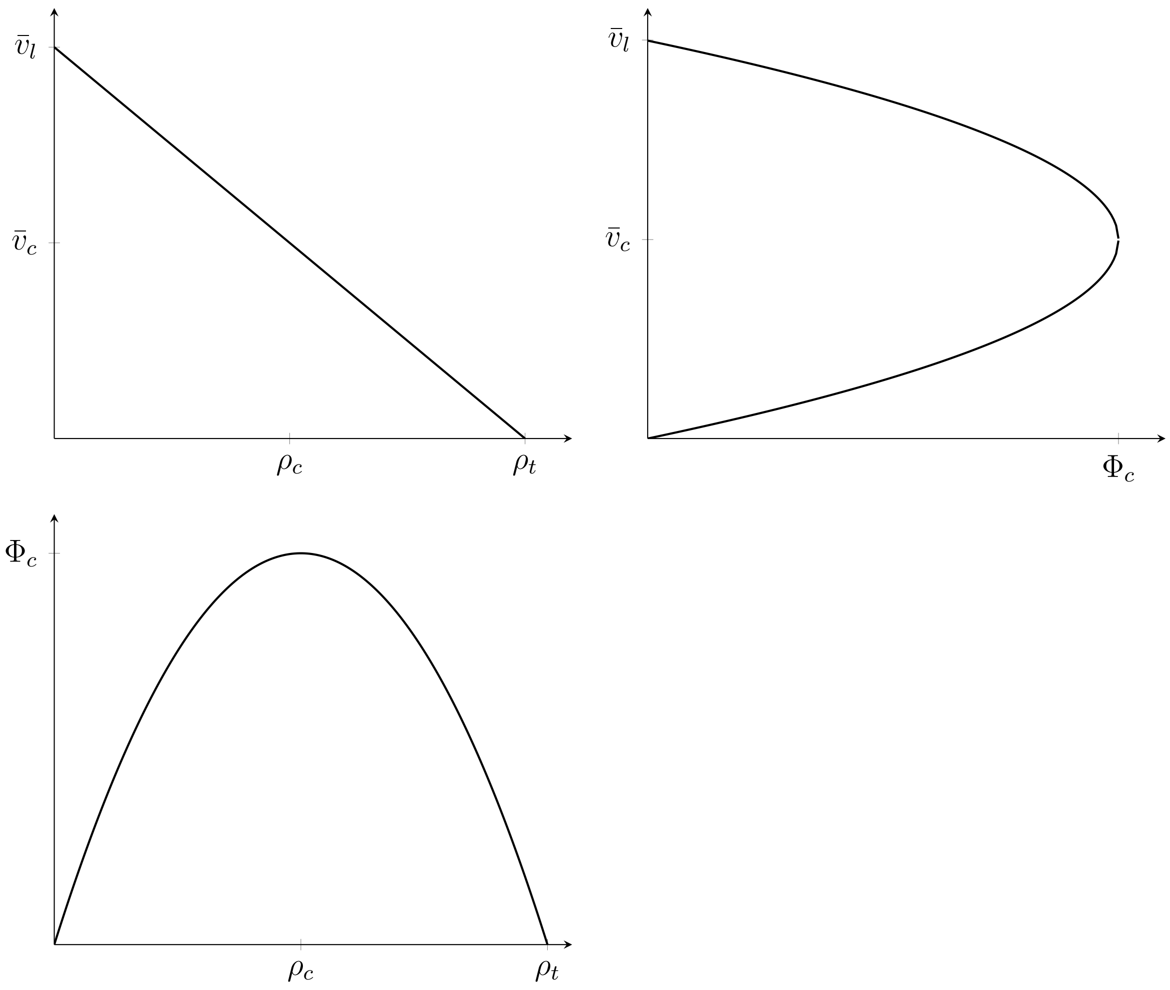


Figura 1.1: *Diagrammi fondamentali d Greenshield.*

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 Networks

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). Affiancata alla matrice di adiacenza si usa definire spesso la matrice dei pesi $\mathcal{W}_{ij}$, la quale definisce il peso di ciascun collegamento tra nodi. In particolare, la matrice $\mathcal{W}_{ij}$ possiede le seguenti propietà:

- $\mathcal{A}_{ij} = 0 \Longrightarrow \mathcal{W}_{ij} = 0$;
- $\mathcal{A}_{ij} = 1 \Longrightarrow \mathcal{W}_{ij} \neq 0$.

Dalla matrice di adiacenza è possibile definire il grado del nodo i-esimo come

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

che indica il numero dei link per ogni nodo.
Una volta note le matrici di adiacenza e il vettore dei gradi la matrice Laplaciana del network è definita come

$$\mathcal{L}_{ij} = d_i \delta_{ij} - \mathcal{A}_{ij} \tag{2.1}$$

ed ha le seguenti proprietà:

- è semi-definita positiva;
- $\mathcal{L}_{ij} > 0 \Longleftrightarrow i = j$;
- $\sum_j \mathcal{L}_{ij} = \sum_i \mathcal{L}_{ij} = 0$, 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.

2.1.1 Random Walk su network

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:

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

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

$$\delta_\alpha(i, t + \Delta t) = \sum_j \xi^\alpha_{ij} \delta_\alpha(j, t)$$

dove ξ^α_{ij} 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

$$n_i(t) = \sum_\alpha \delta_\alpha(i, t)$$

ed è possibile dimostrare [7] che la seguente equazione è un integrale del moto

$$\sum_i n_i(t) = N \tag{2.2}$$

2.2 Costruzione del modello

Si consideri ora un network generato dalla matrice di adiacenza simmetrica $\mathcal{A}_{ij}$ i cui nodi rappresentano gli incroci di una rete stradale. Alla matrice di adiacenza è associata una matrice dei pesi $\mathcal{S}_{ij} \geq 0$ che, associando un diverso peso ad ogni link, definisce la lunghezza delle strade. Su questo network si definiscano ora le classi di agenti $C_1, \ldots, C_k$, ognuna delle quali caratterizzata da un nodo sorgente src e destinazione dst e denotate come $C_\alpha(src, dst)$. Ogni individuo dovrà, nel corso della simulazione, muovere tra i nodi $src \to dst$ in modo da seguire la geodetica, ossia il percorso più breve. Il costo di un percorso, tuttavia, non viene calcolato sulla base della lunghezza: data la mobilità dei veicoli, risulta più accurato considerando il tempo di percorrenza. A tale fine si definisca ora un'altra matrice dei pesi $\mathcal{V}_{ij} \geq 0$ rappresentante la velocità massima alla quale un agente può andare su una determinata strada. L'elemento di matrice

$$\mathcal{T}_{ij} = \begin{cases} \frac{\mathcal{S}_{ij}}{\mathcal{V}_{ij}} & \mathcal{V}_{ij} \neq 0 \\ 0 & \mathcal{V}_{ij} = 0 \end{cases} \tag{2.3}$$

rappresenta dunque il costo temporale associato al link $i \to j$.
Discretizzando il tempo con un passo Δt, il costo di un percorso risulta esprimibile come

$$G = c\Delta t \tag{2.4}$$

dove $c \in \mathbb{N}$. Nel limite di individui non interagenti tra di loro è quindi possibile calcolare costo della geodetica, chiamata *best path*, come

$$G_{\text{best}} = c_{\text{best}}\Delta t = \min\left\{\sum_{i=src}^{j=dst} \mathcal{T}_{ij}\right\} \tag{2.5}$$

Si assuma ora che gli agenti muovano sulla rete seguendo un moto stile random walk. Per ogni classe di individui C_α è quindi necessario definire una matrice stocastica (di transizione) π_{ij}^α con le proprietà descritte nella sezione precedente. Nella realtà una strada ha una capacità finita, intesa come numero massimo di veicoli presenti su di essa in un generico istante di tempo t. Tale vincolo viene implementato nel modello tramite l'introduzione di una densità massima ρ_{max} posseduta da ogni strada.

2.2.1 Algoritmo di evoluzione

È ora necessario definire come i veicoli muovano effettivamente sul network. La probabilità di transizione di osgni classe π_{ij}^α viene assegnata nel modo seguente:

- fissato il nodo di partenza i si inizia a scorrere sul passo successivo nel nodo j;
- se il nodo j si trova sul percorso G_{best}, definito dall'Eq. (2.5), allora viene assegnato un peso $\pi_{ij} = 1$;
- altrimenti, se il link esiste ($\mathcal{A}_{ij} \neq 0$) viene assegnato un peso $\pi_{ij} = \tanh \beta T$, dove β è un parametro di controllo del modello e T rappresenta una temperatura statistica;
- una volta controllati tutti i possibili j il vettore riga i-esimo viene poi normalizzato in modo tale da avere $\sum_j \pi_{ij} = 1$.

Si osservi che grazie all'introduzione della temperatura statistica T, graficata in Fig. 2.1, è possibile permettere agli agenti di "sbagliare" percorso e uscire dalla geodetica, introducendo così delle fluttuazioni. Inoltre, per $T \to \infty$ l'evoluzione del sistema diventa equivalente ad un random walk su network in quanto ogni scelta di percorso ha la medesima probabilità.

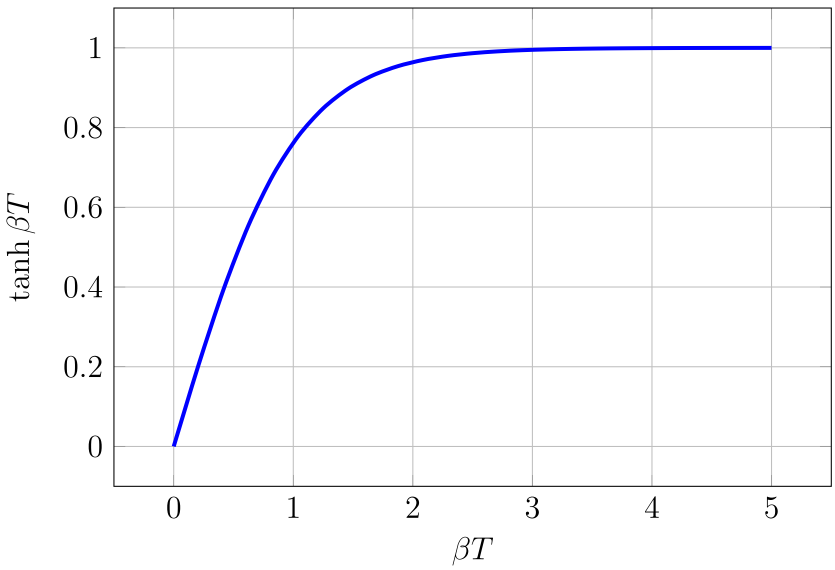


Figura 2.1: *Probabilità di errore in funzione della temperatura statistica.*

Considerando ora un individuo generico appartente alla classe C_α si procede nel modo seguente:

- si controlla se sia in grado di muoversi, quindi se $c = 0$. In caso negativo si sconta uno step temporale $c = c - 1$ e si prosegue con gli altri veicoli;
- in caso affermativo, il passo successivo sarà deciso stocasticamente dalla matrice π_{ij}^{α};
- se la strada di arrivo è piena si perde uno step temporale, altrimenti l'individuo si immette sulla strada che connette i nodi $i \rightarrow j$ regolando la sua velocità secondo la legge
$$v(t) = v_{max} \left(1 - k \frac{\rho(t)}{\rho_{max}} \right) \tag{2.6}$$
dove $\rho(t)$ rappresenta la densità di veicoli presenti sulla strada al tempo t e k è un parametro di controllo del modello;
- in base alla velocità acquisita all'invidiuo viene poi assegnata una nuova penalità temporale $c = \frac{L}{v(t)\Delta t}$, con L lunghezza della strada in cui si trova.

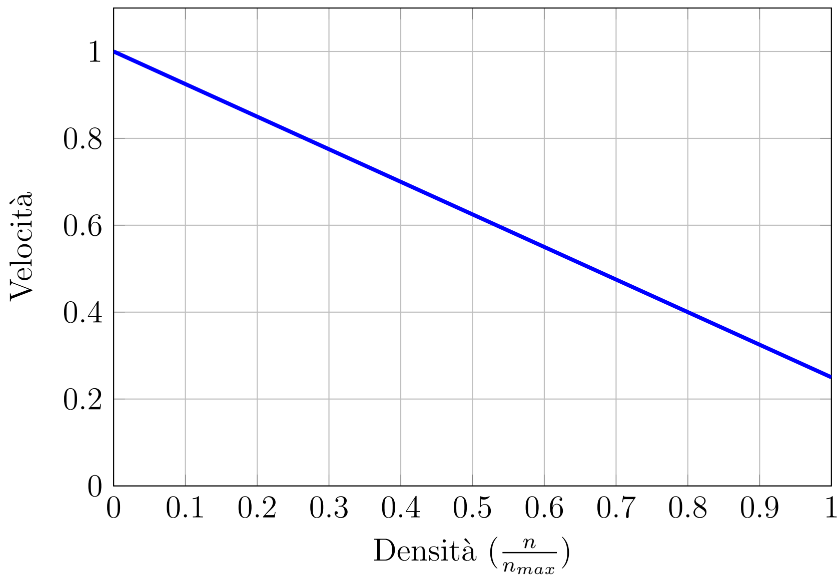


Figura 2.2: *Ipotesi dell'andamento della velocità in funzione della densità per* $k = 0.75$.

Si noti come l'Eq. (2.5), graficata in Fig. 2.2, tenda all'ipotesi di Greenshield per $k = 1$ (cfr. Eq. (1.1)).

Capitolo 3

Risultati

3.1 Modalità di esecuzione

Nella sezione precedente si è costruito il modello ed è emerso come questo si basi su alcuni parametri di controllo. Variando questi parametri è di fatto possibile ottenere risultati estremamente diversi tra di loro. Assumendo che uno step temporale del modello sia equivalente a 1 s

3.2 Diagrammi fondamentali

3.3 Isteresi

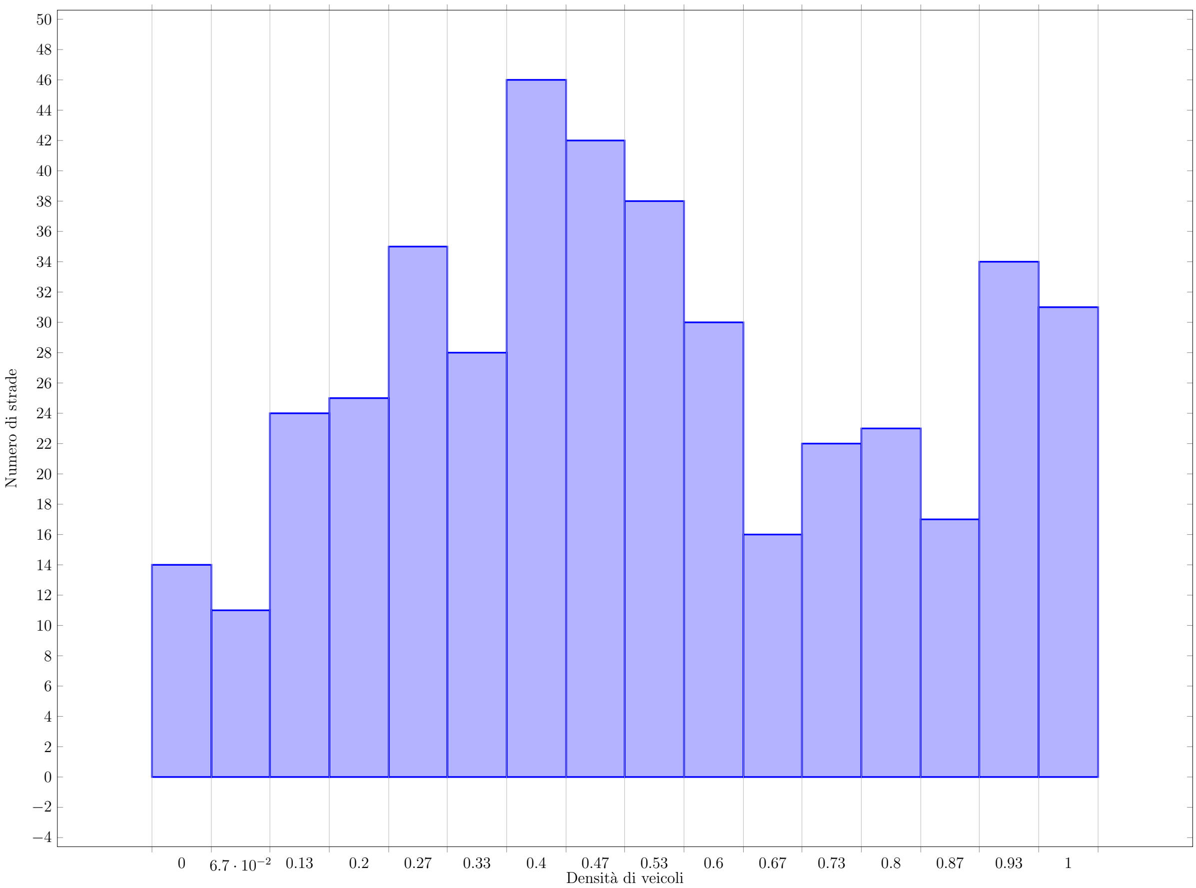


Figura 3.1: *Prova.*

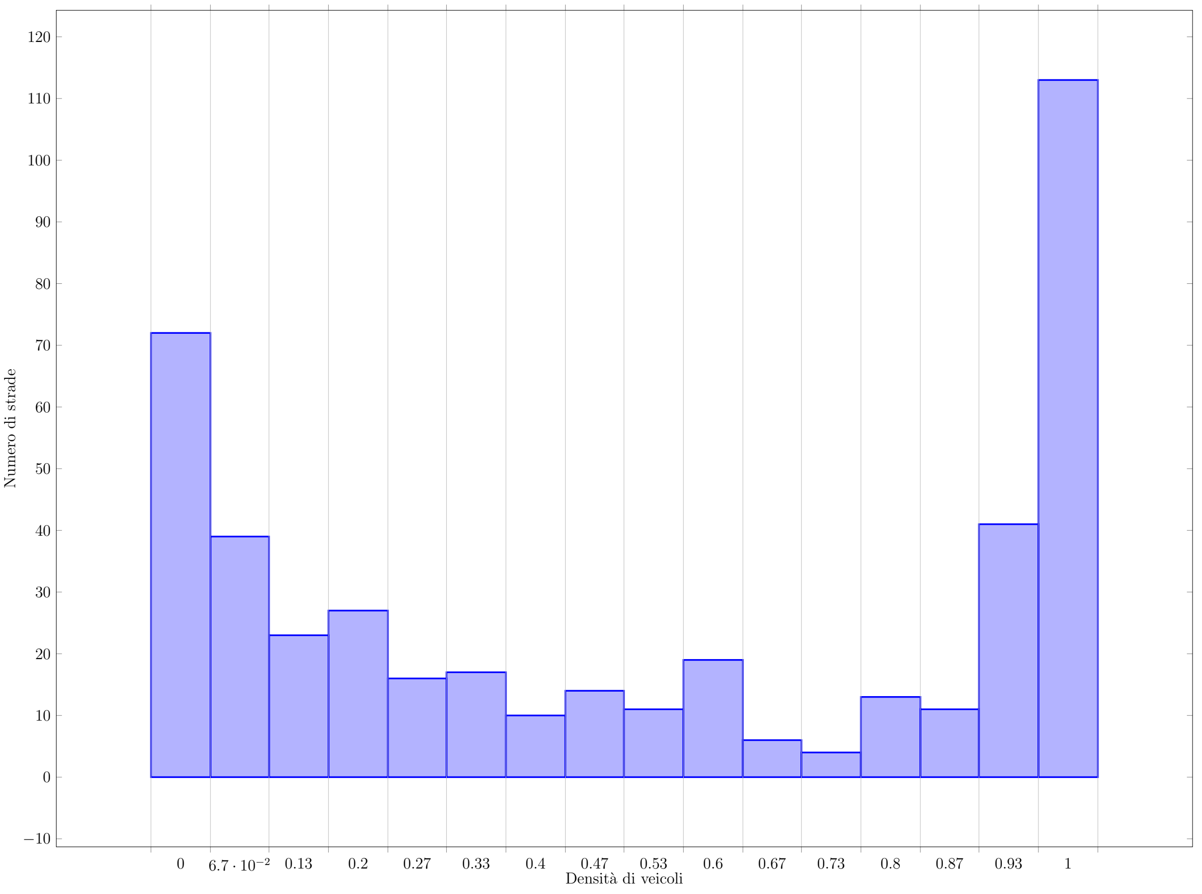


Figura 3.2: *Prova.*

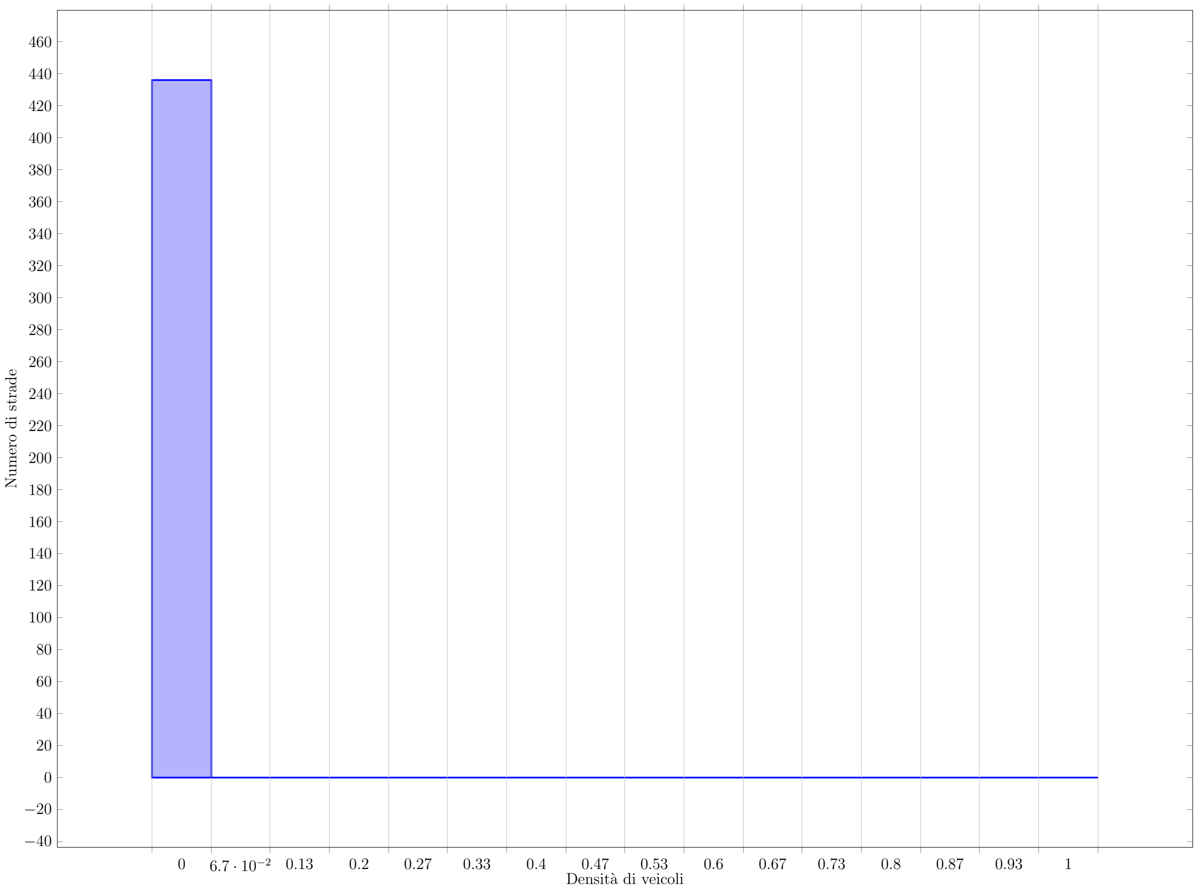


Figura 3.3: *Prova.*

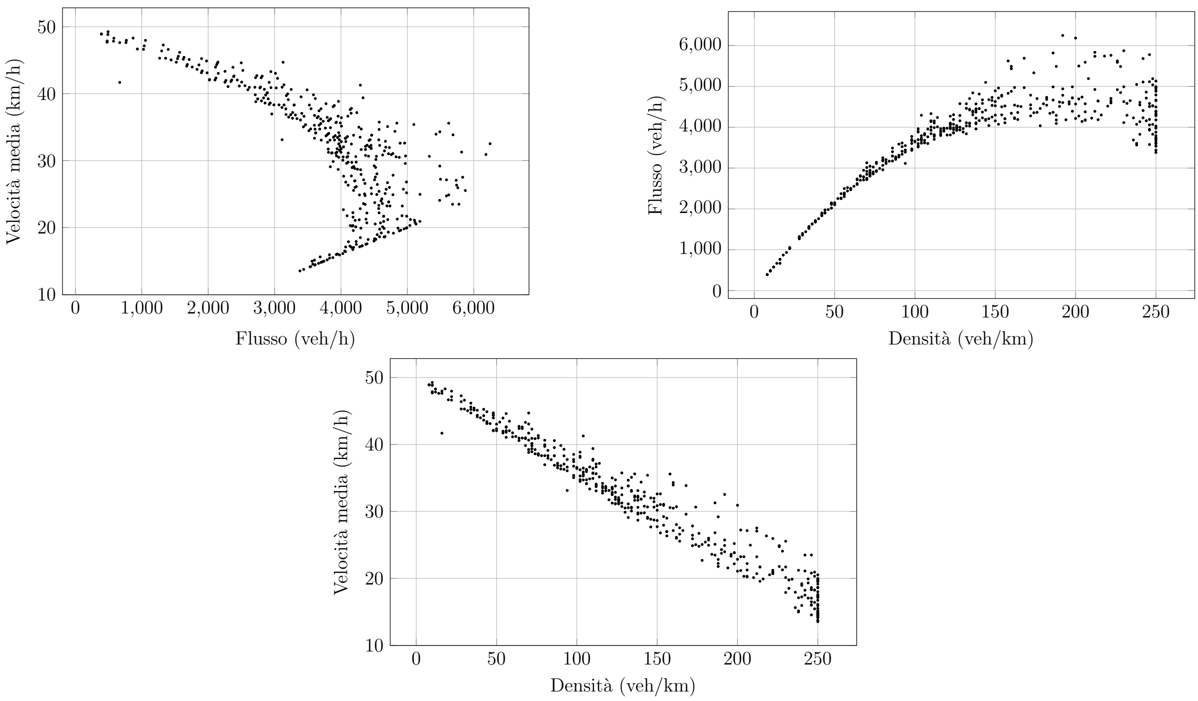
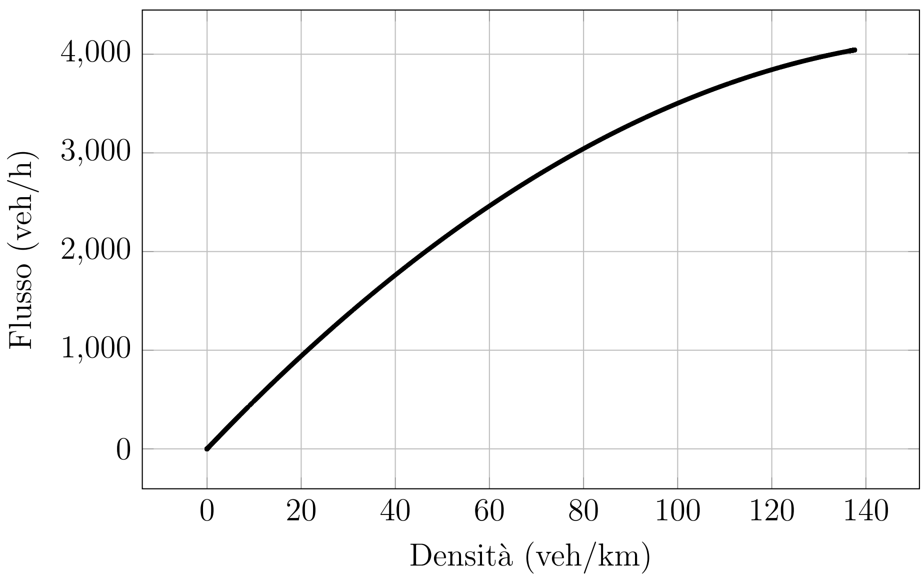


Figura 3.4: *Diagrammi fondamentali con distribuzione omogenea.*



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

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

Come visibile in Fig. 2.2 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 [8] 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. 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] R. Wang, *The economic cost of traffic congestion in Florida*, ago. 2010. DOI: 10.13140/RG.2.2.11217.43368.

[2] D. A. Hennessy e D. L. Wiesenthal, "Traffic congestion, driver stress, and driver aggression," *Aggressive Behavior: Official Journal of the International Society for Research on Aggression*, vol. 25, n. 6, pp. 409–423, 1999.

[3] K. Zhang e S. Batterman, "Air pollution and health risks due to vehicle traffic," *Science of the total Environment*, vol. 450, pp. 307–316, 2013.

[4] D. Kondor, I. Bojic, G. Resta, F. Duarte, P. Santi e C. Ratti, "The cost of non-coordination in urban on-demand mobility," *Scientific Reports*, vol. 12, n. 1, p. 4669, mar. 2022, ISSN: 2045-2322. DOI: 10.1038/s41598-022-08427-2. indirizzo: https://doi.org/10.1038/s41598-022-08427-2.

[5] B. Schaller, "The new automobility: Lyft, Uber and the future of American cities," 2018.

[6] S. L. Prof. L.H. Immers, "Traffic Flow Theory," 2002.

[7] A. Bazzani, "Random walks on Graphs, Master Equation and Maximal Entropy Principle," 2015.

[8] T. Cormen, C. Leiserson, R. Rivest e C. Stein, *Introduction to Algorithms, Second Edition.* gen. 2001, pp. 504–508, ISBN: 0-262-03293-7.